Голомб Ю. О. Ігнатович А. О.

Національний університет "Львівська політехніка",

кафедра електронних обчислювальних машин

**Використання платформи Amazon Web Services для побудови високопродуктивних веб-застосунків**

*© Голомб Ю. О., Ігнатовчи А. О., 2019*

**Розглянуто проблему побудови високопродуктивних веб-застосунків, проаналізовано типову архітектуру веб-застосунків, розглянуто структурну схему веб аплікації побудованої на Amazon Web Services.**

**Ключові слова: Amazon Web Services, Serverless, веб-застосунки.**

**Building high loaded web-application is always a hot topic. Analyzed typical architecture of web-applications. Structure schema of web-application build on Amazon Web Services infrastructure.**

**Keywords: Amazon Web Services, Serverless, web-applications.**

**Вступ.** Ми живемо в епоху, коли веб-застосунки є невід’ємною частиною нашого життя. Майже для кожної настільної аплікації існує аналог який в тій чи іншій мірі є аналогом. Кожен успішний бізнес має свої веб-сторінки на просторах Інтернету. Вони можуть використовуватися в рекламних цілях або бути головною частиною бізнес-плану. Станом на 2017 рік загальні витрати на маркетинг в Інтернеті в Сполучених Штатах Америки було витрачено більше 88 мільярдів доларів[1]. В 2018 році загальна кількість користувачів Інтернет сягла позначки 3,9 мільярдів[2]. Кількість користувачів найпопулярнішої соціальної мережі сягає 360 мільйонів. Щоб обробляти, зберігати, та аналізувати таку величезну кількість даних які щодня проходять через веб-застосунки подібного масштабу необхідно використовувати найсучасніші досягнення в архітектурі веб-аплікацій.

**Стан проблеми.** За останні роки кількість відвідувачів веб-застосунків значно зросла як і кількість функціоналу які вони надають. Класична монолітна архітектура не здатна задовольнити вимоги високонавантажених аплікацій, адже архітектура такого роду дозволяє масштабуватися тільки вертикально, тобто збільшуванням швидкодії процесора та обсягу оперативної пам’яті. Іншою проблемою сучасних веб-застосунків монолітної архітектури є централізованість і як наслідок наявність єдиної точки збою. Якщо виходить з ладу сервер то подальша робота веб-застосунку є неможливою.

Для вирішення проблеми масштабування та проблеми єдиної точки збою необхідно використати новий підхід який полягає у створенні аплікації з використанням безсервених технологій, які дозволяють горизонтальне масштабування, тобто не збільшення продуктивності одного сервера а збільшення кількості обробляючих вузлів. Хмарні обчислення володіють багатьма переваагами в порівнянні з традиційними рішеннями для побудови інфраструктур сервісів[3].

**Постановка задачі**. Розглянути можливості використання платформи Amazon Web Services для побудови високопродуктивних веб-застосунків.

**Розв’язання задачі.** Для розв’язку поставленої задачі було вирішено спершу оглянути всі можливості які надають Amazon Web Services для створення веб-застосунків.

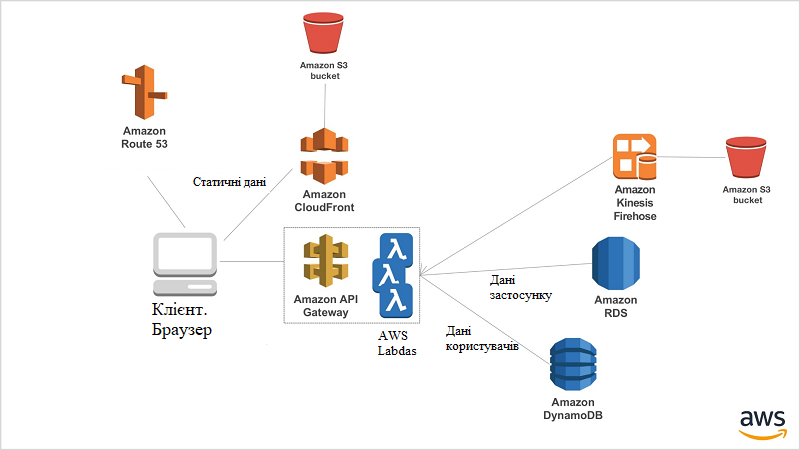
Ключовою ланкою веб-застосунка є обчислювальна машина яка виступає в якості веб-сервера. В якості такого роду обчислювальних пристроїв Amazon Web Services пропонують

* Amazon Elastic Compute Cloud (Amazon EC2) – гнучкий сервіс який пропонує створення віртуального сервера з гнучкими параметрами (швидкодія процесора, кількість оперативної та постійної пам’яті)
* AWS Lambda – безсерверна технологія, надає можливість створювати спеціальні функції (лямбди), кожна з яких виконується при мережних викликах. При використанні данного підходу оплачується тільки кількість часу який лямбда пропрацює на відміну від сервера якому потрібно працювати потрібно.
* AWS Fargate та Amazon ECS – сервіси для оркестрації контейнерів, таких як наприклад Docker-контейнер. Дозволяють запускати, керувати та масштабувати контейнери для відповідності навантаженню.

Варіанти обчислювальних машин надані Amazon Web Services дозволяють забезпечити найрізноманітніші вимоги щодо веб-серверів. Існують варіанти як для класичної монолітної архітектури(ЕС2) так і для високопродуктивних мікросервісних та безсерверних архітектур. Усі з запропонованих варіантів надають широкий спектр по масштабуванню як у кількості так і у якості обчислювальних одиниць.

Наступною але не менш важливою ланкою розглянемо пропоновані рішення баз даних які існують в Amazon Web Services. Якщо є необхідність в реляційних базах даних то можна використати Amazon Aurora, Amazon RDS, Amazon Redshift ці рішення відрізняються в базах даних які можна на них використовувати та підтримують всі найбільш популярні як: MySQL, PostgreSQL, MariaDB, Oracle, SQL Server[4]. Вищеподані сервіси забезпечують простоту адміністрування через панель адміністратора, високе масштабування, надзвичайно високу доступність та надійність яка сягає 99,5%.

Також для повноцінного існування веб-застосунку є необхідним файлове сховище де зберігатимуться статичні файли, як наприклад медіа-файли. Для цього типу рішень існує велика кількість рішень для різних ситуацій. Пропонуються класичні сховище загального призначення як S3, високодоступне та високострокове файлове сховище яке забезпечує нормативні вимоги(Amazon Glacier), повністю керований сервіс резервного копіювання який спрощує систему резервного копіювання (AWS Backup).

Типова архітектура для веб-застосунку побудованого на Amazon Web Services

Згідно з такою архітектурою функції системи доменних імен(DNS) виконує сервіс Amazon Route 53 який надає підтримку протоколів IP4 та IP6, та є високомасштабованою та виключно надійною.

Браузер клієнта отримує статичні дані веб-сторінки, так звану front-end частину через Amazon CloudFront який являє собою безпечну мережу доставки контенту з широкими можливостями програмування. Даний сервіс дозволяє розповсюджувати статичні та медіа дані по всьому світу з низькою затримкою та високою швидкістю. Самі ж дані зберігаються на Amazon S3 сховищі.

HTTP виклики з клієнтської частини до API серверної частини проходять через Amazon API Gateaway, це керований сервіс який надає послуги для створення, публікації, та забезпечення безпеки API в будь-яких масштабах. Далі виклики переадресовуються на так звані лямбда функції, які створюються по необхідності та в разі потреби вимикаються. Лямбди виконують операції над даними в базах даних, якими керують такі сервіси: Amazon DynamoDB та Amazon RDS.

Веб застосунок побудований за поданою архітектурою здатен легко масштабуватися в залежності від вимог за декілька команд виконаних з консолі розробника в Amazon Web Services. При коректних налаштуваннях, якщо певна ланка не справлятиметься з вхідним навантаженням то в автоматичному режимі буде піднята копія ланки яка візьме на себе частину викликів. Таким чином навіть під час збільшення навантаження в короткий термін, який би був критичним при монолітній архітектурі, та вивів би її з ладу, Amazon Web Services забезпечить веб-застосунок необхідною кількістю продуктивності для забезпечення доступності всіх її частин.

**Висновки**. В роботі розглянуто можливості використання платформи Amazon Web Services для побудови високопродуктивних веб-застосунків. Сервіси даної платформи як найкраще підходять для створення високопродуктивних веб-застосунків з використанням сучасних технологій.

**Література**

1. David Silverman, Brian Gaffney, IAB internet advertising revenue report. New York. 2018
2. Number of internet users worldwide from 2005 to 2018 [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>.
3. Эммерих В. Конструирование распределенных обьектов. Методы и средства програмирования интероперабельных обьектов. Пер с англ, - Мир - 2012 - 510с.
4. Getting Started with AWS, 2012. – 50 с.
5. Nicolas Dao. THE SERVERLESS SERIES What Is Serverless?... – [Електронний ресурс]. – Access mode: https://hackernoon.com/the-serverless-series-what-is-serverless-d651fbacf3f4