

УДК 004.07

Болож О. – ст. гр. СНМ-51

*Тернопільський національний технічний університет імені Івана Пулюя*

## **ОГЛЯД SERVERLESS АРХІТЕКТУРИ ТА ЇЇ ПЕРЕВАГИ**

Науковий керівник: старший викладач Шимчук Г.

Bolozh O.

*Ternopil Ivan Puluj National Technical University*

## **OVERVIEW OF SERVERLESS ARCHITECTURE AND ITS ADVANTAGES**

Supervisor: Senior Lecturer Shymchuk G.

Ключові слова: serverless, serverless architecture, BaaS, FaaS, cloud platform, AWS Lambda, Google Cloud Functions, DB, Microsoft Azure, back-end

Key words: serverless, serverless architecture, BaaS, FaaS, cloud platform, AWS Lambda, Google Cloud Functions, DB, Microsoft Azure, back-end

Serverless архітектура – це спосіб створення та запуску додатків, який зменшує потребу в управлінні ресурсами. Такий підхід дозволяє виконавцям запускати додатки без управління фізичними серверами.[1]. Це дозволяє хмарним провайдерам виконувати код, розподіляючи ресурси та масштабуючи інфраструктуру в залежності від потреб, які виникають. Таким прикладом може бути розміщена на певній хмарній інфраструктурі база даних, яка з часом розростається і потребує більше ресурсів, таких як оперативна пам'ять чи кількість ядер процесора. Тут потрібно особисто перевіряти метрики і вручну підбирати тарифи, а у разі використання serverless – масштабування проходить автоматично, безболісно і менш затратно.

Прикладами платформ що надають можливості впровадження безсерверної архітектури є AWS Lambda, Azure Functions, Google Cloud Functions тощо.[1]

Serverless архітектура функціонує наступним чином: сервери дозволяють користувачам спілкуватися з додатком і отримувати доступ до його бізнес-логіки, але управління серверами займає значний час і ресурси. Командам доводиться обслуговувати серверне обладнання, дбати про оновлення програмного забезпечення та безпеки, а також створювати резервні копії на випадок збоїв. Впроваджуючи безсерверну архітектуру, розробники можуть перекласти ці обов'язки на стороннього постачальника, що дозволить їм зосередитися на написанні коду додатків.[2]

Однією з найпопулярніших безсерверних архітектур є "Функція як послуга" (Function as a Service, FaaS), де розробники пишуть код своїх додатків як набір окремих функцій. Кожна функція виконує певне завдання, коли її запускає певна подія, наприклад, вхідний електронний лист або HTTP-запит. Після звичайних етапів тестування розробники розгортають свої функції разом з тригерами в акаунті хмарного провайдера. Коли функція викликається, хмарний провайдер або виконує її на працюючому сервері, або, якщо сервер наразі не працює, запускає новий сервер для виконання функції. Цей процес виконання абстрагований від розробників, які зосереджені на написанні та розгортанні коду програми.[2]

Ще однією послугою що використовується в Serverless є "Backend як послуга" (Backend as a Service) – такий архітектурний підхід позбавляє розробників від деяких завдань пов'язаних з серверною розробкою, таких як хостинг хмарного сховища та управління базами даних.

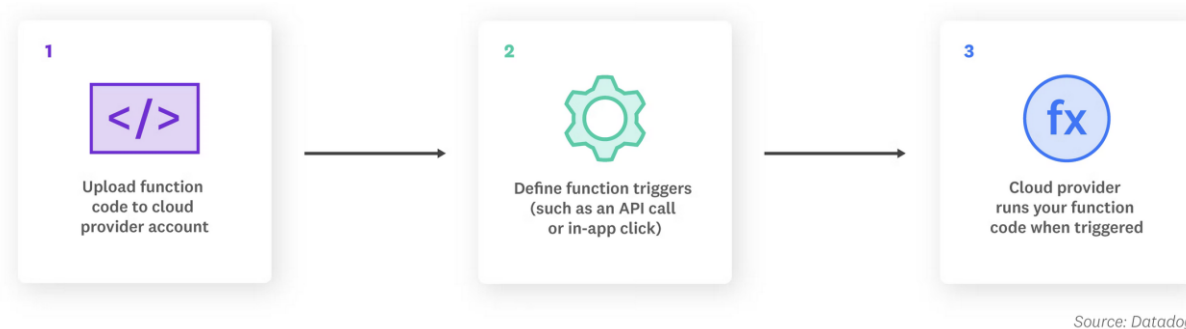


Рисунок 1 – Принцип функціонування FaaS

Основні переваги Serverless архітектури:

- нижча вартість, якщо врахувати зменшення потреби в персоналі DevOps для налаштування та підтримки інфраструктури з плином часу, а також оплату за використання, що означає, що періоди з низьким трафіком можуть коштувати \$0 за годину;[3]
- швидкість розробки різко зростає, оскільки набагато простіше збирати рішення разом і розгортати їх у виробництво;[3]
- стабільність оскільки хмара керує всіма сервісами, ризик що щось піде не так – мінімальний або ж відсутній взагалі;
- готовність до різких навантажень, як вже було описано раніше Serverless масштабується згідно навантажень, які виникають.

Деякі сценарії в яких може бути використана безсерверна архітектура:

- реалізація асинхронної обробки повідомлень у додатках;
- обробка даних для забезпечення потужного машинного навчання;
- створювати додатки з високою затримкою в реальному часі, наприклад, мультимедійні додатки, для автоматичного розподілу пам'яті та складної обробки даних;
- обслуговування непередбачуваних робочих навантажень для швидкозмінних потреб розробки, додавання функцій на вимогу клієнтів та інших складних потреб у масштабуванні;
- динамічно змінювати розмір зображень або перекодувати відео та спрощувати обробку мультимедійних даних на різних пристроях;
- для побудови спільної системи диспетчеризації доставки;
- для додатків на основі Інтернету речей (IoT) і розумних пристроїв;
- у прикладних модулях на основі сценаріїв трансляції відео в реальному часі;
- реалізація безперервної інтеграції (CI) та безперервної доставки (CD).

Література:

1. Solanki J. Serverless Architecture – What It Is? Benefits, Limitations & Use cases [Електронний ресурс] / Jignesh Solanki. – 2023. – Режим доступу до ресурсу: <https://www.simform.com/blog/serverless-architecture-guide>.
2. Serverless Architecture Overview [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.datadoghq.com/knowledge-center/serverless-architecture>.
3. McCumskey G. A Guide to Serverless Architecture [Електронний ресурс] / Gareth McCumskey. – 2022. – Режим доступу до ресурсу: <https://www.serverless.com/blog/serverless-architecture>.