

**ОСОБЛИВОСТІ ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ АРХІТЕКТУРИ  
AWS**

Немов М. Р.

Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ»  
Науковий керівник: Узун Д. Д.

**Актуальність.** В сучасному світі, де обчислення в хмарі визначають нові стандарти ефективності та масштабованості, Amazon Web Services стали важливим інструментом для багатьох компаній та організацій. Забезпечення надійності архітектури в AWS є ключовим аспектом в розробці та управлінні інфраструктурою, наприклад для реалізації веб-сайтів, обчислювальних потужностей для навчання нейромереж або Big Data, серверів для різноманітних додатків. У зв'язку з цим виникає необхідність дослідження особливостей та стратегій забезпечення надійності в середовищі AWS.

**Метою** даної роботи є розгляд основних аспектів та особливостей, що визначають надійність архітектури AWS. Доступність і запит до сайтів або серверів не завжди може бути задоволена, оскільки зміни для інфраструктури не завжди можуть бути гнучкими й швидкими при ручному налаштуванні. А при виході з ладу сервера архітектура може повністю стати цілком неприцездатною. Дослідження спрямоване на визначення тих технічних аспектів, які дозволяють AWS забезпечити доступність, цілісність даних.

**Основні положення.** При проектуванні інфраструктури важливо уникати єдиних точок відмови, де відмова одного елемента може призвести до повної недоступності системи. AWS надає різноманітні інструменти для забезпечення надійності. Сервіси, такі як Amazon EC2 Auto Scaling та Amazon Load Balancer, забезпечують доступність та надійність інфраструктури. Amazon EC2 Auto Scaling являє собою автоматизований процес регулювання розміру інфраструктури, який дозволяє автоматично збільшувати або зменшувати обсяг ресурсів в залежності від поточного навантаження системи для оптимізації продуктивності й витрат. Тут використовується загальний підхід до обчислювальних потужностей у хмарних провайдерах «Design services, not servers», де сервер – це одноразовий, витратний, ресурс. Найкраща практика це, коли сервери не повинні зберігати в собі важливі дані. Для них важливо налаштувати розгортку програми або утиліти для нового віртуального сервера і відновлення функціонала. Балансувальник навантаження Amazon Load Balancer дозволяє з'єднати між собою дві різні групи серверів через себе. Load balancing є технікою розподілу навантаження між різними компонентами системи, щоб забезпечити оптимальний розподіл ресурсів і

покращити ефективність роботи системи, уникнути перевантажень та забезпечити надійність. Для забезпечення доступності баз даних RDS використовується snapshot – це знімок бази даних, який фіксує стан даних в конкретний момент часу, що дозволяє вам відновлювати базу даних до певного стану в разі втрати даних чи виникнення інших проблем. Amazon RDS надає можливість використовувати репліки читання для забезпечення доступності. Read replica копіює оновлення з основного екземпляра бази даних, і при відмові основного сервера, запити автоматично переадресовуються до репліки, забезпечуючи безперервну роботу системи. У разі повної відмови інфраструктури її можна відновити у спосіб «IaC». Тобто описати інфраструктуру, щоб її компоненти автоматично і швидко були перестворені чи навпаки – бути вивільненими для оптимізації коштів. Для цього використовується сервіс Amazon CloudFormation, який дозволяє описувати інфраструктуру у вигляді коду та автоматично створювати та управляти ресурсами AWS.

**Висновки.** Ці практики дозволяють бути інфраструктурі не тільки надійною та гнучкою, а й більш оптимальною у витратах. Ці особливості включені й для інших хмарних провайдерів як Azure або GoogleCloud, використовуючи їх аналоги AWS. Доступність і надійність відіграють ключову роль, оскільки можуть вплинути на досвід користувача використання вашого додатка в бізнес цілях.

#### Список літератури.

1. *Rajesh Daswani* AWS Certified Cloud Practitioner Exam Guide: Build your cloud computing knowledge and build your skills as an AWS Certified Cloud Practitioner. // Packt Publishing;
2. AWS Well-Architected Framework. *Amazon Web Services*. URL – <https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/index.en.html> (дата звернення: 12.11.2023).

#### Відомості про авторів

Немов Микита Русланович, студент кафедри комп'ютерних систем, мереж і кібербезпеки, НАУ «ХАІ», m.r.nemov@student.csn.khai.edu

Узун Дмитро Дмитрович, доцент кафедри комп'ютерних систем, мереж і кібербезпеки, к.т.н., доцент, d.uzun@csn.khai.edu