SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

План лекції . Тема 5. Інструменти та технології системного моніторингу.

- Огляд популярних інструментів системного моніторингу, таких як Nagios, Zabbix, Prometheus, MS SCOM.
- Практичне використання інструментів для моніторингу системних ресурсів та додатків.
- Налаштування сповіщень та автоматизація управління системним моніторингом.

Вступ

У нас п'ята лекція курсу. Ми дібралися безпосередньо до інструментів, якими здійстнюється моніторинг. Нагадаю

Моніторинг є ключовою складовою управління та підтримки інформаційних систем та технологій. Існує кілька видів моніторингу, кожен з яких має свої характеристики та важливість для специфічних цілей та потреб.

Ця лекція присвячена інструментам системного моніторингу

Системний моніторинг - це комплексний процес автоматичного або ручного нагляду, аналізу та оцінки параметрів та функцій певної системи або групи систем. Цей процес організується для надання повної та об'єктивної інформації про стан системи в реальному часі або на певних етапах її роботи. Основна мета - забезпечити вчасне виявлення та реагування на проблеми або відхилення в роботі системи для забезпечення її стабільності та ефективності.

Нагадаю, **про велику роль системного моніторингу у сучасних інформаційних технологіях**, а саме на значенні системного моніторингу для бізнесу та організацій:
Системний моніторинг представляє собою ключовий елемент в управлінні та забезпеченні ефективності інформаційних технологій (IT) в сучасних бізнес-середовищах. Ми пам'ятаємо, що цей процес включає систематичний нагляд за функціональністю, продуктивністю та надійністю IT-систем, дозволяючи забезпечити безперебійну роботу інфраструктури та уникнути багатьох проблем для бізнесу за порівняно невеликі кошти.
Загалом, ці питання обговорювалися на першій лекції курсу.

Огляд популярних інструментів системного моніторингу, таких як Nagios, Zabbix, Prometheus, MS SCOM.

Nagios

❖ **Nagios vs Icinga. Історична довідка**

У 1996 році американець Ітан Ґалстад (Ethan Galstad) розробив програму для MS-DOS, яка дозволяла перевіряти доступність серверів Novell NetWare за допомогою сторонніх програм. На основі цієї простої конструкції через три роки у 1999 році була розроблена система з відкритим вихідним кодом NetSaint, який став попередником того, що сьогодні відоме як програмне забезпечення для моніторингу Nagios. Netsaint був розроблений Ітаном для власних потреб. Таким чином, перша версія Nagios датується минулим століттям: 1999 роком.
Через 10 років (Травень 2009) на світ з'явилася Icinga (дериватив Nagios). Про причини, які змусили команду розробників Icinga на чолі з Міхаелем Люббеном (Michael Lubben) - одним з засновників Icinga та провідним розробником доповнень до Nagios, відокремитися і розпочати власний проект є цікава стаття Nagios Vs. Icinga: the real story of one of the most heated forks in free software Треба згадати, що команда Icinga, як і раніше, пересилає всі свої напрацювання до Nagios та зберігається дуже висока сумісність цих двох моніторингових систем. Насправді, Nagios має майже нескінченну кількість відгалужень: Icinga, OpsView, Op5, Centreon, Naemon, Shinken.

❖ **Що таке Nagios** – це інструмент безперервного моніторингу з відкритим вихідним кодом, який контролює мережу, програми та сервери. Він може знаходити та усувати проблеми, виявлені в інфраструктурі, та зупиняти майбутні проблеми, перш ніж вони торкнуться кінцевих користувачів. Це дає повний статус вашої IT-інфраструктури та її продуктивність.

❖ **Чому Nagios.** Nagios пропонує такі функції, які роблять його доступним для великої групи користувачів –
- Може контролювати сервери баз даних, такі як SQL Server, Oracle, MySQL, Postgres
- Надає інформацію про рівень програми (Apache, Postfix, LDAP, Citrix тощо. буд.).
- Забезпечує активний розвиток.
- Має відмінну підтримку від величезної активної спільноти.
- Nagios працює на будь-якій операційній системі.
- Може пінгувати, щоб побачити, чи доступний хост.

❖ **Переваги Nagios.** Nagios пропонує такі переваги для користувачів –
- Допомагає позбутися періодичного тестування.
- Виявляє збої за частки секунди, коли подія перебуває у «переривчастій» стадії.
- Знижує вартість обслуговування без шкоди продуктивності.
- Забезпечує своєчасне оповіщення персоналу та керівництва про контроль та поломки.

❖ **Архітектура Nagios.** Наступні пункти стосуються архітектури Nagios –
- Nagios має архітектуру сервер-агент.
- Сервер Nagios встановлений на хості
- Плагіни (агенти) встановлені на віддалених хостах/серверах, які мають контролюватись. Підтримуються методи безагентного моніторингу.
- Nagios посилає сигнал через процес планування для запуску плагінів на локальних / віддалених хостах / серверах.
- Плагіни збирають дані (завантаження процесора, використання пам'яті тощо. буд.) і відправляють їх у планувальник.
- Потім графіки процесу надсилають повідомлення адміністраторам та оновлюють графічний інтерфейс Nagios.

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

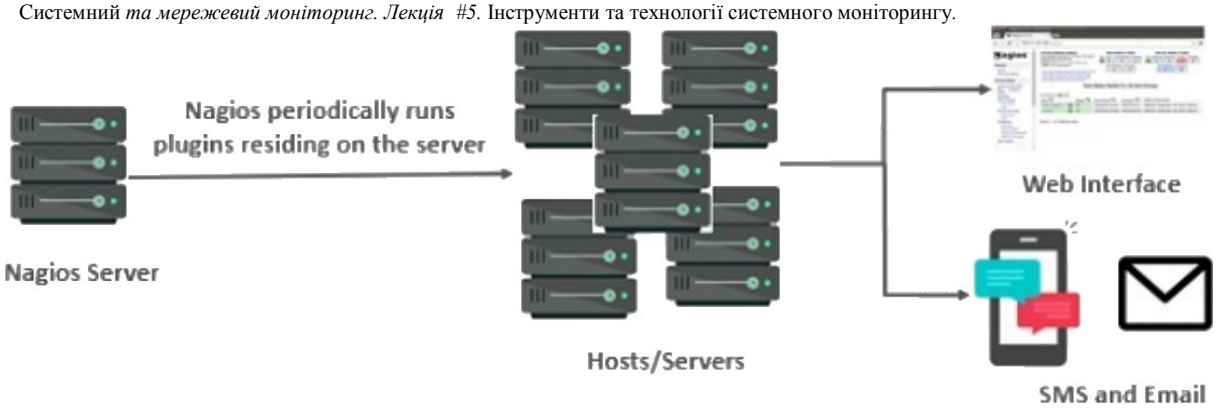


Рис. 05.01. Загальна архітектура Nagios

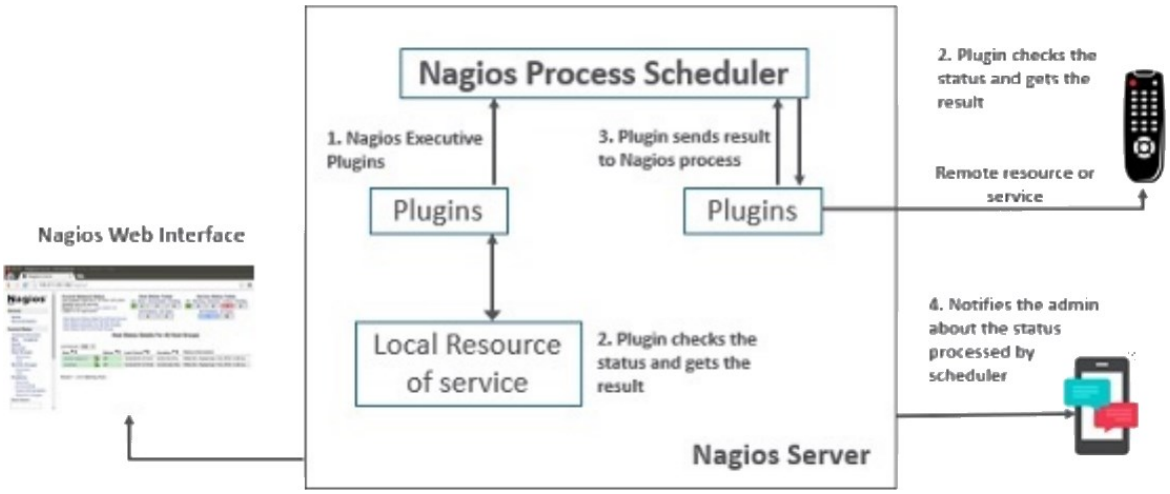


Рис. 05.02. Архітектура серверу Nagios

- Методи **безагентного моніторингу**, доступні в Nagios:
  - **SNMP (Simple Network Management Protocol)**: використовується для моніторингу мережевих пристроїв, таких як маршрутизатори, комутатори та сервери, різноманітної периферії та інших пристроїв, що підтримують SNMP.
  - **WMI (Windows Management Instrumentation)**: використовується для моніторингу систем Windows.
  - **JMX (Java Management Extensions)**: використовується для моніторингу Java-додатків.
  - **HTTP/HTTPS**: використовується для моніторингу веб-серверів та інших веб-додатків.
  - **SSH**: використовується для моніторингу систем Linux та Unix.

- **Nagios – продукти.** Nagios містить лінійку різноманітних продуктів, які дозволяють налаштувати більшість схем та видів системного та мережевого моніторингу. Розглянемо актуальну лінійку продуктів.
  - **Nagios XI**
    Забезпечує моніторинг всіх компонентів IT-інфраструктури, таких як додатки, послуги, мережа, операційні системи і т.д. Nagios XI дає повне уявлення про вашу інфраструктуру та бізнес-процеси. Графічний інтерфейс користувача легко налаштовується, забезпечуючи гнучкість використання. Стандартна версія цього інструменту коштує 1995 доларів, а корпоративна – 3495 доларів. Хоча ціни необхідно уточнювати.
  - **Nagios Core**
    Nagios Core - ядро моніторингу IT-інфраструктури. Продукт Nagios XI також ґрунтується на Nagios Core. Щоразу, коли виникає будь-яка проблема в інфраструктурі, він відправляє попередження/повідомлення адміністратору, який може швидко вжити заходів для вирішення проблеми. Цей інструмент є абсолютно безкоштовним.
  - **Nagios Log Server**
    Nagios Log Server робить пошук даних журналу дуже простим та легким. Усі дані журналу зберігаються в одному місці з налаштуванням високої доступності. Він може легко надсилати оповіщення, якщо будь-яка проблема виявлена в даних журналу. Він може масштабуватись до 1000 серверів, забезпечуючи велику потужність, швидкість, об'єм сховища та надійність для вашої платформи аналізу журналів. Ціна цього інструменту залежить від кількості екземплярів – 1 екземпляр 3995 дол. США, 2 екземпляри 4995 дол. США, 3 екземпляри 5995 дол. США, 4 екземпляри 6995 дол. США, 10 екземплярів 14995 дол. США. Нагадаю, що ціни необхідно уточнювати.
  - **Nagios Fusion**
    Nagios Fusion забезпечує централізоване представлення всієї системи моніторингу. За допомогою Nagios Fusion ви скануєте налаштування окремих серверів моніторингу для різних регіонів. Його можна легко інтегрувати з Nagios XI та Nagios Core, щоб забезпечити повну видимість інфраструктури. Цей інструмент коштує $2495.
  - **Nagios Network Analyzer**
    Nagios Network Analyzer надає повну інформацію про мережну інфраструктуру адміністратору з потенційними загрозами в мережі, щоб адміністратор міг вдатися до швидких дій. Він ділиться дуже докладними даними мережі після глибокого аналізу мережі. Цей інструмент коштує $1995.
  - **Nagios плагіни та додатки.**

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

З усієї лінійки продуктів лише Nagios Core є безкоштовною версією системи моніторингу. Треба відмітити, що його потужності можна значно розширити за допомогою різноманітних плагінів та додатків, які розробляються спільнотою та іншими сторонніми розробниками. Ці додатки дозволяють користувачам налаштовувати моніторинг для різних типів ресурсів та додатків, роблячи Nagios Core більш адаптивним до конкретних потреб користувачів. Ключові можливості розширення Nagios Core безкоштовно включають:

- **Плагіни для моніторингу різних ресурсів**. Спільнота та розробники активно створюють плагіни, які додають підтримку для моніторингу різних системних ресурсів, таких як CPU, пам'ять, диск, мережа, тощо.
- **Розширення для моніторингу додатків та сервісів**. Додатки можуть додавати підтримку моніторингу конкретних додатків, служб чи пристроїв, що робить систему більш гнучкою.
- **Інтеграція з іншими інструментами**. Існує можливість інтеграції Nagios Core з іншими інструментами та сервісами, що розширює його можливості управління та відображення даних.
- **Офіційні плагіни Nagios.** Є 50 офіційних плагінів Nagios. Офіційні плагіни Nagios розробляються та підтримуються офіційною командою плагінів Nagios.
- **Плагіни спільноти**. Існує понад 3000 сторонніх плагінів Nagios, розроблених сотнями членів спільноти Nagios.
- **Плагіни користувача**. Ви також можете написати свої власні плагіни. Існують рекомендації та документація, які необхідно дотримуватися при написанні плагінів користувача.

Взагалі кажучи, спільнота надає широкий спектр безкоштовних розширень, які дозволяють адаптувати Nagios Core під конкретні вимоги користувача, роблячи його ефективним та потужним інструментом системного моніторингу.

- **Встановлення Nagios.** Інсталяція Nagios виконується у ОС Linux з використанням LAMP-стеку (Linux, Apache, MySQL, PHP) або LEMP-стеку (Linux, Nginx, MariaDB, PHP). Детально інсталяція актуальної версії Nagios Core описана у методичних вказівках до лабораторних робіт і може бути виконана за кілька кроків.

- **Файли конфігурації Nagios** знаходяться в /usr/local/nagios/etc. Давайте розберемось з функціональністю та значимістю кожного файлу.
  - **nagios.cfg** Основний конфігураційний файл Nagios Core. Цей файл містить розташування файлу журналу Nagios, інтервал оновлення стану хостів та сервісів, файл блокування та файл status.dat. У цьому файлі визначено користувачів та групи Nagios, у яких працюють екземпляри. У ньому вказуються шляхи до всіх файлів конфігурації окремих об'єктів, як-от команди, контакти, шаблони, хости, тощо.
  - **cgi.cfg** За замовчанням файл конфігурації CGI Nagios називається cgi.cfg. Він повідомляє CGI (Computer Generated Imagery. У буквальному перекладі воно означає «картинки, створені на комп' ютері».), де знайти основний конфігураційний файл. CGI зчитуватиме основний та основний конфігураційні файли для будь-яких інших даних, які можуть їм знадобитися. Він містить всю інформацію про користувачів та групи, а також їхні права та дозволи. Він також має шлях до всіх зовнішніх інтерфейсів Nagios.
  - **resource.cfg** У цьому файлі визначаються макроси $USERx$, які, у свою чергу, можуть використовуватися для визначення команд у файлах конфігурації хоста. Макроси $USERx$ корисні для зберігання конфіденційної інформації, такий як імена користувачів, паролі тощо. Вони також зручні для зберігання шляху до плагінів та обробників подій – якщо ви вирішите перемістити плагіни або обробники подій до іншого каталогу в майбутньому, ви можете просто оновити один або два макроси $USERx$ замість зміни великої кількості визначення команд. Файли ресурсів також можуть бути використані для зберігання директив конфігурації для зовнішніх джерел даних, таких як MySQL.
  - **objects/commands.cfg** Містить деякі приклади визначення команд, які можна використовувати у визначеннях хоста, служби та контакту. Ці команди використовуються для перевірки та моніторингу хостів та сервісів. Ці команди можливо запускати локально на Linux консолі.
  - **objects/contacts.cfg.** Містить інформацію про контакти та групи Nagios. За замовчуванням в одному контакті вже є адміністратор Nagios.
  - **objects/templates.cfg.** Файл містить деякі приклади шаблонів визначень об'єктів, на які посилаються інші визначення хоста, служби, контакту і т. д. в інших файлах конфігурації.
  - **objects/timeperiods.cfg.** Містить деякі приклади визначень часових періодів, які можна використовувати у визначеннях хоста, служб, контактів та залежностей.

- **Особливості Nagios Core.** Nagios – інструмент моніторингу з безліччю функцій.
  - Nagios Core – програмне забезпечення з відкритим кодом, безкоштовний для використання.
  - Потужний механізм моніторингу, який може масштабувати та керувати тисячами хостів та серверів.
  - Комплексна веб-панель моніторингу, що дозволяє побачити всі компоненти мережі та дані моніторингу.
  - Він має багатотентенантні можливості, коли кілька користувачів мають доступ до панелі моніторингу Nagios.
  - Він має архітектуру, що розширюється, яка може легко інтегруватися зі сторонніми додатками з декількома API.
  - Nagios має дуже активну та велику спільноту з більш ніж 1 мільйоном користувачів по всьому світу.
  - Система швидкого оповіщення відправляє оповіщення адміністраторам відразу після виявлення будь-якої проблеми.
  - Доступні плагіни для підтримки Nagios, плагіни з кодом користувача також можуть використовуватися з Nagios.
  - Він має хороший журнал та систему бази даних, що зберігає все, що відбувається в мережі з легкістю.
  - Функція запобіжного планування допомагає дізнатися, коли настав час оновити інфраструктуру.

- **Програми Nagios.** Nagios може застосовуватись для широкого спектру застосувань.
  - Моніторинг ресурсів вузла, як-от дисковий простір, системні журнали тощо.
  - Моніторинг мережевих ресурсів – http, ftp, smtp, ssh тощо.
  - Моніторинг файлів журналів безперервно, щоб ідентифікувати інфра-проблему.
  - Моніторинг Windows / Linux / Unix / веб-додатків та його стан.
  - Плагін Nagios (NRPE) може здійснювати віддалений моніторинг сервісів.
  - Паралельний запуск сервісних перевірок.
  - Можна використовувати тунелі SSH або SSL для віддаленого моніторингу.
  - Надсилати оповіщення / повідомлення електронною поштою, смс, месенджери про події.
  - Формувати рекомендації щодо оновлення IT-інфраструктури.

- **Nagios – Хости та сервіси.** Для розуміння принципів роботи Nagios, необхідно згадати про такі поняття як хости та сервіси. Хости та сервісні конфігурації є будівельними блоками Nagios Core.
  - **Хост** – це як комп'ютер; це може бути фізичний, або віртуальний пристрій.

- **Сервіс** – це служба або параметр, що використовує Nagios для перевірки певної інформації про хост.

Ви можете створити файл хоста чи сервісу в каталозі або підкаталозі сервера Nagios та визначити у ньому хост чи сервіс. Каталог, де система має «шукати» файли конфігурацій хостів та сервісів описується у nagios.cfg
Конфігураційні файли Nagos, на мій погляд, зроблені не найкращим чином. Їх імена можуть змінюватися від версії до версії. Наприклад, в одній з попередніх версій опис всіх серверів та їх сервісів рекомендувалося зберігати в одному файлі. Звичайно, що це не найкращий варіант, тому на кожен сервер – свій файл (простіше шукати і правити, якщо щось не так). На щастя, Nagios дозволяє змінювати структуру конфігів, з чим ми вже працювали у лабораторних роботах.

Zabbix

- **Що таке Zabbix** – це відкрите програмне забезпечення для моніторингу та управління мережею, серверами, додатками та різноманітними іншими IT-ресурсами. Як інструмент системного моніторингу з відкритим вихідним кодом, Zabbix дозволяє виявляти, вирішувати та передбачати проблеми в інфраструктурі комп'ютерних систем. За допомогою різноманітних сенсорів та збірників даних, Zabbix надає повний обзор стану IT-середовища, забезпечуючи користувачам засоби для ефективного управління та підтримки продуктивності. Інсталяція Zabbix виконується у ОС Linux з використанням LAMP-стеку (Linux, Apache, MySQL, PHP), LEMP-стеку (Linux, Nginx, MariaDB, PHP) або LAPP-стеку (Linux, Apache, PostgreSQL, PHP)

- **Історична довідка.**

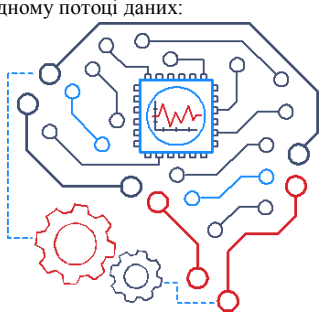
Zabbix був розроблений програмістом Олексієм Владишевим у 1998 році як проект внутрішнього програмного забезпечення. Через 3 роки, в 2001 був випущений публічно під ліцензією GPL. Більше трьох років минуло до виходу першої стабільної версії, 1.0, яка вийшла 2004. Версія 2.0 побачила світ у травні 2012. На сьогодні офіційну підтримку Zabbix виконує ZABBIX Company з реєстрацією у м. Рига, Латвія на чолі з «батьком» цієї системи моніторингу - Alexei Vladishev. Актуальна на сьогодні версія – 6.4.

- **Zabbix дозволяє збирати різні метрики**, і це далеко не повний перелік:
  - Мережеві пристрої
  - Хмарні сервіси, контейнери, віртуальні машини
  - Моніторинг операційних систем
  - Лог-файли
  - Бази даних
  - Додатки
  - Сервіси
  - IoT датчик
  - Моніторинг веб-сторінок
  - Моніторинг кінцевих точок HTTP/HTTPS
  - Підтримуючи всіх стандартних для галузі протоколів
  - Збір даних із зовнішніх кінцевих точок API

- **Вбудований високопродуктивний агент для всіх операційних систем та апаратних платформ**:
  - Незначний вплив на продуктивність
  - Доставка даних за принципом push або pull
  - Гнучкі інтервали планування
  - Легко розширюється за допомогою плагінів або зовнішніх скриптів
  - Несприйнятливість до проблем зв'язку (буферизація даних у пам'яті чи диску)
  - Зворотна сумісність для зручності оновлення
  - Доступ до Windows WMI та лічильників продуктивності
  - Можна створювати звіти на декількох серверах Zabbix для резервування та високої продуктивності
  - Надійний протокол TLS або загальний ключ шифрування для обміну даними
  - Розгортання агента Zabbix у вигляді пакета, використання MSI або програми встановлення командного рядка у Windows або розгортання попередньо скомпільованих двійкових файлів агента

- **Використання безагентного моніторингу із застосуванням будь-яких протоколів.** Zabbix підтримує цілу низку різних протоколів для віддаленого моніторингу сервісів:
  - Веб-моніторинг
  - Синтетичний моніторинг із підтримкою сценаріїв
  - Опитування та вилов SNMP (v1/2c/3)
  - Моніторинг програм Java
  - IPMI
  - Перевірки SSH/Telnet
  - Моніторинг ODBC
  - Перевірки ICMP та TCP
  - Легко розширюється за допомогою зовнішніх скриптів або плагінів
  - Збір даних із кінцевих точок HTTP
  - Підтримка протоколів Modbus та MQTT

- **Визначення гнучких порогів.** За допомогою Zabbix можна автоматично виявляти стан проблеми у вхідному потоці даних:
  - Високопродуктивне виявлення проблем у режимі реального часу
  - Гнучкі можливості визначення
  - Розділяйте стан вирішення проблем і самі проблеми
  - Декілька рівнів значущості
  - Аналіз вихідних даних
  - Захист від схлопування
  - Виявлення аномалій
  - Прогнозування проблем
  - Виявлені проблеми можна класифікувати за допомогою тегів для розумних оповіщень
  - Експорт виявлених проблем подій у режимі реального часу до сторонніх систем (Elastic, Splunk і т.д.)



Zabbix надає гнучкі розумні можливості визначення порогових значень. Хоча поріг спрацьовування може бути простим ("більше, ніж x"), користувач може використовувати всі можливості підтримуваних функцій та операторів для статистичного аналізу історичних даних .

- **Інсталяція Zabbix** може бути виконана декількома способами, зазвичай залежно від ваших потреб та технічного стеку. Ось кілька типових методів:
  - **Використання офіційних пакунків** - надає офіційні репозиторії для багатьох популярних дистрибутивів Linux, таких як Ubuntu, Debian, CentOS, Red Hat, та інших. Ви можете встановити Zabbix, використовуючи пакунки з цих репозиторіїв за допомогою менеджера пакунків вашої операційної системи.
  - **Використання Docker** для запуску Zabbix. Завантажте образи Zabbix з Docker Hub і запустіть контейнери за допомогою команд Docker. Це дозволяє швидко розгорнути тестове середовище Zabbix або навіть вирішити виробничні завдання.
  - **Ручне встановлення**, якщо ви бажаєте більшої контролю над процесом інсталяції або плануєте використовувати специфічні конфігурації. Це зазвичай включає встановлення всіх необхідних компонентів, таких як веб-сервер, база даних та інші залежності, а потім налаштування Zabbix через веб-інтерфейс або конфігураційні файли.
  - **Використання Zabbix Appliance** - готових віртуальних машин або апліансів, які можуть бути завантажені та запущені на віртуальних платформах, таких як VMware або Hyper-V. Це швидкий спосіб отримати робочий екземпляр Zabbix без необхідності вручну налаштовувати середовище.
  - **Використання Zabbix Cloud** - хмарні рішення, такі як Zabbix Cloud, який надає можливість встановлення та налаштування Zabbix на серверах, які управляються Zabbix.

Більшість варіантів інсталяції Zabbix, зазвичай, є безкоштовними або мають безкоштовні версії. Проте, є деякі відмінності в ліцензуванні та доступних функціях в залежності від обраного методу та варіанту використання:

- **Відкрита версія (Open Source)**: Зазвичай, встановлення Zabbix з використанням офіційних пакунків або ручним способом використовує відкриту версію Zabbix, яка є безкоштовною і має відкритий вихідний код. Це означає, що ви можете використовувати та модифікувати програмне забезпечення безкоштовно, але деякі розширені функції можуть бути доступні тільки в комерційних версіях.
- **Комерційні версії**: Zabbix також пропонує комерційні версії з додатковими функціями, підтримкою та послугами, такими як розширена моніторингова функціональність, технічна підтримка, консультування, навчання та інші. Вартість цих версій може варіюватися в залежності від обсягу функціональності та рівня підтримки.
- **Хмарні рішення**: Хмарні рішення можуть мати різні цінові плани, від безкоштовних пробних версій до платних підписок. Ціни можуть залежати від обраного рівня функціональності, обсягу моніторингу, кількості ресурсів, що моніторяться та інших факторів.

Рис. 05.03. Приклади візуалізації Zabbix

- **Потужна візуалізація даних.** Zabbix відображає зібрані дані різними можливими способами. Можливо визначити панелі інструментів на основі віджетів, що відображають необхідну інформацію:
  - Великий вибір різноманітних віджетів
  - Просте розміщення та масштабування віджетів за допомогою функції drag and drop
  - Кожен віджет легко налаштовується відповідно до ваших потреб.
  - Відображення метрик, проблем , інфраструктури та географічних карт на панелі управління
  - Відображення поточної інформації про SLA бізнес-сервісів на панелі керування
  - Дозволяє аналізувати та зіставляти показники за допомогою графіків, визначати користувацькі графіки.

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

- **Прив'язка об'єктів моніторингу до інтерактивної географічної карти.**
  - Вибір кількох постачальників геокарт
  - Відображення географічного огляду вашого середовища на панелі інструментів Zabbix
  - Доступ до будь-якого з об'єктів моніторингу з геокарти
  - Угруповання об'єктів моніторингу у кластер на географічній карті
  - Відстеження стану окремих об'єктів моніторингу чи всього кластера загалом

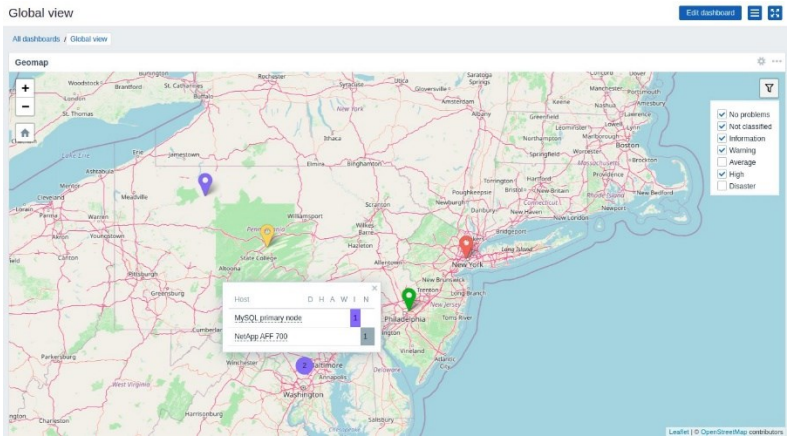
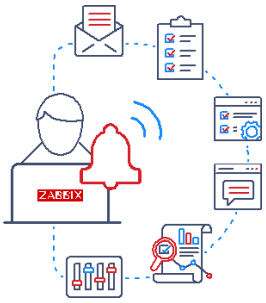


Рис. 05.04. Прив'язки об'єктів моніторингу Zabbix до інтерактивної карти.

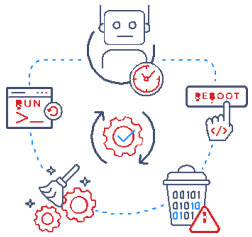
- **Проактивне реагування з прогнозуванням тенденцій.** Незважаючи на те, що для виявлення проблем добре мати пороги , ефективніше реагувати на проблеми проактивно. Функції прогнозування Zabbix допоможуть вам досягти цієї мети:
  - Прогнозування значення для раннього оповіщення
  - Прогнозування часу, що залишився до досягнення порогового значення проблеми

- **Отримання повідомлень про критичні проблеми**
  - **Канали обміну повідомленнями.** Використовується кілька каналів обміну повідомленнями для оповіщення відповідальної особи або осіб про різні події, що відбуваються у середовищі:
    - Email
    - SMS для надійних оповіщень
    - Онлайн SMS-шлюзи
    - Платформи для спілкування :
      - ✓ Slack
      - ✓ MS Teams
      - ✓ Telegram
      - ✓ Express.ms
      - ✓ Rocket.chat
      - ✓ And more
    - Вебхуки для інтеграції із зовнішніми системами обміну повідомленнями, ITSM та системами тикетингу



  - **Налаштування оповіщень.** Визначення різних повідомлень для різних каналів надсилання повідомлень. Можливо використовувати шаблони за замовчуванням або створити та налаштувати свій власний шаблон.
    - Налаштування повідомлення залежно від типу проблеми та ролі одержувача
    - Доповнення повідомлення будь-якою інформацією про час виконання та інвентар .
    - Надсилання запланованих PDF звітів для поглибленого та довгострокового аналізу даних

- **Zabbix може вирішувати деякі проблеми автоматично.**
  За допомогою Zabbix можливо не тільки отримувати повідомлення про проблему, але й автоматично усувати її.
  Для усунення проблеми може бути виконаний сценарій або команда:
  - Перезапуск сервісу
  - Керування хмарними ресурсами
  - Виконання автоматичної зміни масштабу ресурсів
  - Виконання будь-якої іншої логіки користувача



- **Використання єдиного інтерфейсу для всієї інфраструктури**
  - Веб-інтерфейс Zabbix надає безліч способів представлення візуального огляду IT-середовища:
    - Багатосторінкові панелі на основі віджетів
    - Просте розміщення віджетів за допомогою функції drag and drop
    - Налаштування інтервалів автоматичного оновлення панелі керування
    - Можливість клонувати існуючу панель
    - Приватні та публічні панелі
  - Кожен із елементів панелі керування надзвичайно гнучкий, підтримує безліч режимів перегляду, фільтрів та враховує права доступу користувача:

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

- Фільтрування та відображення тільки необхідних даних
- Налаштування віджетів для відображення даних на різних рівнях деталізації
- При відображенні даних на панелі приладів дотримуються права доступу користувачів

➢ Збирання та відображення інформації про інвентаризацію
- Використання зібраних метрик для надання інвентаризаційної інформації про хости
- Поєднання вибірки власних даних інвентаризації з Zabbix API для надання додаткових даних інвентаризації
- Отримання огляду спільного інвентарю, групуванням хостів
- Надання та відстеження координат об'єктів моніторингу на геокарті
- Динамічне оновлення існуючої інформації про інвентар на основі зібраних метрик

❖ **Просте розгортання Zabbix у інфраструктурі**

➢ Моніторинг із коробки для провідних виробників програмного та апаратного забезпечення.
- Cisco
- HPE
- Microsoft
- IBM
- VMware
- Meraki
- Juniper
- F5
- І багато інших

➢ Інтеграція Zabbix з існуючими системами.
- Моніторинг Docker контейнерів
- Web сервери - IIS, Apache, Nginx та інші
- Бази даних , такі як MySQL, PostgreSQL, Microsoft SQL, MongoDB та інші
- Моніторинг будь-якої операційної системи – Linux, Windows, Solaris, BSD, MacOS та інших
- Хмарні сервіси , такі як AWS, Amazon cloud, Google cloud та інші
- Сервіси IP-телефонії

Nagios vs Zabbix

Ті, хто давно працює у світі IT – вважають Nagios “стандартом де-факто” у сфері моніторингу з відкритим вихідним кодом. І це правда, тому що ця компанія, яка перша почала застосовувати моніторинг правильно. До Nagios існували програми, але вони були дуже дилетантськими. Так само існували окремі якісні інструменти, які були гарними лише для конкретного завдання. Перша версія Nagios датується минулим століттям: 1999 роком. Минуло багато років, і технологія розвивалася: Nagios розвивався через екосистему сторонніх доповнень, якими намагалися доповнити функції, що бракувало системі. Ці розробки розширили можливості налаштування Nagios та перетворили його на універсальний інструмент.
Zabbix з'явився у 2001 році. Це повноцінна розробка, а не відгалуження Nagios, і її головна особливість у тому, що вона має більш цілісний погляд на моніторинг, що охоплює продуктивність, а не тільки стан, оскільки це - один із найістотніших недоліків Nagios. Zabbix також має систему управління WEB, що дозволяє керувати ним централізовано, без громіздких конфігураційних файлів, як це було у випадку з Nagios.
Zabbix та Nagios вимагають встановлення безлічі плагінів, щоб бути ефективними та пропонувати повний набір функціональних можливостей. Zabbix не має «офіційної» бібліотеки плагінів для спільноти, хоча має список OID для SNMP-запитів. Крім того, він не пропонує можливості роботи з інструментами Enterprise, такими як Oracle, Exchange, Active Directory та іншими ядрами, на відміну від Nagios, який завдяки своїм розширенням це «вміє».
Nagios має на 100% відкриту гігантську бібліотеку, але за більшістю складових цієї бібліотеки немає компанії, що її підтримує, або дбає про її актуальність. Це один з його основних недоліків.
Що стосується спільноти користувачів, то знову, найбільшою спільнотою є Nagios, з тієї простої причини, що він з'явився першим. Щодо кількості відгалужень та форків Nagios вже згадувалось у цій лекції. Велика кількість відгалужень Nagios призводить до деякої хаотичності екосистеми, коли доходить до впровадження плагінів або інструментів під одну задачу, розроблених різними розробниками. Кожна гілка має свою філософію і згодом робить її сумісність з іншими гілками та з батьківським проектом (Nagios) «через танці з бубном». Але також треба відмітити, що така різноманітність це палиця з двома кінцями. Вона дозволяє знайти плагін під потреби користувача, а не використовувати лише те, що прийшло «з коробки».

Prometheus

❖ **Що таке Prometheus.** Це набір інструментів для моніторингу та попередження систем із відкритим вихідним кодом, створений у SoundCloud. З моменту створення в 2012 році багато компаній і організацій впровадили Prometheus, і проект має активну спільноту розробників і користувачів . На сьогодні, це окремий проект із відкритим кодом, який підтримується незалежно від будь-якої компанії. Щоб підкреслити це та уточнити структуру управління проектом, Prometheus приєднався до Cloud Native Computing Foundation у 2016 році як другий розміщений проект після Kubernetes.

Prometheus збирає та зберігає свої показники як дані часових рядів, тобто інформація про показники зберігається з міткою часу, коли вона була записана, поряд із необов’язковими парами ключ-значення, які називаються мітками.

❖ **Основні риси Prometheus.**

➢ **Багатовимірна модель даних із даними часових рядів, ідентифікованих назвою метрики та парами ключ/значення.**
До речі, про цю принципову особливість Prometheus ми згадували у другій лекції. Prometheus принципово зберігає всі дані як часові ряди: потоки значень із мітками часу, що належать до того самого показника та того самого набору мічених параметрів. Окрім збережених часових рядів, Prometheus може генерувати тимчасові похідні часові ряди в результаті запитів.

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

- **PromQL, гнучка мова запитів для використання цієї розмірності.** PromQL (Prometheus Query Language) дозволяє користувачеві вибирати та агрегувати дані часових рядів у реальному часі. Результат виразу можна або відобразити у вигляді графіка, переглянути як табличні дані в браузері виразів Prometheus, або використати зовнішніми системами через HTTP API .
- **Відсутність залежності від розподіленого сховища**; окремі серверні вузли є автономними
- **Pull model.** Збирання часових рядів відбувається через модель вилучення через HTTP. Prometheus сама відправляє запити на різні джерела даних і отримує від них потрібну інформацію. Це може бути, наприклад, інформація про стан різних систем або програм, що моніторяться. Основна перевага такого підходу полягає в тому, що Prometheus може ефективно збирати дані з різних джерел, що дозволяє моніторити різні системи або сервіси у вашому середовищі. Крім того, цей підхід надає гнучкість у налаштуванні та спрощує процес системного моніторингу, оскільки ви можете легко налаштувати, які дані ви хочете збирати та звідки.
- **Prometheus Pushgateway.** Проштовхування часових рядів підтримується через проміжний шлюз. Час від часу Prometheus потрібно стежити за компонентами, які неможливо очистити. Prometheus Pushgateway дає змогу переміщати часові ряди з короткотривалих пакетних завдань на рівні обслуговування до проміжного завдання, яке Prometheus може вилучити. У поєднанні з простим текстовим форматом експозиції Prometheus це дозволяє легко інструментувати навіть сценарії оболонки без клієнтської бібліотеки.
- **Цілі виявляються за допомогою виявлення служби або статичної конфігурації.** Існує два основних методи виявлення цілей у Prometheus **Service Discovery** та **статична конфігурація**:
  - **Виявлення служби (Service Discovery).** Prometheus може використовувати вбудовані механізми виявлення служб для автоматичного визначення цілей. **Service Discovery** в Prometheus полягає у виявленні автоматично цілей моніторингу - це можуть бути різні системи, сервіси або компоненти, які ви хочете включити до моніторингу. Prometheus може використовувати вбудовані механізми виявлення служб для цього. Основні методи виявлення служб включають:
    1. **DNS виявлення**. Prometheus може використовувати DNS для автоматичного виявлення служб. Вказуються шаблони імен хостів, які використовуються для моніторингу, і Prometheus автоматично виявить ці хости.
    2. **Consul або Kubernetes**. В середовищі, де використовуються інструменти, такі як Consul (інструмент для розподіленої системи конфігурації, виявлення служб та управління цими службами) або Kubernetes (система оркестрації контейнерів, що призначена для автоматизації розгортання, масштабування та керування додатками, що розгорнуті у контейнерах), Prometheus може використовувати їхні API для автоматичного виявлення цілей моніторингу. Наприклад, в Kubernetes Prometheus може автоматично виявляти всі сервіси, що запущені у кластері.
    3. **EC2 Service Discovery**. В середовищі AWS, Prometheus може автоматично виявляти цілі моніторингу, які базуються на екземплярах EC2, використовуючи дані, які надаються сервісом EC2 та іншими AWS-специфічними інструментами.
    4. **Azure Service Discovery**. Аналогічно, для середовищ Azure існують механізми виявлення, які можуть бути використані для автоматичного визначення цілей моніторингу на базі інформації про ресурси та служби, що надаються платформою Azure.
    5. **GCP Service Discovery**. Google Cloud Platform (GCP) також має вбудовані інструменти для виявлення служб, які можуть бути використані Prometheus для автоматичного визначення цілей моніторингу в середовищі GCP.
    6. **OpenStack Service Discovery**. В середовищі OpenStack, існують різні інструменти та API, які можуть бути використані для автоматичного виявлення служб і ресурсів для моніторингу.

    Перелічені механізми виявлення служб дозволяють автоматизувати процес додавання та видалення цілей моніторингу, що спрощує управління системою моніторингу, особливо в великих та динамічних середовищах.

  - **Статична конфігурація.** Користувачі можуть вказати цілі (адреси систем або сервісів, де розташовані експортери) явно в конфігураційному файлі prometheus.yml. Цей підхід використовується для статичного визначення цілей без виявлення служби, що може бути корисним у випадках, коли сервери відомі заздалегідь, або коли потрібна точна конфігурація лише обраних хостів.
- **Підтримка кількох режимів побудови графіків і інформаційних панелей.** Два найпоширеніших інструменти для цього - Grafana та Prometheus's own built-in Expression Browser.
  - **Grafana** - це інструмент візуалізації та моніторингу, який може використовувати дані, зібрані Prometheus. Він надає широкий спектр можливостей для створення графіків, інформаційних панелей та панелей моніторингу за допомогою різних типів даних, функцій та налаштувань візуалізації. Grafana дозволяє створювати діаграми, графіки, теплові карти, стовпчаті та кругові діаграми та інші типи візуалізацій для аналізу та відображення даних, що збираються Prometheus.
  - Вбудований **Expression Browser** в Prometheus надає можливість візуалізувати дані непрямо в інтерфейсі самого Prometheus. Він використовується для відображення графіків та відслідковування значень метрик прямо в браузері. Це зручний спосіб швидко переглянути дані та здійснити базовий аналіз без необхідності налаштування додаткових інструментів візуалізації.

❖ **Архітектура Prometheus.** Екосистема Prometheus складається з кількох компонентів, багато з яких є необов'язковими.
- **Головний сервер Prometheus** , який збирає та зберігає дані часових рядів
- **Клієнтські бібліотеки** для інструментування програмного коду
- **Push-шлюз** для підтримки короткострокових завдань
- **Експортери** спеціального призначення для таких сервісів, як HAProxy, StatsD, Graphite тощо.
- **Менеджер сповіщень** для обробки сповіщень
- **Засоби підтримки**

Більшість компонентів Prometheus написані на Go , що полегшує їх створення та розгортання як статичних двійкових файлів.

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

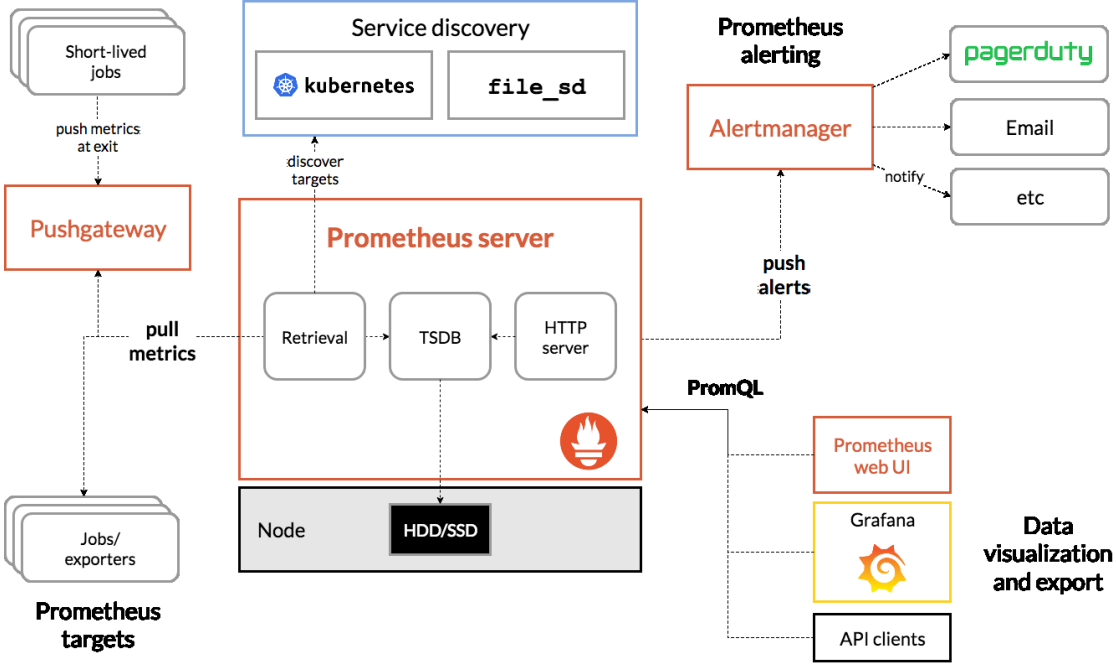


Рис. 05.05. Архітектура Prometheus і деякі компоненти його екосистеми.

Prometheus збирає показники з інструментальних завдань безпосередньо або через проміжний push-шлюз для короткострокових завдань. Він зберігає всі зібрані зразки локально та запускає правила для цих даних, щоб агрегувати та записувати нові часові ряди з наявних даних або створювати сповіщення. Для візуалізації зібраних даних можна використовувати Grafana або інші споживачі API.
Незважаючи на те, що Prometheus можна використовувати для побудови графіків конкретних запитів, це не повноцінна інформаційна панель і її потрібно підключити до Grafana для створення інформаційних панелей. Це є суттєвим недоліком системи через додаткову складність налаштування.

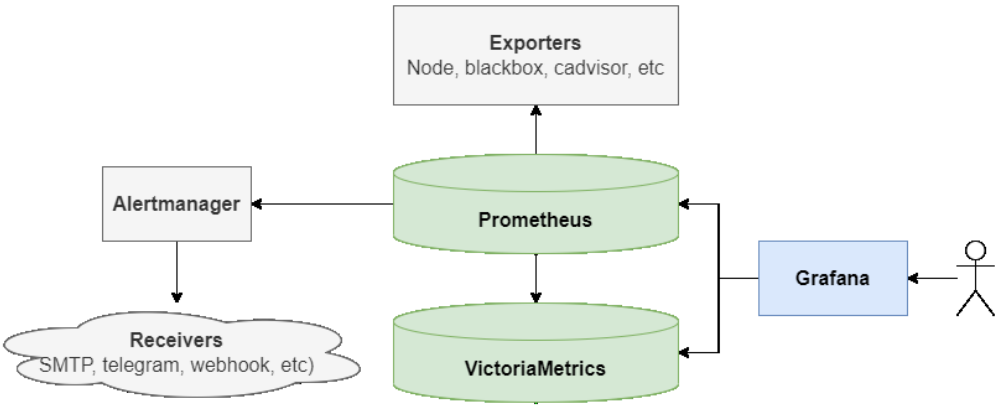


Рис. 05.05. Схема взаємодії компонентів Prometheus.

- Розглянемо докладніше схему взаємодії компонентів системи моніторингу з урахуванням Prometheus. Базова конфігурація складається з трьох компонентів екосистеми :
  - **Експортери (exporters).** Експортер збирає дані та повертає їх у вигляді набору метрик. Експортери поділяються на офіційні (написані командою Prometheus) та неофіційні (написані розробниками різного програмного забезпечення для інтеграції з Prometheus). За необхідності є можливість писати свої експортери та розширювати наявні додатковими метриками.
  - **Prometheus.** Отримує метрики від експортерів і зберігає їх у БД часових рядів. Підтримує потужну мову запитів PromQL (Prometheus Query Language) для вибірки та агрегації метрик. Дозволяє будувати прості графіки та формувати правила повідомлень (alerts) на основі виразів PromQL для відправки через Alertmanager
  - **Alertmanager.** Обробляє повідомлення від Prometheus і розсилає їх. За допомогою механізму приймачів (receivers) реалізовано інтеграцію з поштою (SMTP), Telegram, Slack та ін. системами, а також надсилання повідомлень у власний API за допомогою вебхуків (webhook). Таким чином, базова конфігурація дозволяє збирати дані, писати складні запити та надсилати повідомлення на їх основі. Однак, по-справжньому потенціал Prometheus розкривається при додаванні двох додаткових компонентів (або як мінімум одного – Grafana)
  - **VictoriaMetrics.** Отримує метрики з Prometheus за допомогою remote write . Підтримує мову запитів MetricsQL , синтаксис якого сумісний із PromQL. Надає оптимізоване споживання ресурсів зберігання даних і високопродуктивне виконання запитів. Ідеально підходить для довготривалого зберігання великої кількості метрик
  - **Grafana.** Надає засоби візуалізації та додаткового аналізу інформації з Prometheus та VictoriaMetrics. Є приклади дашбордів практично під будь-які завдання, які за потреби можна легко доопрацювати. Створення власних дашбордів також інтуїтивно (зрозуміло, крім деяких тонкощів) – достатньо знати основи PromQL / MetricsQL Де-факто використання Grafana разом з Prometheus вже стало стандартом, у той час як додавання в конфігурацію VictoriaMetrics безумовно опціональне і необхідне для високонавантажених систем.

- **Інсталяція Prometheus.**
  - З використанням **офіційних пакунків** для різних операційних систем, таких як Linux, Windows та macOS. Ви можете завантажити ці пакунки з веб-сайту Prometheus та встановити їх за допомогою стандартних процедур для вашої операційної системи.
  - **Встановлення** Prometheus **у контейнері Docker** є популярним методом, оскільки це дозволяє легко запускати та масштабувати Prometheus на різних платформах. Ви можете завантажити образ Prometheus з Docker Hub і запустити контейнер за допомогою команд Docker.
  - Prometheus може бути розгорнута **в середовищі Kubernetes як контейнеризований додаток**. Це дозволяє автоматизувати розгортання, масштабування та управління Prometheus за допомогою інструментів Kubernetes.
  - Як альтернативу, ви можете встановити **Prometheus вручну**, завантаживши вихідний код з GitHub та компілюючи його на своєму сервері. Цей метод може бути корисним для більш гнучкого налаштування або для встановлення Prometheus на платформи, для яких немає офіційних пакунків.

  Платформи, на яких можна встановити Prometheus, включають усі основні операційні системи, такі як Linux, Windows та macOS. Крім того, вона також може бути встановлена на хмарних платформах, таких як Amazon Web Services (AWS), Google Cloud Platform (GCP) та Microsoft Azure, а також на власних серверах або віртуальних машинах.

- **Коли підходить?**

  Prometheus добре працює для запису будь-яких чисто числових часових рядів. Він підходить як для машиноцентричного моніторингу, так і для моніторингу високодинамічних сервіс-орієнтованих архітектур. У світі мікросервісів його підтримка збору багатовимірних даних і запитів є особливою перевагою.

  Prometheus розроблено для надійності, щоб бути системою, до якої ви звертаєтеся під час збою, щоб дозволити вам швидко діагностувати проблеми. Кожен сервер Prometheus є автономним і не залежить від мережевого сховища чи інших віддалених служб. На цю систему моніторингу можете повністю покластися, коли інші частини вашої інфраструктури зламано, і не потрібно налаштовувати розгалужену інфраструктуру, щоб використовувати його.

- **Коли не підходить?**

  Prometheus цінує надійність. Завжди можливо переглянути доступну статистику щодо системи, навіть за умов збою. Якщо необхідна 100% точність, наприклад для виставлення рахунків за запитом, Prometheus не є гарним вибором, оскільки зібрані дані, швидше за все, не будуть достатньо детальними та повними. У такому випадку краще використовувати якусь іншу систему для збору й аналізу даних для виставлення рахунків.

MS SCOM

- **Що таке SCOM?** System Center Operations Manager (SCOM) — це кросплатформна система моніторингу центру обробки даних для операційних систем і гіпервізорів . Він використовує єдиний інтерфейс, який показує інформацію про стан, працездатність і продуктивність комп'ютерних систем. Він також надає сповіщення, створені відповідно до певної доступності, продуктивності, конфігурації або ситуації безпеки. Він працює з Microsoft Windows Server і хостами на базі Unix/Linux .

- **Історія MS SCOM.**
  - **SeNTry ELM**. Продукт починався як система керування мережею SeNTry ELM, розроблена британською компанією Serverware Group plc .
  - **Enterprise Event Manager.** У червні 1998 року права інтелектуальної власності придбала компанія Mission Critical Software, Inc., яка перейменувала продукт у Enterprise Event Manager. Mission Critical повністю переписав продукт, назвавши нову версію OnePoint Operations Manager (OOM).
  - **Microsoft Operations Manager (MOM).** Mission Critical Software об'єдналася з NetIQ на початку 2000 року та продала права на продукт Microsoft у жовтні 2000 року. Пізніше його було перейменовано на Microsoft Operations Manager (MOM) — у 2003 році Microsoft почала роботу над наступною версія MOM: вона називалася Microsoft Operations Manager 2005 і була випущена в серпні 2004 року. Service Pack 1 для MOM 2005 було випущено в липні 2005 року з підтримкою Windows 2003 Service Pack 1 і SQL Server 2000 Service Pack 4 Крім того, необхідно було підтримувати SQL Server 2005 для компонентів оперативної та звітної бази даних. Розробка наступної версії — на той час її кодова назва «MOM V3» — почалася в 2005 році.
  - **System Center Operations Manager (SCOM).** Microsoft перейменувала продукт на System Center Operations Manager і випустила System Center Operations Manager 2007 у березні 2007 року. System Center Operations Manager 2007 було розроблено на основі свіжого коду, і, хоча він має спільні риси з Microsoft Operations Manager, він не є оновленням попередніх версій.
  - **System Center Operations Manager 2007 R2 (SCOM 2007 R2).** У травні 2009 року System Center Operations Manager 2007 мав так званий випуск «R2» — загальним удосконаленням була підтримка крос-платформи для серверів UNIX і Linux. Замість публікації окремих пакетів оновлень, виправлення помилок у продукті після випуску System Center Operations Manager 2007 R2 було випущено у формі так званих накопичувальних оновлень (CU).
  - **System Center Operations Manager 2012 R2 (SCOM 2012 R2).** Додана підтримка для операційних систем Windows Server 2012 R2 та включає ряд функцій для моніторингу та управління інфраструктурою.
  - **System Center Operations Manager 2016 (SCOM 2016).** Принесла нововведення, такі як покращення у масштабованості, підтримка Windows Server 2016 та інтеграція з Microsoft Operations Management Suite (OMS).
  - **System Center Operations Manager 2019 (SCOM 2019)**. SCOM 2019 принесла ряд покращень, у тому числі в області ефективності, безпеки та інтеграції з хмаровими сервісами.
  - **System Center Operations Manager 2022 (SCOM 2022).** Остання версія на цей момент. Суттєво відрізняється можливостями від попередньої версії: оновлений моніторинг Office 365 та MS Azure, підтримка Server 2022, інтеграція Microsoft Teams. Це вже не просто «локальний» моніторинг із новими функціями, зосередженими на Azure Stack HCI а повна підтримка хмари, що надається як послуга. Крім того додана підтримка агностичних пакетів керування версіями у вигляді Management Pack (MP).
    Створені нові вбудовані ролі.

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

- Роль Адміністратор із правами лише на читання . Надає всі дозволи на читання в Operations Manager, включаючи звіти.
- Можливість створення функцій користувачів, що налаштовуються, з конкретними дозволами. Тепер Управління агентами підтримує дві нові підкатегорії - Розгортання агентів та Відновлення агентів , які неявно надають дозволи на Очікуючі дії агента .
- Додано профіль Повноважного адміністратора – це адміністратор з правами лише на читання, за винятком створення звітів. Може бути створена налаштована роль користувача з повноважним адміністратором як базовий профіль і додано до неї один або кілька дозволів з доступних категорій.

❖ **Архітектура System Center Operations Manager (SCOM)**

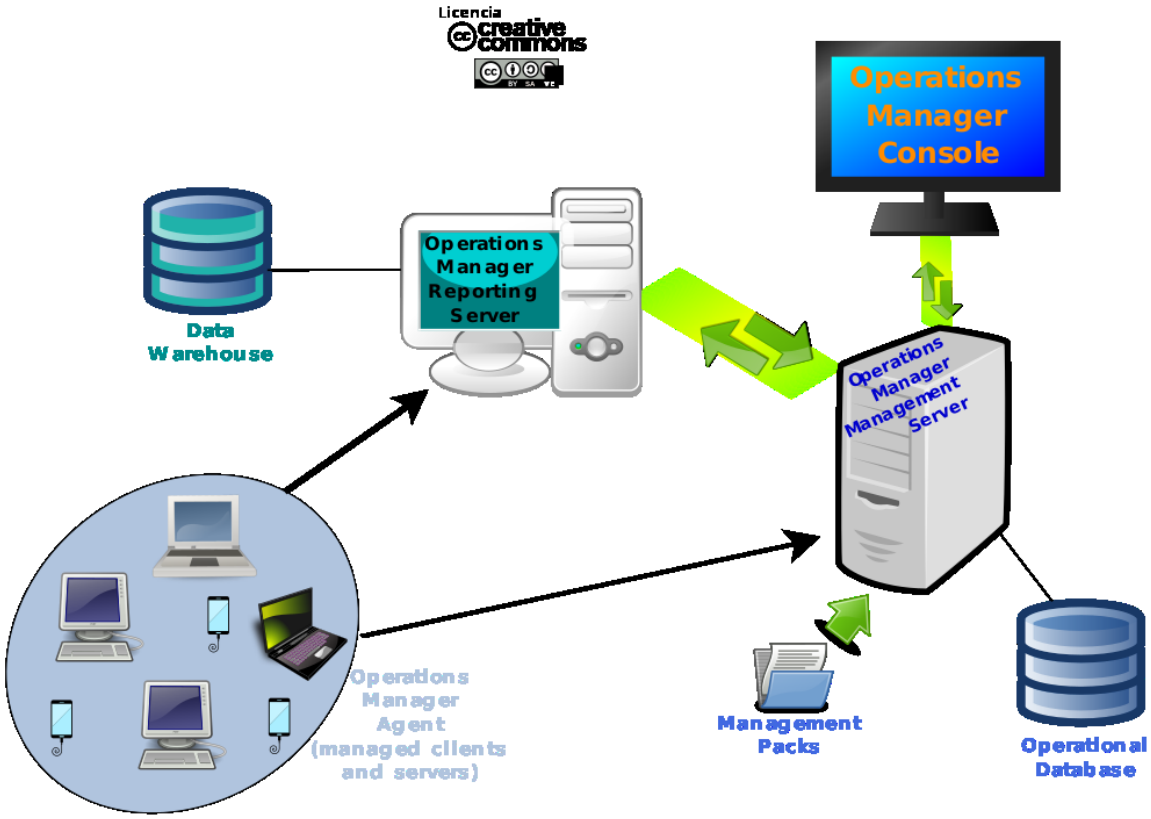


Рис. 05.06. System Center Operations Manager: основні компоненти продукту.

Основна ідея полягає в тому, щоб розмістити частину програмного забезпечення, агента , на комп'ютері, який слід контролювати.
Агент спостерігає за кількома джерелами на цьому комп'ютері, включаючи журнал подій Windows, для певних подій або попереджень, створених програмами, які виконуються на контрольованому комп'ютері. Після появи та виявлення попередження агент пересилає попередження на **центральний сервер SCOM**. Ця серверна програма SCOM підтримує базу даних **Operation DataBase**, яка містить історію сповіщень. Сервер SCOM застосовує правила фільтрації до сповіщень у міру їх надходження; правило може ініціювати певне сповіщення людині, наприклад повідомлення електронною поштою чи месенжером, створити запит на підтримку мережі або запустити інший робочий процес, спрямований на усунення причини сповіщення відповідним чином.

➢ **Бази даних SCOM.** Microsoft System Center Operations Manager використовує дві бази даних для різних цілей:

- **Операційна база даних (Operations Database)** використовується для зберігання інформації про реальний час моніторингу та подій, що створюються агентами SCOM. Вона містить дані про стан систем та служб, що перебувають під наглядом SCOM, а також історію сповіщень та подій, що виникають на контрольованих комп'ютерах. Ця база даних дозволяє адміністраторам відслідковувати стан систем та вчасно реагувати на можливі проблеми.
- **Сховище даних (Data Warehouse)** використовується для зберігання історичних даних моніторингу, аналізу та звітності. Воно забезпечує можливість аналізу трендів, прогнозування проблем та відслідковування ефективності системи з часом. Даний склад даних може бути використаний для створення звітів, аналізу працездатності та моніторингу великих обсягів даних.

Перенесення даних між цими базами даних може відбуватися залежно від налаштувань і потреб моніторингу. Наприклад, історичні дані можуть бути перенесені з операційної бази даних в сховище даних для подальшого зберігання та аналізу. Нові дані можуть також бути відображені в операційній базі даних та одночасно перенесені в сховище даних для подальшого аналізу та звітності.

SCOM використовує термін «пакет керування» для позначення набору правил фільтрації, характерних для деяких контрольованих програм. У той час як корпорація Майкрософт та інші постачальники програмного забезпечення надають пакети керування для своїх продуктів, SCOM також забезпечує створення спеціальних пакетів керування. Хоча роль адміністратора потрібна для інсталяції агентів, налаштування комп'ютерів, що контролюються, і створення пакетів керування, право просто переглядати список останніх попереджень можна надати будь-якому дійсному обліковому запису користувача.
Кілька серверів SCOM можна об'єднати разом для моніторингу кількох мереж через межі логічного домену Windows і фізичних мереж. У попередніх версіях Operations Manager веб-служба використовувалася для підключення кількох окремо керованих груп до центрального розташування. Починаючи з Operations Manager 2007, веб-служба більше не використовується. Замість цього використовується пряме TCP-з'єднання з використанням порту 5723 для цих зв'язків.

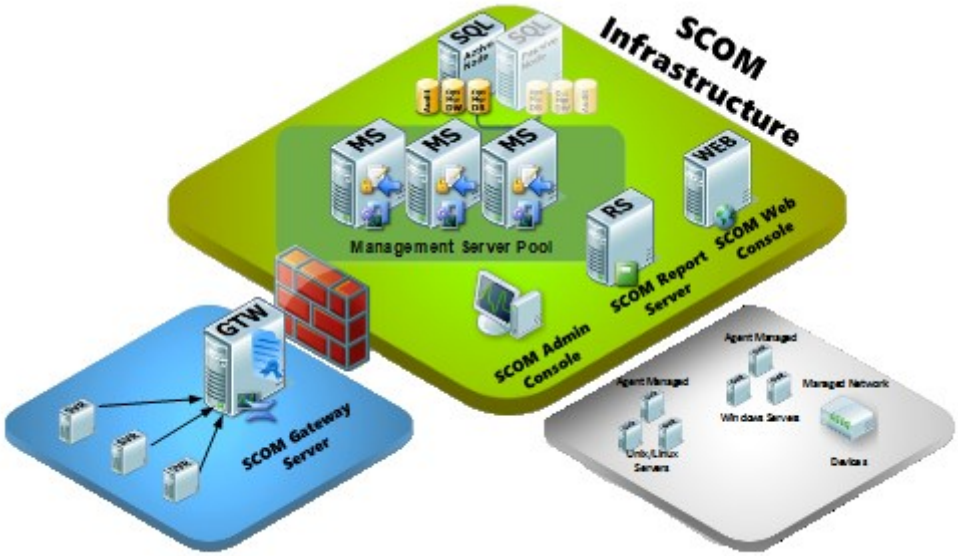


Рис. 05.07. Інфраструктура System Center Operations Manager.

- **Інтеграція з Microsoft Azure.** Для моніторингу серверів, які працюють у Microsoft Cloud Infrastructure Azure, можна ввімкнути джерела даних Log Analytics, які збирають і надсилають свої дані на локальні сервери управління SCOM. З листопада 2020 року корпорація Майкрософт оголосила про запуск SCOM у вигляді повністю керованого хмарного інстансу у своєму середовищі Azure під кодовою назва «Aquila» з широким функціоналом:
  - Спостереження показників витрат і ефективності майже в реальному часі
  - Автоматизовані дії для оптимізації витрат, включаючи графіки запуску/вимкнення ресурсу
  - Фінансові сфери для управління проектами, контролю бюджету та можливості відкликання платежів
  - Портал самообслуговування з керуванням Role-Based Access Control, який дозволяє користувачам самостійно керувати своїми правами доступу до різних ресурсів або функцій в системі на основі їхніх ролей або політик безпеки.
  - Прогнози витрат на основі ML для планування потужності
  - Обмін звітами на основі URL-адреси на платформі
- **Командна оболонка.** Починаючи з Operations Manager 2007, продукт включає розширюваний інтерфейс командного рядка під назвою The Command Shell , який є налаштованим екземпляром Windows PowerShell , що забезпечує інтерактивний доступ до даних і операцій Operations Manager на основі сценаріїв.

- **Пакет керування.** SCOM можна розширити, імпортувавши пакети керування - Management Pack (MP), які визначають, як SCOM відстежує системи. За замовчуванням SCOM відстежує лише базові служби, пов’язані з ОС, але нові MP можна імпортувати для моніторингу таких служб, як сервери SQL, SharePoint, Apache, Tomcat, VMware і SUSE Linux. Багато продуктів Microsoft мають MP, які випускаються разом із ними, і багато компаній, що не належать до Microsoft, також пишуть MP для своїх власних продуктів. У той час як достатня кількість IT-інфраструктури контролюється за допомогою поточних доступних MP, нові MP можуть створюватися кінцевими користувачами, щоб контролювати те, що ще не охоплено. Створення пакета керування можливе за допомогою System Center Operations Manager 2007 R2 Resource Kit, Visual Studio with Authoring Extensions і Visio MP Designer.

- На початку лекції ми розглядали безкоштовні системи моніторингу. Які ж **бонуси дає використання "платного" System Center Operations Manager (SCOM)** у порівнянні з іншими системами моніторингу:
  - **Широкий функціонал Microsoft**: SCOM інтегрується з іншими продуктами Microsoft і забезпечує комплексний моніторинг для об'єктів, таких як операційні системи, сервери баз даних, та додатки, що використовують технології Microsoft.
  - **Глибока інтеграція з іншими продуктами Microsoft**: SCOM взаємодіє із іншими компонентами Microsoft System Center, такими як System Center Configuration Manager (SCCM) та System Center Virtual Machine Manager (SCVMM), що дозволяє вам отримати більше комплексний погляд на інфраструктуру.
  - **Підтримка для гетерогенних середовищ**: SCOM підтримує моніторинг не тільки продуктів Microsoft, але й інших вендорів, надаючи можливість створювати агенти та пакети для моніторингу різноманітних технологій.
  - **Можливості управління та автоматизації**: SCOM дозволяє не лише моніторити, а й використовувати правила та робочі процеси для автоматизації відповідей на події та проблеми.
  - **Інтеграція з Azure та хмарними рішеннями**: SCOM підтримує інтеграцію з Microsoft Azure, що дозволяє розширювати моніторинг на хмарові сервіси та ресурси.
  - **Широкий спектр Management Packs**: Microsoft та інші вендори надають різноманітні Management Packs для SCOM, що спрощує налаштування моніторингу для різних технологій та додатків.

  Звісно, вибір між SCOM та іншими системами моніторингу також залежить від конкретних потреб, інфраструктури та вартості власності.

<u>Практичне використання інструментів для моніторингу системних ресурсів та додатків</u>

- **Налаштування моніторингу ресурсів серверів.** Моніторинг ресурсів серверів є критичним аспектом у забезпеченні надійності та ефективності IT-інфраструктури. У цьому контексті, розглянемо процес налаштування моніторингу для основних ресурсів сервера: центральний процесор (CPU), оперативна пам'ять (RAM), пристрої зберігання (диск), та мережеві підключення.
  - **Моніторинг CPU, пам'яті, диска та мережі**

SNM. #3. Інструменти моніторингу та аналізу даних ***
Системний *та мережевий моніторинг. Лекція #5.* Інструменти та технології системного моніторингу.

- **CPU.** Використовуйте моніторингові інструменти або агенти для вимірювання використання CPU. Метрики можуть включати загальне використання, завантаженість окремих ядер та інші параметри. Встановлення порогів для використання CPU має виконуватися з огляду на стандарт «заліза».
  Наприклад, алерт при використанні CPU, що перевищує 90% протягом певного часу (30-40 хв) для хмарних хостів, 85% до 60 хв для віртуальних хостів та 80% до 30 хв для дискретного сервера.
- **RAM.** Моніторте використання оперативної пам'яті для уникнення переповнення. Зокрема, стежте за вільною та використаною RAM. Встановлення порогів для використання оперативної пам'яті.
  Наприклад, алерт при використанні RAM більше 80%.
- **Диск.** Моніторте доступність простору на диску, швидкість читання/запису та ступінь використання файлових систем. Встановлення порогів для вільного простору на диску та швидкості читання/запису. Наприклад, алерт при вільному просторі менше 10% логічного диску.
- **Мережа.** Вимірюйте пропускну здатність та використання мережі. Встановлення порогів для використання пропускної здатності. Наприклад, алерт при використанні мережі більше 90%.

- **Визначення порогів та алертів**
  - **Визначення порогів** повинно відбуватися відповідно до конкретних потреб інфраструктури та додатків. Різні сценарії можуть вимагати різних значень порогів.
  - **Тестування рішення.** Перевірка обраного рішення для визначення, які пороги є оптимальними для системи. Тестування дозволяє виявити можливі проблеми та уточнити налаштування, яке має зменшити хибні спрацювання алертів, але зберегти високу функціональність моніторингу.
- **Алерти**
  - **Конфігурація системи алертів.** Визначення того, як система повинна відправляти алерти - електронною поштою, через систему керування подіями, або іншими засобами. Як правило, конфігурація алертів прописується у стандартах підприємства. Є корпоративні канали спілкування, які і обираються для відправки таких повідомлень.
  - **Задання пріоритетів.** Розробка системи пріоритетів для алертів. Наприклад, алерти про високе використання CPU можуть мати вищий пріоритет, ніж алерти про вільне місце на диску.
  - **Визначення дій**. Визначення конкретних дій, які слід приймати при отриманні алерту. Це може включати автоматичні заходи, збір інформації для аналізу або відправку повідомлення адміністратору.

Налаштування моніторингу ресурсів серверів включає в себе не лише визначення того, що слід моніторити, але і як ефективно використовувати алерти для вчасного виявлення проблем і забезпечення оптимального функціонування інфраструктури.

- **Моніторинг додатків.** Розглянемо процес запуску та конфігурації моніторингу для додатків, а також важливість відслідковування продуктивності та доступності.
  - **Запуск та конфігурація моніторингу додатків**
    - **Вибір моніторингових інструментів**, а саме визначення, які інструменти моніторингу будуть використовуватися. Деякі додатки можуть мати вбудовані засоби моніторингу, або ви можете використовувати універсальні рішення, такі як Prometheus, Nagios, Zabbix чи інші.
    - **Конфігурація моніторингових агентів.** Якщо додаток вимагає агент для моніторингу, налаштуйте агенти на серверах, де розгорнуто додаток.
    - **Визначення параметрів моніторингу**. Оберіть, які параметри додатку слід моніторити. Це може включати використання ресурсів (CPU, пам'ять, диск), стан служб, пропускну здатність мережі та інші ключові метрики.
    - **Налаштування порогів та алертів.** Встановлення порогів для параметрів моніторингу та алертів. Наприклад, алерт при перевищенні певного рівня використання ресурсів або при відключенні служби.
  - **Відслідковування продуктивності та доступності**
    - **Використання ресурсів.** Спостерігайте за використанням CPU, оперативної пам'яті, диска та інших ресурсів, щоб вчасно виявляти проблеми та забезпечити оптимальну продуктивність.
    - **Логи та події.** Аналізуйте логи додатку для виявлення помилок, попереджень та інших важливих подій, які можуть вказувати на проблеми у роботі додатку.
    - **Моніторинг відгуків**. Вимірюйте час відгуку додатку для оцінки його продуктивності та відчуття кінцевого користувача та спостерігайте доступність додатку за допомогою пінгів або HTTP-запитів.
    - **Алерти та автоматизація відновлення.** Встановіть алерти для виявлення проблем доступності та автоматизуйте відновлення сервісів або додатку в разі виявлення неполадок.
    - **Тестування відмовостійкості.** Проводьте тестування відмовостійкості для переконання в тому, що додаток може витримати можливі неполадки та продовжувати працювати.

Ефективний моніторинг додатків дозволяє оперативно реагувати на проблеми, підтримувати високу доступність та оптимізувати продуктивність.

<u>Налаштування сповіщень та автоматизація управління системним моніторингом</u>

- **Встановлення правил сповіщень.** Розглянемо, як правильно вибирати сповіщення для різних сценаріїв та налаштовувати ескалації.
  - **Вибір правильних сповіщень для різних сценаріїв**
    - **Критичні алерти.** Використовуйте SMS або push-сповіщення для найважливіших алертів. Електронні листи можуть бути ефективними лише для певних категорій критичних подій.
    - **Загроза безпеки або інциденти.** Використовуйте телефонні дзвінки або системи голосового сповіщення для оперативного реагування на загрози безпеки. SMS або електронні листи можуть додатково використовуватися для швидкого розповсюдження інформації.
    - **Інформаційні повідомлення.** Електронні листи чи сповіщення через платформи спільної роботи підходять для інформаційних повідомлень, які не потребують негайного реагування.
    - **Врахування ролей та відповідальностей.** Налаштуйте сповіщення відповідно до ролей користувачів і їх відповідальностей. Наприклад, інженерам можуть надсилатися технічні деталі алертів, тоді як менеджерам - загальна інформація.
    - **Групові сповіщення.** Використовуйте сповіщення для груп користувачів, коли проблема впливає на кілька відділень або команд.
  - **Конфігурація ескалацій**

- **Визначення тайм-фреймів.** Налаштуйте часові інтервали для ескалацій, щоб забезпечити, що проблеми будуть вирішені у визначені строки.
- **Автоматизовані ескалації.** Використовуйте автоматизовані системи для ескалацій, щоб уникнути затримок у випадках, коли персонал не реагує.
- **Особистий зв'язок.** Встановлюйте правила для особистого зв'язку, таких як телефонні дзвінки, в разі важливих або неурядових подій.
- **Включення вищого керівництва.** Ескалюйте алерти до вищого керівництва у випадках, коли проблема вимагає внутрішньоорганізаційних рішень чи рішень на вищому рівні.
- **Журналювання подій.** Ведення журналу подій та аналіз ескалацій для подальшого вдосконалення процесу моніторингу.
- **Оцінка ефективності.** Регулярно оцінюйте ефективність ескалацій та внесіть зміни в процеси, якщо це необхідно.

Встановлення правил сповіщень та ескалацій важливо для забезпечення того, щоб відповідальні особи отримували необхідну інформацію у вчасний спосіб та вчасно реагували на події, що виникають в інфраструктурі.

- **Автоматизація реакції на події.** Розглянемо використання скриптів та автоматизаційних засобів, а також інтеграцію з іншими системами управління.
  - **Використання скриптів та автоматизаційних засобів**
    - **Розробка автоматизованих сценаріїв.** Використовуйте мови програмування або спеціальні скриптові мови для розробки сценаріїв автоматизації.
    - **Взаємодія з API**. Використовуйте API для взаємодії з іншими системами та автоматизації завдань.
    - **Використання Ansible, Puppet, Chef тощо.** Інтегруйте засоби автоматизації конфігурації та деплойменту для автоматизації відновлення стану системи.
    - **Системи оркестрації, такі як Kubernetes.** Використовуйте для автоматизації управління контейнерами та додатками.
  - **Інтеграція з іншими системами управління**
    - **API і інтеграційні рішення.** Використовуйте API для взаємодії між різними інструментами моніторингу та управління.
    - **Інтеграція з системами керування подіями (SIEM).** Передавайте дані про події до систем SIEM для докладного аналізу та виявлення загроз безпеки.
    - **Автоматизовані заходи відновлення.** Налаштуйте автоматизовані заходи для відновлення стану системи без втручання адміністратора.
    - **Інтеграція з іншими системами управління (ITSM.)** Виводьте автоматично створені та закриті тікети у систему управління обслуговуванням IT для документування та аналізу подій.
    - **Моніторинг результатів автоматизації.** Визначайте показники ефективності та моніторте результати автоматизованих дій.
    - **Вдосконалення сценаріїв.** Постійно вдосконалюйте та адаптуйте автоматизовані сценарії на основі вивчених уроків.

Використовуючи скрипти, автоматизаційні засоби та інтеграцію з іншими системами, можна створити гнучкі та ефективні процеси автоматичної реакції на різноманітні сценарії.

Висновки

Ми одним оком подивилися на кілька інструментів системного моніторингу з дуже широкого існуючого спектру, який вимагає уважного вибору залежно від конкретних завдань та бюджетних обмежень.
Nagios: Підходить для ефективного моніторингу серверів та мережі, але може вимагати додаткових ресурсів для розширення функціональності.
Zabbix: Відмінно підходить для моніторингу ресурсів та додатків, з високою гнучкістю та можливістю інтеграції.
Prometheus: Ефективний у контейнеризованих середовищах, забезпечує моніторинг та аналіз метрик.
MS SCOM: Оптимальний для Microsoft-орієнтованих інфраструктур, з інтеграцією з продуктами Microsoft.
Особливу роль при виборі інструментів системного моніторингу відіграють бюджетні обмеження. Іншими словами: вибір між безкоштовними та комерційними інструментами залежить від фінансових можливостей та потреб користувача.
Навіть дуже поширених універсальних інструментів системного моніторигу існує кілька десятків. Nagios, Zabbix, Prometheus та MS SCOM - лише вершина айсберга.
Перед налаштуванням системи моніторингу першорядним завданням залишається глибоке дослідження ринку інструментів, з урахуванням конкретних вимог проекту та фінансових реалій

.