

ЛАБОРАТОРНА РОБОТА № 6-7

"Розробка системи моніторингу стану серверного обладнання та сповіщення про інциденти"

Мета: Навчитися моделювати програмні системи за допомогою компонентних діаграм UML, розуміти структуру компонентів, їх взаємодію та залежності.

Хід роботи:

Завдання 1: Діаграма компонентів

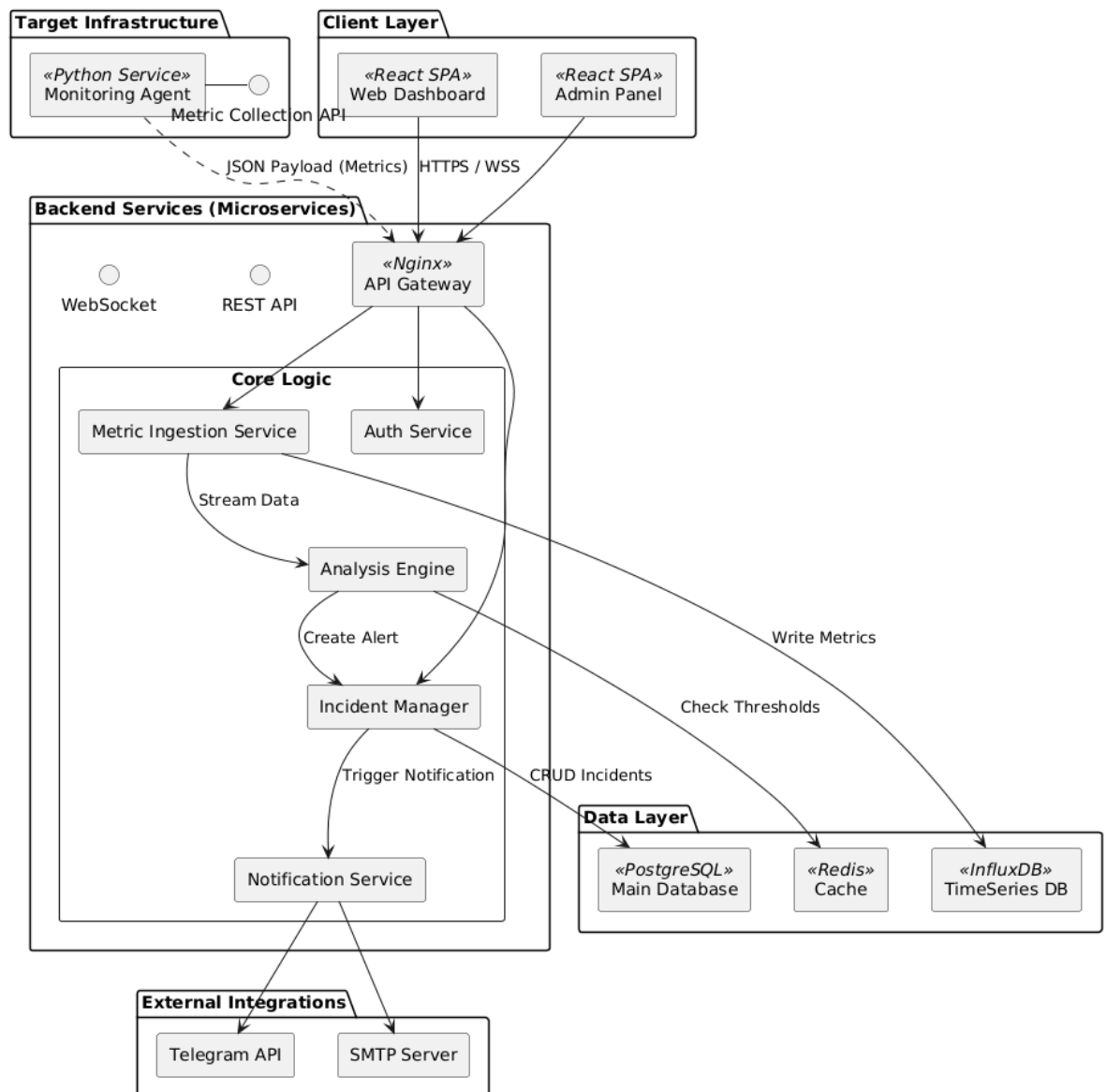


Рис.1. Діаграма компонентів

					ДУ «Житомирська політехніка».22.121.31.000 – Лр6-7						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Ярошинський С.В.			Звіт з лабораторної роботи			Лім.	Арк.	Аркушів	
Перевір.		Левківський В.Л.								1	4
Керівник								ФІКТ Гр. ІПЗ-22-2[1]			
Н. контр.											
Зав. каф.											

Ця діаграма відображає архітектуру системи, розділену на логічні шари: клієнтський рівень, шар API та бізнес-логіки, шар даних та зовнішні інтеграції. Особливістю системи є наявність компонента **Monitoring Agent**, який встановлюється на цільових серверах.

Архітектура системи побудована за мікросервісним (або сервіс-орієнтованим) принципом.

1. **Client Layer:** Представлений веб-інтерфейсом (Web Dashboard), який взаємодіє з системою через REST API для отримання історичних даних та WebSocket для відображення графіків у реальному часі.
2. **Target Infrastructure:** На серверах, що підлягають моніторингу, працює легковаговий Monitoring Agent, який збирає телеметрію (CPU, RAM, Disk) і відправляє її на сервер.
3. **Backend Services:**
 - API Gateway: Єдина точка входу, що маршрутизує запити та балансує навантаження.
 - Metric Ingestion Service: Відповідає за високонавантажений прийом даних від тисяч агентів.
 - Analysis Engine: "Мозок" системи, що перевіряє метрики на відповідність пороговим значенням.
 - Notification Service: Ізольований компонент для роботи з зовнішніми API (Telegram, Email).
4. **Data Layer:** Використовується гібридний підхід: TimeSeries DB (наприклад, InfluxDB) для швидкого запису метрик і реляційна PostgreSQL для зберігання користувачів та інцидентів. Redis використовується для кешування гарячих даних.

		Ярошинський С.В.			ДУ «Житомирська політехніка».22.121.31.000 – Лр6-7	Арк.
		Левківський В.Л.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2: Діаграма розгортання

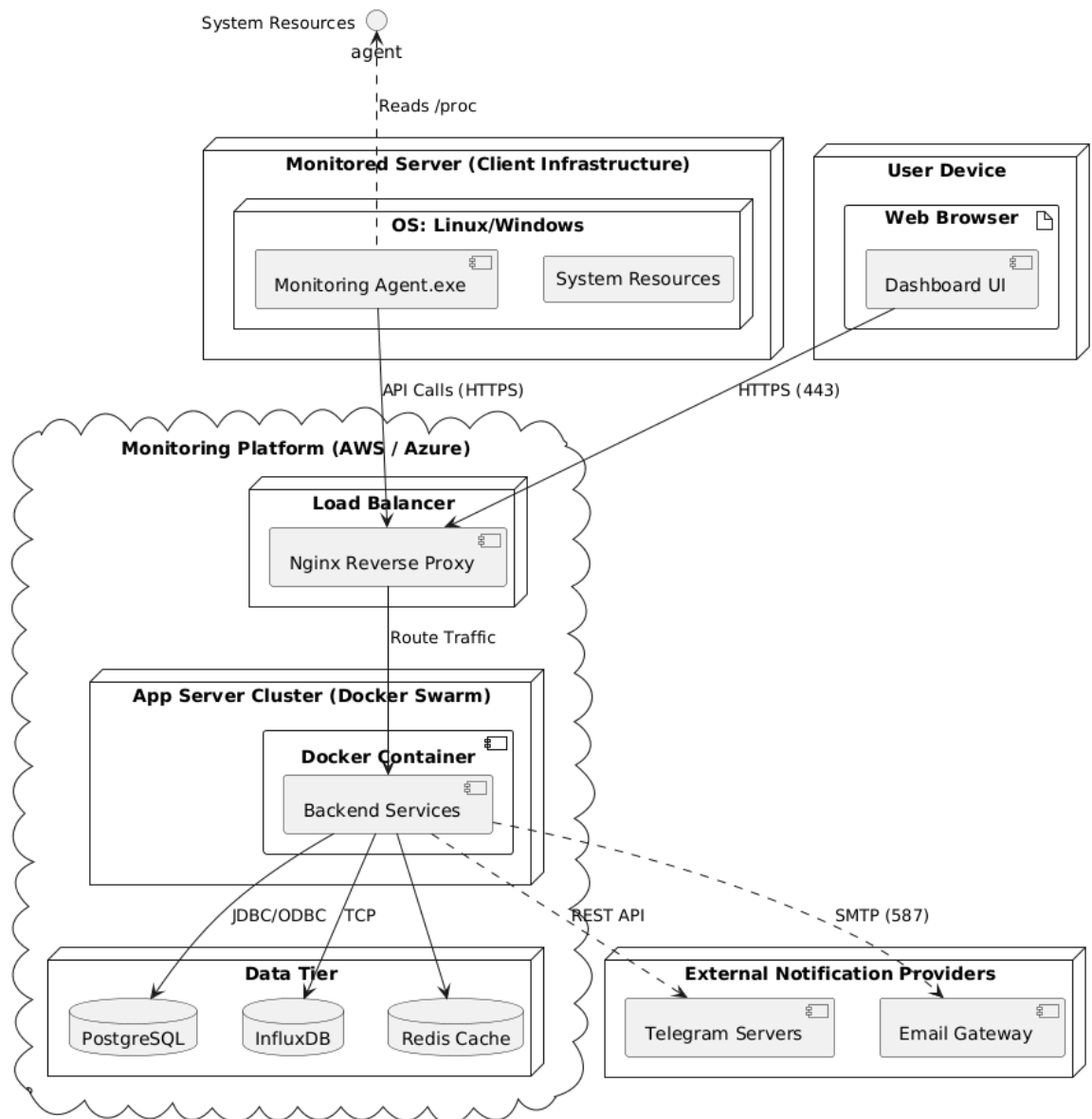


Рис.2. Діаграма розгортання

Діаграма відображає фізичну топологію розміщення програмних артефактів на обчислювальних вузлах (серверах, контейнерах).

Система розгортається у хмарному середовищі (Cloud) для забезпечення високої доступності.

- Monitored Server:** Це зовнішні сервери (клієнтські), на яких встановлено лише один виконуваний файл – Monitoring Agent. Він зчитує системні ресурси та передає їх у хмару через захищений канал HTTPS.
- Monitoring Platform:** Ядро системи працює в контейнеризованому середовищі (Docker).

- Вхідний трафік обробляється балансувальником навантаження Nginx.
- Бізнес-логіка виконується в кластері серверів застосунків (App Server Cluster), що дозволяє горизонтально масштабувати систему при збільшенні кількості підключених серверів.
- Рівень даних (Data Tier) винесено на окремі оптимізовані інстанси баз даних.

3. **User Device:** Користувачі отримують доступ до системи через браузер, завантажуючи статичні файли фронтенду.

Висновок: у ході виконання лабораторної роботи ми навчилися моделювати програмні системи за допомогою компонентних діаграм UML, зрозуміли структуру компонентів, їх взаємодію та залежності.

		Ярошинський С.В.			ДУ «Житомирська політехніка».22.121.31.000 – Лр6-7	Арк.
		Левківський В.Л.				
Змн.	Арк.	№ докум.	Підпис	Дата		4