



### Компоненти:

- ✓ Користувач (User): уособлює учня, вчителя або адміністратора, які взаємодіють із системою через вебзастосунок або Telegram-бот.
- ✓ Front-end (Хостинг статичного сайту): S3 Bucket із підключеним CloudFront для розміщення вебінтерфейсу React-додатку. Забезпечує швидку доставку контенту користувачам через мережу доставки вмісту (CDN).
- ✓ Elastic Load Balancer (Балансувальник навантаження): розподіляє вхідні запити між кількома інстансами серверів бекенду для забезпечення масштабованості та високої доступності системи.
- ✓ Back-end (Мікросервіси на EC2): група мікросервісів (Users Service, Tests Service, Notation Service, GeneralData Service, WorkTime Service) розгорнутих на віртуальних машинах EC2 з автоматичним масштабуванням. Відповідають за бізнес-логіку, обробку тестів, облік статистики та управління користувачами.

- ✓ PostgreSQL (Керована реляційна база даних): Amazon RDS на базі PostgreSQL для зберігання структурованих даних: інформації про користувачів, результати тестів, нотатки та активність.
- ✓ ElastiCache (Redis) (Кеш-сервер): використовується для кешування популярних запитань тестів, зменшення навантаження на базу даних та прискорення відповіді системи.
- ✓ SQS Queue (Черга повідомлень): сервіс чергування повідомлень для асинхронної обробки задач, таких як надсилання сповіщень через Telegram-бот або оновлення статистики після проходження тестів.
- ✓ Telegram Bot Service (Окремий сервіс бота): мікросервіс, що обробляє повідомлення від користувачів у Telegram, надсилає сповіщення, отримує статуси виконання завдань.
- ✓ CloudWatch (Моніторинг і логування): система моніторингу продуктивності, збору логів та відправки сповіщень про помилки або збої в роботі сервісів.

### **Робочий процес (Workflow):**

- ✓ Взаємодія користувача: кінцевий користувач надсилає запити через браузер або Telegram-клієнт до застосунку, розміщеного у Front-end на S3.
- ✓ Обробка на бекенді: запити потрапляють через Elastic Load Balancer до відповідних мікросервісів на EC2.
- ✓ Інтеграція сервісів мікросервісами:
  - звернення до PostgreSQL для зберігання або отримання даних (результати тестів, інформація про користувачів);
  - звернення до ElastiCache Redis для даних кеш (питання тестів, короткочасні результати);

- публікація подій у SQS Queue для асинхронної обробки (оновлення статистики, відправка сповіщень через Telegram-бот);
- спілкування із Telegram Bot Service для обробки повідомлень користувачів та надсилання завдань;
- логування важливих подій та помилок у CloudWatch для аналізу та моніторингу системи.