План лекції . Тема 2. Архітектура системного моніторингу.

- Компоненти системи моніторингу: агенти, сервери, бази даних.
- Розгорнуті та розподілені архітектури моніторингу.
- Протоколи зв'язку між компонентами системи моніторингу.

Компоненти системи моніторингу: агенти, сервери, бази даних.

Архітектура системного моніторингу - це структурована організація компонентів, процесів та взаємодії між ними в системі моніторингу для забезпечення ефективного збору, аналізу та відображення інформації про стан і функціонування інформаційно-технічної системи.

Для розуміння архітектури моніторингу розглянемо спрощений узагальнений алгоритм моніторингу. Відповідно до цієї спрощеної моделі, система моніторингу складається з чотирьох основних компонентів: спостереження, аналізу, дії та зберігання.

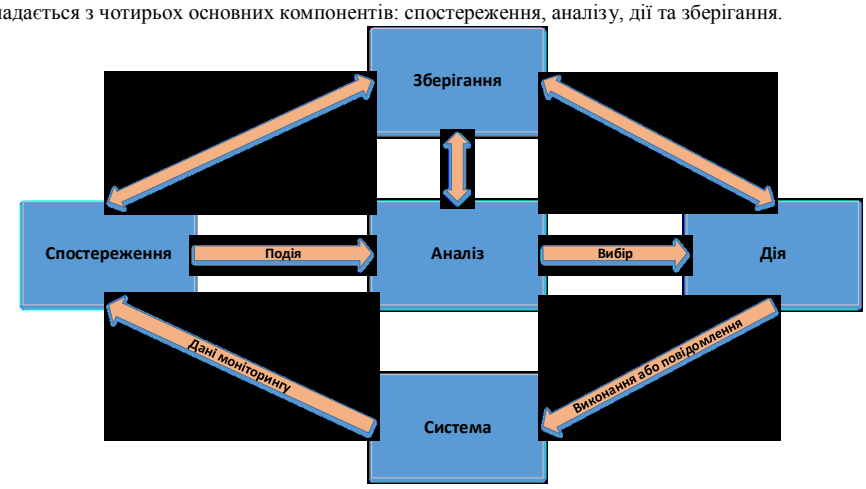


Рис. 2.1. Узагальнений алгоритм моніторингу

І відповідно до цієї моделі, алгоритм моніторингу виконує чотири завдання, які також характеризують основні компоненти алгоритму, як показано на рисунку 2.1. Компонент спостереження монітора отримує дані з точки спостереження за системою (або підсистемою) та генерує події моніторингу для подальшого аналізу.
Компонент зберігання даних двобічно взаємодіє з усіма іншими компонентами алгоритму.
Компонент аналізу реалізує політику моніторингу, тобто правила та умови, які необхідно перевірити. Якщо виявлено порушення, моніторинг запускає дії, які можуть варіюватися від простого звітування до примусового виконання певної поведінки в цільовій системі. Прикладами примусового виконання є просте припинення процесу, плавне вимкнення, заходи відновлення або коригуюче втручання.
Як вже згадувалося, всі інші компоненти алгоритму мають доступ до компонента зберігання для виконання різноманітних завдань, наприклад, для резервного копіювання необроблених даних або контрольних журналів, як бази знань для аналізу або для дій з реєстрації.

Ми розглянули алгоритм, а тепер, архітектура системного моніторингу, яка зазвичай включає в себе наступні ключові компоненти та взаємозв'язки:

- **Джерела даних.** Це компоненти, які постачають інформацію для моніторингу. Це може включати мережеве обладнання, сервери, програмне забезпечення, бази даних тощо. Джерела даних можуть використовувати різні протоколи та інтерфейси для передачі інформації.

- **Збір даних.** Цей компонент відповідає за збір даних від джерел та їхнє передавання до системи моніторингу. Включає в себе агенти, сенсори, протоколи (наприклад, SNMP, NetFlow) та інші механізми для отримання інформації.

- **Основна система моніторингу.** Це ядро системи, де збираються, аналізуються та зберігаються дані. Включає в себе базу даних для зберігання історії подій, движок правил для виявлення аномалій, систему управління для організації процесів моніторингу.

- **Візуалізація та звітність.** Компонент, який відповідає за представлення інформації користувачеві. Включає в себе інтерфейс користувача, графіки, діаграми, засоби відображення статусів та інші елементи для зручного сприйняття даних.

- **Система сповіщень та реагування.** Цей компонент відповідає за генерацію сповіщень при виявленні проблем або аномалій. Також може включати механізми автоматичного реагування або взаємодії з іншими системами для вирішення проблем.

- **Безпека.** Компонент, що забезпечує захист системи моніторингу від несанкціонованого доступу, шифрування даних та інші заходи для збереження конфіденційності та цілісності інформації.

- **Адміністративний інтерфейс.** Інтерфейс, який дозволяє адміністраторам налаштовувати та керувати системою моніторингу, встановлювати правила, конфігурувати засоби візуалізації та інші параметри.

Ці компоненти взаємодіють між собою для створення комплексної системи моніторингу, яка забезпечує адміністраторам інформацію про стан та продуктивність інформаційно-технічної інфраструктури та допомагає вчасно реагувати на можливі проблеми.

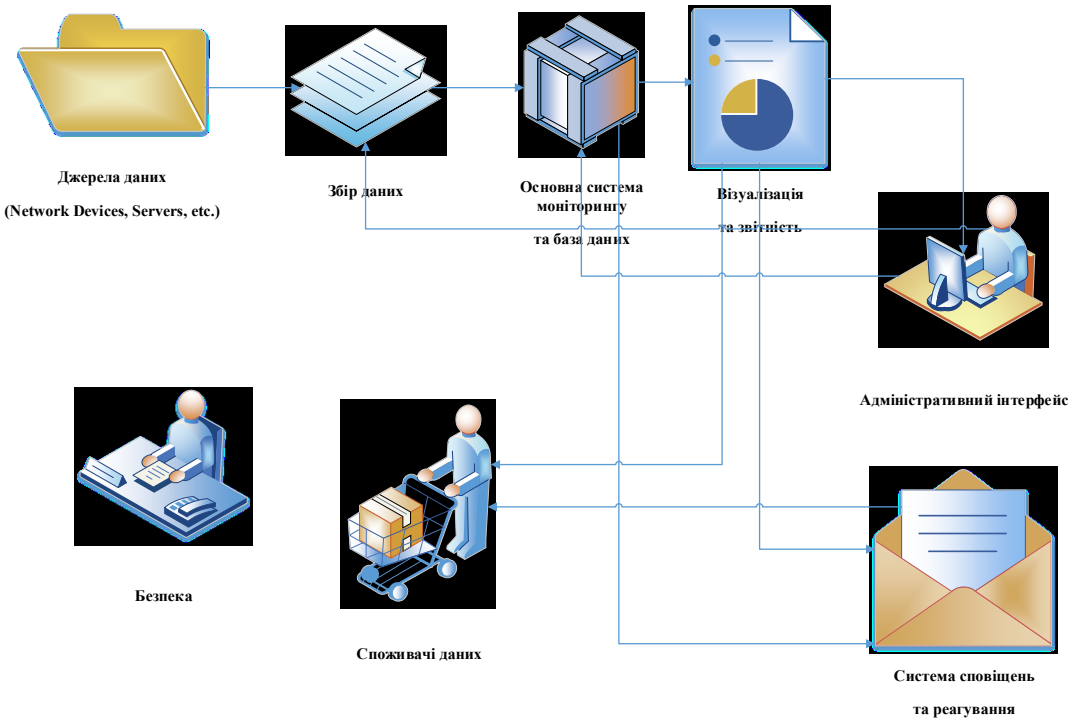


Рис. 2.2. Розширена діаграма, яка відображає взаємозв'язки між компонентами архітектури системного моніторингу

На рис. 2.2 показана розширена діаграма взаємозв'язків компонентів системного моніторингу, що включає роль людини в цій системі. Трошки конкретики про наведені у діаграмі компоненти.

- **Джерела даних (Network Devices, Servers, etc.)**: надають інформацію для моніторингу і є ключовими компонентами для збору інформації з інформаційно-технічної інфраструктури. Дані від цих джерел допомагають адміністраторам моніторити та аналізувати стан систем, вчасно виявляти проблеми, планувати розширення та оптимізацію мережі.. В реальному середовищі системного моніторинга ці компоненти можуть виглядати наступним чином:.

  - ✓ **Network Devices (Мережеве обладнання)**. Може включати маршрутизатори, комутатори, файрволи, мережеві перетворювачі та інше. Для моніторингу такого обладнання можна використовувати протоколи, такі як SNMP (Simple Network Management Protocol), які надають інформацію про стан мережевого обладнання, трафік, завантаження та інші метрики. Мережевому моніторингу буде присвячена одна з наступних лекцій даного курсу.

  - ✓ **Servers (Сервери)**. Фізичні або віртуальні сервери, які надають різні сервіси та додатки. Для моніторингу серверів використовують агенти (програми, що встановлюються на сервері) або взаємодіють з різними протоколами, такими як SNMP або WMI (Windows Management Instrumentation).

  - ✓ **Other Devices (Інші пристрої)**. Включає різноманітне обладнання, таке як датчики температури, вологості, UPS (забезпечення електроживлення обладнання у випадку відсутності чи аномалій в основному живленні), принтери та інше. Для моніторингу цих пристроїв можуть використовуватися різні протоколи або спеціалізовані програмні рішення.

  - ✓ **Applications (Додатки)**. Це програмне забезпечення, яке надає певні сервіси або функціональність. Для моніторингу додатків можуть використовуватися різні підходи, такі як взаємодія з API, використання агентів або моніторинг системних ресурсів, пов'язаних з додатком. Як приклад перевірка бази даних додатка або перевірка логів. У самому простому випадку – перевірка стану процесу, що ми навчимося гарно робити під час виконання лабораторних робіт.

  - ✓ **Security Devices (Засоби безпеки)**. Включають файрволи, системи виявлення та запобігання вторгненням (IDS/IPS), антивіруси та інші засоби безпеки. Для цих пристроїв використовуються протоколи та інтерфейси моніторингу безпеки. Тут треба згадати такий клас систем, споріднених до систем системного моніторингу (вибачте за тавтологію) як системи виявлення атак (вторгнень) (англ. Intrusion Detection System, IDS) Одразу приходять на думку такі гарні абревіатури як Suricata, Splunk, ELK Stack (Elasticsearch, Logstash, Kibana), Snort. Але зараз ми не будемо зупинятися на цій цікавій та перспективній тематиці, бо вона лежить трошки не у темі лекції.

- **Збір даних (Data Collection)**: збирає дані (хоча це і тавтологія, пробачте) від джерел та передає їх до основної системи моніторингу. У структурі системного моніторингу - це процес отримання інформації про стан і функціонування різних компонентів системи. Цей процес має на меті збирати дані з джерел, таких як мережеве обладнання, сервери, додатки, або інші пристрої, які підлягають моніторингу, і передавати ці дані для подальшого аналізу та візуалізації в системі моніторингу. Збір даних є важливим етапом в системі моніторингу, оскільки від якості та точності цього процесу залежить ефективність моніторингу та здатність системи реагувати на зміни стану. Після збору даних вони можуть бути використані для визначення трендів, виявлення аномалій, та взагалі оцінки продуктивності та доступності системи. Основні етапи та компоненти процесу збору даних:

- ✓ **Визначення джерел даних**. Визначення того, з яких пристроїв чи джерел системи буде здійснюватися збір інформації. Це може включати мережеве обладнання (роутери, комутатори), сервери, додатки, датчики, принтери, WEB-сайти та інше.

- ✓ **Вибір методу збору**. Вибір підходів та методів для отримання інформації з обраного джерела. Наприклад, для мережевого обладнання часто використовують протоколи, такі як SNMP (Simple Network Management Protocol), а для серверів - агенти моніторингу чи інші протоколи (наприклад, WMI для Windows).

- ✓ **Конфігурація агентів чи засобів збору**. Якщо використовуються агенти, їх потрібно налаштувати для збору конкретних метрик або даних. Налаштування включає в себе вибір параметрів моніторингу, інтервалів збору, та інші налаштування залежно від потреб системи.

- ✓ **Передача даних**. Передача зібраних даних до центральної системи моніторингу. Це може включати передачу за допомогою різних протоколів, API-запитів, або інших механізмів взаємодії.

- ✓ **Агрегація та зберігання**. Агрегація та зберігання зібраних даних в базі даних для подальшого аналізу, архівування та створення звітів.

- **Основна система моніторингу (Monitoring Core System) та база даних (DataBase)**: компонент відповідає за приймання, обробку та аналіз зібраних даних в реальному чи поблизькому до реального часу. Основні функції цього компоненту включають:

  - ✓ **Прийом та обробка даних**. Отримання інформації від джерел даних (збірники, агенти, інші джерела) та її подальша обробка. Це включає валідацію даних, конвертацію в однорідний формат, та виявлення можливих аномалій чи проблем.

  - ✓ **Зберігання даних**. Дані, отримані від різних джерел, зберігаються в базі даних для подальшого використання. Звичайно, зберігання включає архівацію історичних даних, що дозволяє вести аналіз та виявляти тренди протягом тривалого періоду.

  - ✓ **Аналіз та моніторинг**. Виконання аналізу даних для виявлення аномалій, визначення трендів та подальший моніторинг. Аналітичні засоби можуть включати в себе різні алгоритми, правила, та методи для ідентифікації проблем чи аномалій у системі.

  - ✓ **Генерація звітів та візуалізація**. Виведення результатів аналізу у вигляді графіків, діаграм, звітів та інших візуальних засобів для зручного сприйняття та розуміння стану системи.

  - ✓ **База даних** в контексті системного моніторингу є сховищем для зберігання різноманітної інформації, отриманої від джерел даних. Основні аспекти її функціонування включають:
    - ❖ **Зберігання інформації**. База даних зберігає отримані дані про стан системи, її компоненти, метрики та інші параметри. Це дозволяє ведення історії подій та зберігання інформації для подальшого аналізу.
    - ❖ **Архівація та запити**. Забезпечення можливості архівації історичних даних та виконання запитів для вибірки необхідної інформації. Це дозволяє адміністраторам та системам аналізувати та використовувати дані для різноманітних потреб.
    - ❖ **Оптимізація доступу**. Оптимізація швидкості доступу до даних для ефективного використання системи моніторингу, особливо коли обсяг даних є великим.
    - ❖ **Безпека даних**. Забезпечення безпеки зберігання та доступу до даних. Це включає в себе встановлення прав доступу, шифрування та інші заходи для забезпечення конфіденційності та цілісності даних.

  Основна система моніторингу та база даних тісно співпрацюють, забезпечуючи збір, аналіз, та зберігання інформації для ефективного моніторингу та управління інфраструктурою. Звичайно, що бази даних це велика окрема тема. Відмічу лише, що більшість систем моніторингу працюють з «легкими», безкоштовними базами типу MySQL або чогось свого, пропрієтарного.

- **Візуалізація та звітність**: є важливим елементом архітектури системного моніторингу, оскільки надає адміністраторам та користувачам зручний та зрозумілий інтерфейс для сприйняття та аналізу зібраних даних. Основні функції цього компонента включають:
  - ✓ **Графіки та діаграми**. Надають можливість побудови графіків та діаграм для візуалізації трендів, змін та стану різних параметрів системи. Це допомагає адміністраторам швидко розпізнати зміни та проблеми.

  - ✓ **Дашборди**. Система дозволяє створювати та налаштовувати дашборди, де можна об'єднувати різні елементи візуалізації для зручного спостереження за ключовими метриками та подіями.

  - ✓ **Звіти та аналітика**. Надають можливість створення звітів та аналітичних звітів на основі зібраних даних. Адміністратори можуть генерувати різноманітні звіти для аналізу ефективності системи та її компонентів.

  - ✓ **Реальний час**. Забезпечує можливість відстежування даних в реальному часі. Це дозволяє оперативно реагувати на події та аномалії, що відбуваються в системі.

  - ✓ **Фільтрація та групування**. Забезпечують можливість фільтрації та групування даних для зручного відображення конкретної інформації. Це особливо корисно в системах з великою кількістю компонентів та метрик.

  - ✓ **Повідомлення та сповіщення**. Інтеграція системи візуалізації з системою сповіщень для оперативного інформування адміністраторів про важливі події чи аномалії.

  - ✓ **Користувацькі налаштування**. Надають можливість адміністраторам налаштовувати інтерфейс з урахуванням їхніх власних потреб та візуальних уподобань.

Компонент візуалізації та звітності важливий для забезпечення адміністраторів інструментами для ефективного моніторингу та управління системою. Він допомагає перетворити складні дані в зрозумілу форму, полегшуючи процес прийняття рішень та реагування на події в реальному часі.

- **Система сповіщень та реагування**: виявляє аномалії та сповіщає адміністраторів та є критично важливою для забезпечення надійності та продуктивності системи. Цей компонент дозволяє адміністраторам оперативно реагувати на події та забезпечує швидке виправлення проблем для забезпечення неперервності роботи інфраструктури. Основні аспекти цього компонента включають:

  - ✓ **Виявлення аномалій**. Система використовує аналітичні алгоритми та правила для виявлення аномальних станів або несподіваних змін в параметрах системи. Це може включати в себе порогові значення, шаблони поведінки та інші методи виявлення незвичайних ситуацій.

  - ✓ **Сповіщення**. При виявленні аномалії система надсилає сповіщення адміністраторам або відповідальним особам. Сповіщення може бути реалізоване через різні канали, такі як електронна пошта, SMS, месенджери чи інші засоби зв'язку.

  - ✓ **Конфігурування правил**. Адміністратори можуть налаштовувати правила сповіщень відповідно до потреб системи та їхніх власних пріоритетів. Це включає визначення умов для сповіщень, методів доставки, та інших параметрів.

  - ✓ **Система пріоритетів**. Врахування різних рівнів важливості для різних типів сповіщень. Деякі події можуть вимагати негайної реакції, тоді як інші можуть бути позначені як менш важливі.

  - ✓ **Інтеграція з іншими системами**. Система Сповіщень може інтегруватися з іншими системами, такими як системи керування інцидентами чи автоматизовані засоби реагування на проблеми.

  - ✓ **Автоматичне реагування**. Можливість налаштовувати автоматичне реагування на деякі типи подій чи аномалій без прямого втручання адміністраторів. Це може включати автоматичне відновлення послуг, виключення проблемних компонентів, то що.

  - ✓ **Журналізація та відстеження**. Запис подій та реакцій в журналах для подальшого аналізу та аудиту. Це дозволяє адміністраторам вивчати історію подій та визначати ефективність заходів реагування.

- **Споживачі даних** - це сутності або системи, які використовують зібрані дані для виконання різноманітних завдань. У якості споживачів даних у системах моніторингу можуть виступати автоматизовані системи, програми, скрипти, а також самі адміністратори, чи навіть звичайні користувачі системи моніторингу (звичайно, якщо Ви як системний адміністратор налаштуєте їм певний доступ) які взаємодіють з системою через адміністративний інтерфейс. Основні ролі споживачів даних включають:

  - ✓ **Автоматизовані системи та інтеграція**. Системи моніторингу часто інтегруються з іншими автоматизованими інфраструктурними системами. Наприклад, інтеграція з системами управління інцидентами або іншими інструментами для автоматизованого реагування на проблеми.

  - ✓ **Споживачі API**. *Абревіатура "API" в даному випадку означає "Інтерфейс програмування застосунків" (англ. Application Programming Interface).* Споживачі можуть використовувати API для отримання даних з системи моніторингу. Це може бути корисно для створення власних звітів, інтеграції з іншими інструментами або створення власних автоматизованих сценаріїв.

  - ✓ **Системи моніторингу продуктивності**. Споживачі даних можуть бути спрямовані на оцінку продуктивності системи моніторингу. Це може включати в себе аналіз завантаження, швидкості відгуку та інших параметрів для забезпечення ефективності системи.

  - ✓ **Адміністраторські застосунки та інтерфейс**. Самі адміністратори, використовуючи адміністративний інтерфейс, є споживачами даних. Вони взаємодіють з системою для налаштування параметрів, виявлення аномалій, аналізу даних та прийняття рішень.

  - ✓ **Генерування звітів та аналітика**. Споживачі можуть використовувати дані для створення різноманітних звітів, аналітичних висновків та статистичних даних для внутрішніх чи зовнішніх потреб.

  - ✓ **Аналіз трендів та прогнозування**. Системи аналізу даних можуть використовувати зібрані дані для виявлення трендів, прогнозування можливих проблем, та прийняття стратегічних рішень.

  - ✓ **Інтерфейс кінцевого користувача**. В деяких випадках, кінцеві користувачі або клієнти можуть бути споживачами даних, якщо система моніторингу надає можливість перегляду певних метрик або стану для зовнішніх зацікавлених сторін.

- **Адміністративний інтерфейс**: дозволяє адміністраторам налаштовувати систему та приймати рішення на основі отриманих даних і є ключовим для забезпечення адміністраторам зручного та ефективного інструмента для конфігурації системи, прийняття рішень та взаємодії з іншими компонентами, налаштовувати параметри системи моніторингу відповідно до потреб та вести контроль над всіма аспектами моніторингу та управління інфраструктурою. Основні характеристики цього компонента включають:

  - ✓ **Керування конфігурацією**. Надає можливість налаштовувати параметри системи моніторингу, включаючи джерела даних, правила моніторингу, інтервали збору, та інші параметри. Це може включати в себе інтерфейс для додавання, вилучення чи зміни джерел даних.

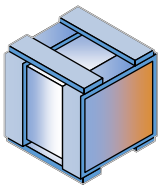
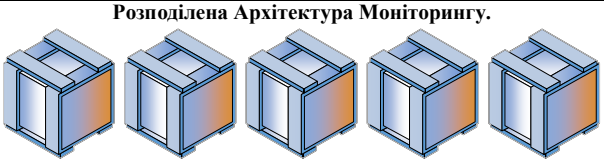
  - ✓ **Налаштування правил сповіщень**. Дозволяє адміністраторам налаштовувати правила та умови, за якими генеруються сповіщення. Це включає в себе визначення порогових значень, типів повідомлень та адреси для сповіщень.

- ✓ **Управління користувачами та доступом**. Забезпечує інтерфейс для керування обліковими записами користувачів, а також налаштування рівнів доступу та привілеїв для різних користувачів системи моніторингу.
- ✓ **Дашборди та візуалізація**. Дозволяє адміністраторам створювати та налаштовувати дашборди для зручного відображення ключових метрик та інформації. Це дозволяє швидко отримувати огляд стану системи.
- ✓ **Керування звітами та аналітикою**. Надає інтерфейс для створення та конфігурації звітів, аналітичних висновків та інших інструментів для детального аналізу даних.
- ✓ **Моніторинг продуктивності системи**. Надає засоби для відстеження та оцінки продуктивності самої системи моніторингу. Це може включати в себе метрики щодо завантаження системи, швидкості відгуку та інші параметри.
- ✓ **Інтеграція з іншими системами**. Забезпечує можливість інтеграції з іншими інфраструктурними системами та інструментами для взаємодії та обміну даними.
- ✓ **Журналізація та аудит**. Запис подій та дій адміністраторів для подальшого аудиту та вивчення історії змін.

- **Безпека**: Забезпечує захист системи від несанкціонованого доступу та збереження конфіденційності даних та відіграє ключову роль у захисті важливих інформаційних ресурсів та забезпеченні конфіденційності, цілісності та доступності даних в системі моніторингу. Основні аспекти цього компонента включають:
  - ✓ **Аутентифікація та авторизація.**
    Аутентифікація: Визначення і перевірка ідентичності користувачів або систем, що намагаються отримати доступ до системи моніторингу. Це може включати в себе використання паролів, біометричних даних чи інших методів.
    Авторизація: Визначення прав доступу та привілеїв для користувачів або систем, які вже автентифіковані. Забезпечення того, що користувачі мають лише той рівень доступу, який їм необхідний.
  - ✓ **Шифрування даних**. Застосування шифрування для захисту даних, які передаються між компонентами системи моніторингу. Це може включати шифрування даних в покоїному стані, а також шифрування під час передачі.
  - ✓ **Моніторинг та аудит безпеки**. Забезпечення системи функціоналом моніторингу та аудиту подій для виявлення можливих загроз, несанкціонованих спроб доступу та аномалій в системі.
  - ✓ **Захист від вторгнень**. Впровадження заходів для запобігання вторгненням в систему моніторингу. Це може включати в себе виявлення та блокування невірних спроб доступу, антивірусний контроль та інші заходи.
  - ✓ **Фізична безпека**. Захист фізичних ресурсів, на яких базується система моніторингу, таких як серверні приміщення, центральні сервери та комунікаційні канали.
  - ✓ **Методи аутентифікації та захисту API**. Забезпечення безпеки взаємодії з іншими системами через API, включаючи використання токенів, ключів доступу та інших методів аутентифікації та захисту.
  - ✓ **Захист інфраструктури**. Забезпечення безпеки компонентів інфраструктури, таких як бази даних, сервери та мережеві пристрої, для запобігання несанкціонованого доступу та атак.
  - ✓ **Резервне копіювання та відновлення**. Реалізація стратегій резервного копіювання та відновлення даних для забезпечення можливості відновлення системи після інцидентів або атак.

Розгорнуті та розподілені архітектури моніторингу.

Таблиця 02.01

Розгорнута архітектура моніторингу.	**Розподілена Архітектура Моніторингу.**
	
Характеристики: • **Централізована**. Всі компоненти, включаючи агенти та сервери моніторингу, розташовані в одному центральному місці. • **Одна центральна база даних**. Використання однієї центральної бази даних для збереження всіх зібраних даних. • **Масштабування за допомогою збільшення ресурсів**. Масштабування системи досягається за рахунок збільшення ресурсів на центральному сервері.	**Характеристики:** • **Розподілена інфраструктура**. Компоненти системи розташовані на різних фізичних чи віртуальних серверах, можливо, розташованих географічно чи топографічно в різних місцях. • **Багатоцентрові бази даних**. Використання багатоцентрових баз даних, можливо, розташованих в різних регіонах або областях.
Ключові Компоненти:	**Ключові Компоненти:**

• **Агенти**. Інстальовані на кожній системі, яку потрібно моніторити. • **Центральний Сервер**. Обробляє та зберігає дані, відправлені агентами. • **Центральна база даних**. Зберігає всі дані з моніторингу.	• **Агенти**. Розподілені на системах у різних місцях. • **Локальні сервери**. Сервери моніторингу, які розташовані ближче до джерел даних. • **Глобальні сервери**. Центральні сервери, які об'єднують та агрегують дані з різних локальних серверів. • **Багатоцентрові бази даних**. Централізовані та локальні бази даних для ефективного зберігання та обробки інформації.
Застосування: Для невеликих або середніх мереж, де централізованого підходу вистачає для обробки обсягу даних.	**Застосування:** • **Глобальні організації**. Для компаній або організацій з розподіленим географічним розташуванням джерел даних. • **Великі IT-інфраструктури**. Для великих комплексних систем, де обсяг даних великий і розподілений.

Різниця між розгорнутою та розподіленою архітектурою:

- **Централізований контроль vs. децентралізований контроль**. Розгорнута архітектура забезпечує централізований контроль, тоді як розподілена архітектура реалізує децентралізований підхід з розподіленими серверами та базами даних.

- **Масштабування**. Розгорнута архітектура масштабується головним чином за рахунок збільшення ресурсів на центральному сервері. Розподілена архітектура може масштабуватися горизонтально за рахунок додавання нових серверів у різних локаціях.

- **Вартість трафіку**. Розгорнута архітектура може мати більшу кількість трафіку між агентами та центральним сервером. У розподіленій архітектурі частина трафіку може бути оброблена локально перед відсиланням на центральний рівень.

Вибір між розгорнутою та розподіленою архітектурою залежить від:

- **Масштабу проекту**. Для невеликих та середніх проектів розгорнута архітектура може бути більш простою. Розподілена архітектура підходить для великих та розгалужених інфраструктур.

- **Географічного розташування**. Якщо джерела даних розташовані географічно розподілено, розподілена архітектура може бути ефективнішою.

- **Вимог безпеки та збереження доступності**. Якщо важливість безпеки та доступності велика, розподілена архітектура може забезпечити більший рівень стійкості до відмов та атак.

Також необхідно згадати, що вибір між розгорнутою та розподіленою архітектурою системи моніторингу не є «незворотнім». В залежності від перелічених вище залежностей та архітектури такої важливої складової системи моніторингу як Monitoring Core System та база даних (Database) архітектура може бути змінена.
Monitoring Core System та Database, як ми вже згадували, є ключовими компонентами сучасних систем моніторингу і саме у їх виборі криється можливість масштабування чи переходу до розподіленої архітектури.
Ось декілька прикладів таких «ядер систем» та стандартних (з коробки) баз даних:

Таблиця 02.02

Core System	База даних	Короткий опис	Взаємодія з БД
Prometheus	**Prometheus Database**	Відкрита система моніторингу та оповіщення, яка призначена для збору та відображення метрик з систем та додатків	Використовує власну базу даних для зберігання зібраних метрик. База даних Prometheus оптимізована для швидкого читання та запитів на дані метрик.
Grafana	**InfluxDB**	Платформа візуалізації та аналізу даних, яка працює поверх інших систем моніторингу, таких як InfluxDB	InfluxDB представляє собою спеціалізовану систему для зберігання та обробки часових рядів даних. Ця база даних часових рядів є ідеальним вибором для зберігання та отримання метричних даних, які використовуються Grafana для подальшої візуалізації та аналізу.
Nagios	**MySQL**	Система моніторингу, яка використовується для виявлення та реагування на проблеми в інфраструктурі.	У деяких випадках Nagios може використовувати MySQL або інші бази даних для зберігання конфігураційних даних та даних стану системи.
Zabbix	**PostgreSQL**	Комплексна система моніторингу, яка надає різні функції, включаючи збір, аналіз та візуалізацію даних.	Зазвичай Zabbix використовує PostgreSQL для зберігання конфігураційних даних, статистики та журналів.
Dynatrace	**Cassandra**	Рішення для моніторингу та управління додатками, яке використовує штучний інтелект для аналізу продуктивності.	Cassandra може використовуватися для зберігання метричних даних, які вивчає Dynatrace для аналізу
MS System Center Operations Manager (SCOM)	**Microsoft SQL Server**	Система моніторингу та управління для операційних систем, інфраструктури та додатків у середовищах Windows. SCOM дозволяє виявляти та реагувати на проблеми у реальному часі, а також здійснювати аналіз продуктивності.	SCOM використовує базу даних для зберігання конфігураційних даних, статусів моніторингу та інших важливих інформаційних показників. Зазвичай, для цього використовується Microsoft SQL Server.

.
Цих кілька прикладів показують різноманітність систем моніторингу та їхніх баз даних, що може включати в себе різні комбінації для різних потреб та вимог.

Таким чином, обираючи між цими архітектурними підходами та типами компонентів системи моніторингу, слід враховувати конкретні вимоги проекту, його розмір та географічні особливості.

Протоколи зв'язку між компонентами системи моніторингу.

Система моніторингу є ключовим елементом для забезпечення стабільності, ефективності та безпеки інформаційних технологій та бізнес-процесів і одразу зрозуміло, що ефективний зв'язок між компонентами цієї системи є дуже важливим, оскільки він впливає на наступні аспекти функціонування та результативності:

- **Збір та передача даних**. Ефективний зв'язок дозволяє надійно та швидко збирати дані з агентів та інших джерел моніторингу і відповідно - затримки або втрати при передачі можуть призвести до неповної чи неправильної інформації, ускладнюючи процес моніторингу та реагування на проблеми.

- **Швидкість реагування на аномалії**. Зв'язок визначає швидкість обміну інформацією про виникнення аномалій чи проблем в системі, а затримки у виявленні або сповіщенні про аномалії можуть погіршити реакцію на критичні ситуації та збільшити час простою системи.

- **Масштабування та ресурсоємність**. Забезпечення ефективного зв'язку необхідно для масштабування системи та розподілених ресурсів. Неправильно спроектований зв'язок може призвести до перевантаження мережі та серверів, обмежуючи можливість системи моніторингу працювати в розподіленому середовищі.

- **Безпека та конфіденційність**. Зв'язок повинен бути забезпечений для захисту передачі конфіденційної інформації та управлінських команд. Небезпечне, або незахищене з'єднання може стати джерелом загроз безпеці та джерелом витоку конфіденційної інформації.

- **Інтеграція та сумісність**. Інтеграція з іншими системами та сумісність з різними пристроями та платформами вимагають ефективного зв'язку. Несумісність може призвести до втрати функціональності та неефективного використання ресурсів.

- **Автоматизація та динамічна керованість**. Ефективний зв'язок дозволяє автоматизувати процеси моніторингу та приймати динамічні управлінські рішення. Поганий зв'язок може ускладнити впровадження автоматизації та обмежити можливості динамічного керування.

- **Легкість управління та налаштування**. Зв'язок повинен бути простим у налаштуванні та управлінні для забезпечення ефективного функціонування системи моніторингу. Складний зв'язок може збільшити час та ресурси, необхідні для управління та підтримки системи.

Ефективність зв’язку між компонентами будь якої системи забезпечується каналами зв’язку та ефективним вибором протоколу.

В системі моніторингу ефективність зв’язку забезпечує надійність, продуктивність та безпеку обміну даними між компонентами. Окрім вибору протоколу та каналів зв'язку, існує кілька інших факторів, які суттєво впливають на ефективність системного моніторингу:

Таблиця 02.03

Фактор зв’язку	Опис фактору	Вплив фактору
Пропускна здатність каналів зв'язку	Пропускна здатність визначає, скільки даних може бути передано через канал зв'язку за одиницю часу.	Велика пропускна здатність дозволяє передавати більше даних, що особливо важливо в сучасних великих мережах.
Затримка та латентність	Затримка показує час, необхідний для передачі сигналу від відправника до отримувача. Латентність визначає загальний час затримки для обробки та передачі даних.	Низька затримка та латентність дозволяють швидше реагувати на події та отримувати актуальну інформацію.
Протоколи та кодеки компресії	Використання ефективних протоколів та кодеків для стиснення даних може значно зменшити обсяг передаваних інформаційних пакетів.	Зменшення обсягу даних покращує швидкість передачі та економить мережевий трафік.
Безпека та шифрування	Використання безпечних протоколів та шифрування даних забезпечує конфіденційність та інтегрітет інформації.	Забезпечення безпеки мережі та даних є важливим для запобігання несанкціонованому доступу та збереження конфіденційності.
Резервування та відновлення	Використання механізмів резервування та відновлення дозволяє побудувати стійку до відмов мережеву інфраструктуру.	Забезпечення надійності та стійкості мережі шляхом автоматичного відновлення та переключення на резервні маршрути.
Балансування навантаження	Розподіл навантаження між різними каналами зв'язку або серверами для оптимізації використання ресурсів.	Балансування навантаження допомагає уникнути перевантаження та забезпечити рівномірне використання ресурсів.
Масштабованість	Можливість системи збільшити свою пропускну здатність та функціональність при збільшенні об'єму оброблюваних даних.	Масштабованість гарантує ефективність зв'язку при зростанні обсягу даних та обчислень.

Протоколи в системі моніторингу відіграють важливу роль у забезпеченні ефективного обміну даними та взаємодії між різними компонентами, такими як агенти, сервери та інші елементи інфраструктури. Вони створюють стандартизовані правила та процедури, які дозволяють різним частинам системи ефективно спілкуватися та виконувати свої функції. Розуміння та використання різних протоколів в системі моніторингу є важливим елементом для забезпечення ефективного та надійного обміну даними, що визначає успішність функціонування системи та якість зібраних інформаційних даних. Роль протоколів у забезпеченні обміну даними та взаємодії:

- **Стандартизація комунікації**. Протоколи визначають стандартні правила та формати для обміну даними між агентами та серверами, що забезпечує єдність у способі взаємодії, незалежно від типу пристрою чи платформи. Стандартизована комунікація спрощує інтеграцію різних компонентів та полегшує підтримку системи.

- **Забезпечення надійності та цілісності даних**. Протоколи визначають механізми для перевірки та підтвердження доставки даних, а також їх цілісності під час передачі. Це гарантує, що дані доставлені правильно та не пошкоджені, а забезпечення надійності даних дозволяє уникнути втрати інформації та запевняє точність результатів моніторингу.

- **Управління безпекою**. Протоколи включають механізми шифрування та аутентифікації, щоб забезпечити безпеку обміну даними між агентами та серверами. Управляння безпекою дозволяє захистити конфіденційні інформаційні ресурси. Захист від несанкціонованого доступу та збереження конфіденційності даних ускладнюється без ефективних протоколів безпеки.

- **Оптимізація трафіку та ресурсів**. Протоколи оптимізують передачу даних, управляючи розміром пакетів, використовуючи стиснення та мінімізуючи надлишкові запити. Ефективний обмін даними сприяє економії мережевого трафіку та ресурсів, забезпечуючи оптимальну продуктивність системи.

- **Підтримка розподілених систем**. Протоколи дозволяють взаємодіяти між розподіленими компонентами системи моніторингу через мережу, що сприяє розширенню системи на великі відстані, взаємодії з різними джерелами та використанню розподілених обчислювальних ресурсів.

- **Підтримка різних платформ та протоколів**. Протоколи можуть бути розроблені для роботи на різних платформах та підтримувати різні версії протоколів для сумісності з різними версіями програмного забезпечення, що забезпечує сумісність з різними системами та версіями, гарантує гнучкість та розширюваність системи моніторингу.

- **Спрощення управління та налаштування**. Протоколи, які мають чіткі та прості правила, спрощують управління та налаштуванням системи моніторингу, що зменшує час та ресурси, необхідні для внесення змін у конфігурацію та підтримки системи.

Таким чином, протоколи зв'язку грають ключову роль у взаємодії між різними компонентами системи моніторингу, забезпечуючи ефективний обмін даними та забезпечуючи спільну мову для взаємодії між агентами, серверами та іншими елементами інфраструктури. Ось кілька основних протоколів, які широко використовуються в системах моніторингу:

Таблиця 02.04

Протокол	Короткий опис	Найпоширеніші компоненти
SNMP (Simple Network Management Protocol)	SNMP є одним з найпоширеніших протоколів для моніторингу та управління мережевими пристроями. Використовується для збору та передачі інформації про стан пристроїв, таких як роутери, комутатори та сервери.	Агенти SNMP розташовані на кожному пристрої, який моніториться, та центральному сервері, який здійснює опитування.
	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). Керівна система 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)).	
HTTP/HTTPS (Hypertext Transfer Protocol/Secure)	Протокол HTTP/HTTPS використовується для взаємодії між веб-інтерфейсами адміністратора та серверами моніторингу. Забезпечує передачу даних через веб-браузер та можливість віддаленого керування	Клієнтські браузери та веб-сервери
TCP/IP (Transmission Control Protocol/Internet Protocol)	Основний стек протоколів для мережевого зв'язку. Використовується для передачі даних між різними компонентами системи моніторингу, зокрема між агентами та центральними серверами	Усі компоненти системи моніторингу, що спілкуються через мережу.
MQTT (Message Queuing Telemetry Transport)	Легкий та ефективний протокол для передачі повідомлень між пристроями, особливо корисний для IoT-систем та моніторингу великої кількості розподілених пристроїв.	Брокер MQTT, який обробляє та маршрутизує повідомлення, та клієнти, такі як агенти чи інші системні компоненти.
	Протокол орієнтований на простоту використання, невисоке навантаження на канали зв'язку, роботу за умов постійної втрати зв'язку, легке вбудовування у систему. Основне призначення - робота з телеметрією від різних датчиків та пристроїв.	

	Використання шаблону передплатника дозволяє пристроям виходити на зв'язок і публікувати повідомлення, які не були заздалегідь відомі або зумовлені, зокрема, протокол не вводить обмежень на формат даних, що передаються.	
WMI (Windows Management Instrumentation)	Використовується для моніторингу та управління ресурсами в операційних системах Windows. Дозволяє отримувати інформацію про систему та виконувати операції над нею	Агенти WMI на системах Windows та центральний сервер.
	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) та 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.	
Syslog	Використовується для запису подій та повідомлень в системному журналі, що є важливим для моніторингу та діагностики проблем в реальному часі.	Агенти, які генерують системні події, та сервери, які обробляють та аналізують ці події.
RMI (Remote Method Invocation)	Використовується для виклику методів об'єктів, що виконуються на віддалених серверах. Застосовується в розподілених системах моніторингу для взаємодії між компонентами.	Клієнтські та серверні компоненти, які викликають та виконують методи.
	RMI є механізмом виклику методів об'єктів, які виконуються на віддалених серверах у мережевому середовищі. По суті, він дозволяє викликати методи об'єктів, що знаходяться на іншому вузлі мережі, так само, як це робиться для локальних об'єктів. RMI є ключовою технологією для розробки розподілених систем у Java. Хоча RMI є вбудованим механізмом у Java, це не означає, що він обмежений тільки Java. Механізм RMI може бути використаний іншими мовами програмування, такими як Python, за допомогою відповідних бібліотек. Існують такі різновиди RMI: • **Java RMI**: Це офіційний механізм реалізації RMI у Java. Використовується для виклику методів об'єктів, що знаходяться на віддалених серверах у мережі. • **CORBA RMI (IIOP RMI)**: CORBA (Common Object Request Broker Architecture) також має свій варіант RMI, який використовує протокол IIOP (Internet Inter-ORB Protocol) для взаємодії між об'єктами. • **Jini RMI**: Jini є технологією, що реалізує динамічне підключення та взаємодію розподілених компонентів. Вона також використовує RMI для забезпечення комунікації між цими компонентами.	
Custom APIs	Деякі системи моніторингу можуть використовувати власні API для обміну даними між своїми компонентами або з іншими системами	Залежить від конкретної реалізації, але може включати агенти, сервери та інші компоненти системи.

Кожен з цих протоколів має свої унікальні характеристики та застосування, і вибір залежить від конкретних вимог та контексту системи моніторингу.

Висновки.

Ми детально розглянули архітектуру системного моніторингу, розкривши ключові компоненти, які визначають ефективність та функціональність цієї системи.

Спочатку виконано розгорнутий аналіз основних компонентів: агенти, сервери та бази даних. Агенти, які встановлені на об'єктах моніторингу, відповідають за збір та передачу даних, сервери обробляють і аналізують отриману інформацію, а бази даних забезпечують зберігання та доступ до історично-статистичних даних.

Далі, розглянуто дві основні архітектурні моделі - розгорнуту та розподілену. Розгорнута архітектура підходить для менших мереж, де компоненти моніторингу зосереджені на одному сервері. З іншого боку, розподілена архітектура дозволяє розміщувати компоненти на різних вузлах, що дозволяє масштабувати та забезпечувати більшу надійність.

Окрему увагу приділена протоколам зв'язку, які виступають ключовим елементом ефективного обміну даними між компонентами системи моніторингу. Зазначили важливість стандартизації комунікації, надійності та безпеки в обраному протоколі.

Ефективність системного моніторингу залежить від правильного підбору архітектурного підходу, належного вибору протоколів та компетентної реалізації компонентів. Лише узгоджена робота всіх елементів забезпечить надійне та ефективне функціонування системи моніторингу.