UDC 004.4 **doi: 10.26906/SUNZ.2024.1.098**

O. Kolesnikov, G. Golovko, V. Yastreba, Ye. Piatyntsev

National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine

LEVERAGING CLOUD TECHNOLOGIES AND SERVERLESS ARCHITECTURE FOR EFFICIENT WEB DEVELOPMENT: A CASE STUDY FROM REAL-WORLD APPLICATION

Abstract. This article provides a review of modern cloud solutions and serverless architecture using Backend as a Service (BaaS) and Function as a Service (FaaS) architecture as an example. In the scope of the article, a parallel is drawn between the consistently growing computing power and cloud technologies' growing popularity and availability for business. The results of an analytical review include a list of the most popular cloud providers from leading corporations and a comparison of low-level and high-level cloud technologies. The advantages and disadvantages of Google Cloud Platform (GCP) and Google Firebase are presented, where GCP is a low-level cloud provider and Firebase is a high-level cloud provider. The importance of understanding the context and specifics of the project when choosing cloud project solutions is emphasized which helps to maintain a balance between flexibility and development efficiency. The study introduces the practical utilization of the Supabase cloud platform for the development of a modern web application. The article convincingly proves the actuality of using Supabase for the development of an information system to optimize the modern personnel recruitment process, indicating specific advantages. An example of Supabase Edge Functions usage to generate feedback using the OpenAI Completions API and the Deno software platform is presented. The article convincingly proves that the use of cloud technologies is a modern strategy for building flexible, efficient, and scalable information systems. The advantages of the usage of the provision of infrastructure provided by world industry leaders are summarized.

Keywords: systems, information systems, information technologies, cloud providers, serverless architecture, GCP, Firebase, Supabase, efficiency, artificial intelligence9
.

Introduction

In today's world, due to the rapid development of technology, computing power is becoming cheaper and more powerful day by day. The growth of computing capabilities of supercomputers and servers is especially noticeable [1] (Fig 1).

In parallel with the growth of computing power, network technologies were also actively developed, which made it possible to significantly increase the speed of the Internet around the world, as well as make communication between hardware more stable.

One of the derivatives of the development of computing power and network technologies is the emergence of cloud technologies, which over the last decade have become an integral part of the best practices of modern software development.

The emergence of the concept of cloud computing marked a fundamental transformation of the concept of building the infrastructure of projects.

Organizations and companies have gained the ability to use significant computing and storage resources without the need for large initial investments in server infrastructure.

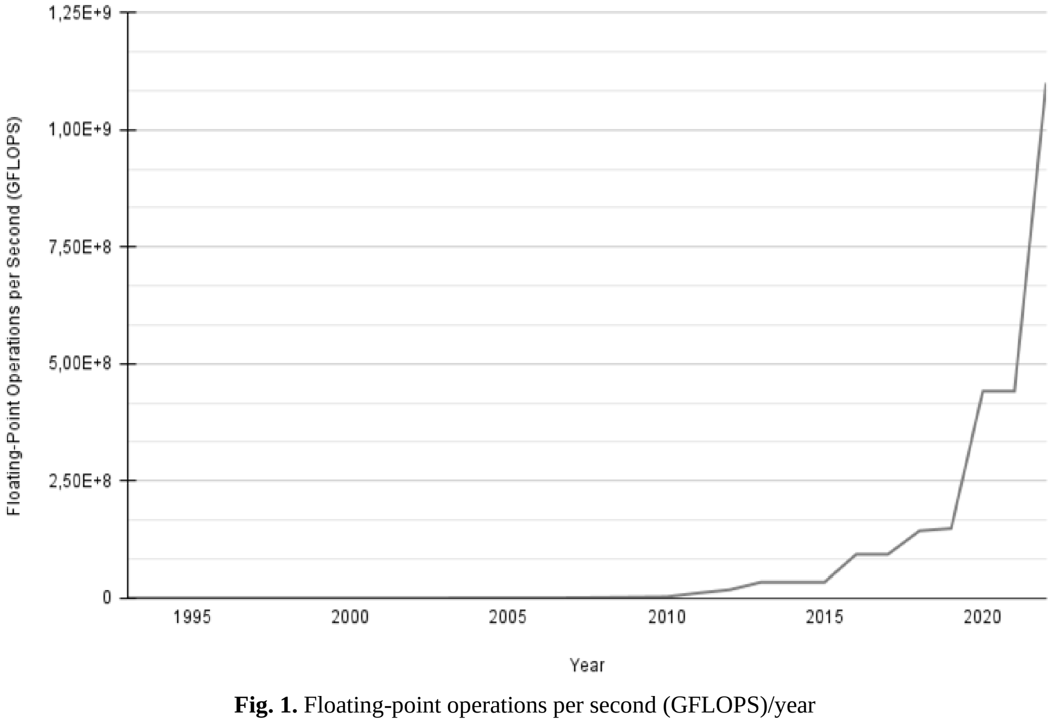

Fig. 1. Floating-point operations per second (GFLOPS)/year

Cloud technologies have gained particularly active use in web development, where serverless architecture has become one of the components for the effective and scalable development of web projects.

The introduction of serverless architecture has led to a move away from the use of traditional servers, which has opened up new opportunities for businesses and developers [2].

In the context of modern software development approaches, the concept of serverless architecture is often misunderstood.

It is commonly assumed that "serverless" means the complete absence of servers, but this is not true.

The proper meaning of the serverless architecture implies a change in the management and use of server resources.

In a traditional concept, teams are required to deal not only with the development of software code and business logic but also with the configuration and maintenance of the server infrastructure. This includes managing servers, ensuring their continuous operation, load balancing, and scaling resources according to the needs of the application.

However, a serverless architecture changes this approach by renouncing the need to directly manage servers. Instead, developers can focus solely on writing and deploying code using computing resources that are automatically allocated and scaled by the cloud provider. This provides efficiency, flexibility, and speed of deployment, reducing the burden on developers and allowing them to focus on creating an innovative product [2].

This article delves into the practical application of cloud technologies and serverless architecture in modern web development, using a real-world example that will provide valuable insight into the practical benefits and drawbacks of cloud technology usage.

The research purpose

The purpose of this article is to do a brief research on the existing cloud providers and solutions, analyze the pros and cons, and demonstrate a real-world example of serverless architecture usage in the information system to automate daily tasks in the recruitment workflow.

Analytical review of existing cloud solutions

The most famous cloud providers from global corporations are:

- Amazon Web Services (AWS);
- Google Cloud Platform (GCP);
- Microsoft Azure.

All three cloud providers are undisputed industry leaders that provide services that give confidence in the reliability of the infrastructure.

Cloud providers provide a wide variety of tools with different levels of abstraction. Depending on the level of abstraction, the ratio of flexibility to the speed of configuration and deployment changes. Lower-level tools are more flexible but more complex to configure, and high-level tools are less flexible but extremely easy and quick to configure which is ideal for Minimal Viable Product (MVP) or software development without complex server architecture requirements [2].

Google, in addition to the Google Cloud Platform, also provides higher-level cloud solutions named Firebase platform. Both Google Cloud Platform and Google Firebase have their advantages and disadvantages depending on the project. Firebase is a Backend as a Service (BaaS) platform that provides higher-level abstraction services that allow building mobile and web applications quickly and with minimal backend code and infrastructure setup.

For example, Firebase provides cloud database MongoDB, hosting, user auth, and other features [3].

Usage of low-level cloud platforms such as Google Cloud Platform include the advantages [3]:

1. Flexibility: GCP provides granular resource management, which is important for complex projects.
2. A wide range of services: from virtual machines (VMs) to network services and data storage.
3. Scalability: Suitable for large projects with complex architecture.

Disadvantages of low-level cloud platforms such as GCP are [3]:

1. Complexity of management: Requires more in-depth knowledge in the field of cloud resource management.
2. Higher costs: Some services may incur higher costs compared to more abstract platforms.

The advantages of high-level cloud platforms are [4]:

1. Simplified development: Firebase provides high-level tools that make it easy to build and manage applications.
2. Rapid deployment: Ideal for projects that require rapid development without complex infrastructure management.
3. Built-in features: Authentication, real-time databases, analytics, and more.

Considered disadvantages of high-level cloud platforms are [4]:

1. Limited flexibility: Firebase can be restrictive for very complex or specific project needs.
2. Platform Dependency: Focusing on using Firebase may lead to difficulties migrating to other platforms in the future.

The choice of platform always depends on the specifics of the project, but in the case of MVP project development, or a project that does not require complex architectural solutions, a platform like Firebase is an ideal solution, because, in addition to saving time, quite often a free quota is enough to test an idea or to make project working at the initial stage, which allows you to focus resources on the business logic of the project [3] (Fig. 2).

Cloud solutions are not ideal and might have different issues such as security or vendor-lock bottleneck, but at the same time, infrastructure is managed by industry leaders which helps to be sure that infrastructure is made following best practices.

The implementation and maintenance of in-house infrastructure require a significant amount of resources which might be critical for startups and small-grade projects [3].

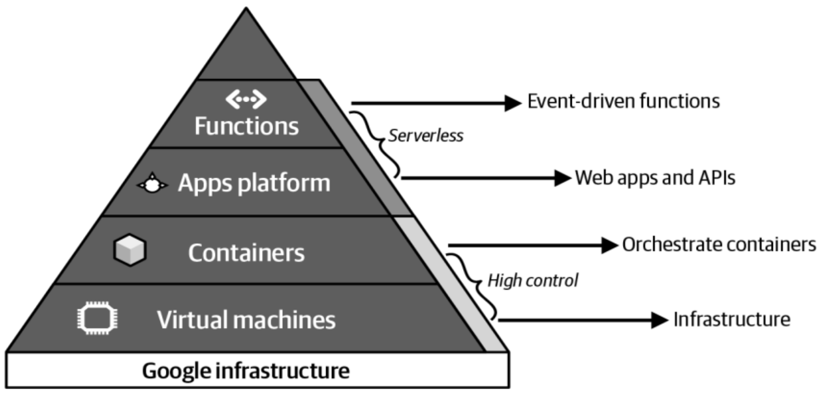

Fig. 2. Google infrastructure hierarchy

Main part

Based on the discovery made in the article "The objective need to implement an information system to automate daily tasks in recruitment workflow" the minimum necessary functionality for the implementation of the information system to automate daily tasks in the recruitment workflow is [7]:

- authentication;
- competency matrices management module;
- candidate management module;
- candidate interview module;
- a module for generating feedback for candidates based on interview results.

Based on the listed functionality, it is possible to conclude that the necessity:

- use of a modern DBMS for data storage, retrieval, and management;
- implementation of authentication or use of ready-made solutions;
- the presence of an API layer, which will avoid direct interaction of the client side with the database, which is considered anti-pattern.

The most effective way to implement a web application with the above-mentioned requirements is to use high-level Backend as a Service (BaaS) tools. BaaS is the serverless cloud model that helps the development team to deploy server-side logic without or with a minimal amount of coding. Usually, it might be a simple user interface [5].

The most popular BaaS solutions for web development are Firebase or Supabase. Supabase is a modern open-source platform that positions itself as an alternative to Google's Firebase. This platform provides a range of features and functionality that may be needed when creating modern web applications [8].

Architecturally Supabase is relative to the Backend as Service and Function as Service. FaaS is a serverless cloud model with a way to execute modular parts of code like specific functions on the edges. In specific cases like the development of the information system which should be proof of concept, Supabase is a platform tool that will help radically increase Time to Market (metric) [8].

One of the best advantages of the Supabase is its community-driven approach. It's open-sourced and is actively maintained by people across the world.

Also, Supabase has a much higher threshold of free quotas compared to Firebase which means that the product might live in the free tier for a longer time during growth. But even when the product is ready to scale the flexible pricing politics at the platform allow continuing growth without radical resource consumption increase.

Supabase provides a bunch of immediately ready-to-use solutions to save time during the development of the aforementioned application, especially the next ones [8]:

1. First things first Supabase deploys for free relational PostgreSQL which is manageable from the Supabase Dashboard with a lot of helpful interactive tools to manage the database. But the most important thing is that it still stays just a regular PostgreSQL database under the hood which might be used in the classic backend or to which it's possible to connect from any favorite software which means that we're not vendor-locked in Supabase.
2. Supabase Authentication provides ready-to-use free authentication functionality with various configurations, a flexible API for its use, and even ready-made UI elements. Various authentication methods are supported: starting with the classic method with a login and password and finishing with authentication with third-party providers such as Google, GitHub, and others.
3. One of the key advantages of Supabase is its authorization system, which uses PostgreSQL's row-level security (RLS) access control mechanism. This provides the ability to fine-tune accesses using specific conditions, thus allowing granular control of data access by different categories of users.
4. Edge Functions infrastructure which will help with OpenAI Completions API endpoint creation.
5. Supabase offers a convenient administration panel interface that allows you to manage services, database, and environment variables and perform monitoring, etc.
6. Supabase automates the process of APIs and related documentation creation using a database schema as a baseline. This provides convenience in interacting with the database through GraphQL or RESTful API endpoints.
7. Using Supabase Codegen to automatically generate TypeScript types ensures the accuracy of syncing database schemas with the client code, which simplifies development and reduces the risk of errors.

8. JavaScript SDK makes integration of the Supabase into the client-side code pretty quick and smooth. JavaScript SDK comes with predefined TypeScript types which in pair with Supabase codegen helps to avoid mistakes.

Supabase is not designed to resolve all problems by itself both with paid and free plans. For example, one of the biggest drawbacks of the Supabase usage on the free tier is the unavailability of database backup management from the user interface. But it's a feature that is pretty easy to do manually or even automate by CI/CD pipelines, for example using GitHub Actions. It's pretty important to choose solutions that do not lock the team from doing fallbacks. Supabase always gives the ability to use the fallback solutions where Supabase can't provide solutions from the box which makes this cloud provider a great choice from the scalability and maintenance point of view.

Features provided by Supabase are granular and useful in a standalone manner.

That means that each feature might be integrated or replaced down the road on demand which makes Supabase usage even more flexible (Fig 3).

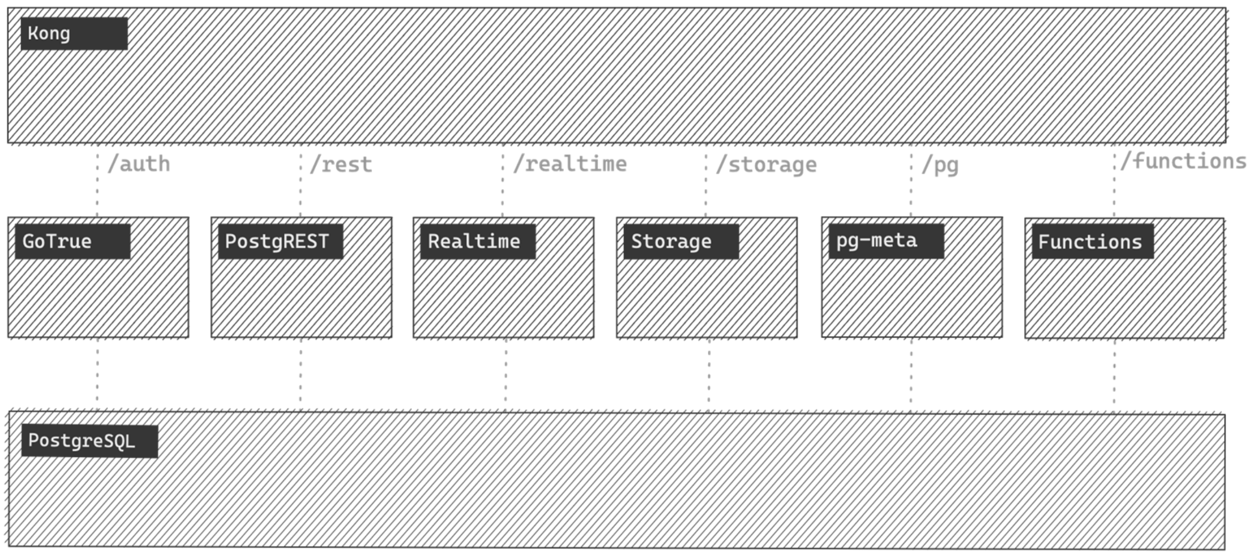

Fig. 3. Supabase architecture

Supabase Edge Functions and OpenAI Completions API. Edge functions are serverless JavaScript/TypeScript functions that are globally distributed across the world in a similar way to CDN. The main benefit of distributed globally serverless functions is reduced latency for final users of the information systems because of evaluation in data centers that are located as close to the user as possible. Edge functions might be used to achieve any common backend requirements. The reduced latency is critical in cases where performance is critical. Edge functions are a great fit for building automatic feedback generation using large language models. ChatGPT-like user experience is the best practice which gives users the most comfortable way of communication with LLM [6].

Edge functions are the implementation of the Function-as-a-Service (FaaS) and Serverless architecture patterns which means that those functions are deployable without the requirement to configure and manage server infrastructure which makes the deployment process easier [8].

Edge functions have next benefits [8]:

1. Increased efficiency of pricing because of the "pay as you go" pricing model. This pricing model is much more flexible compared to classic dedicated servers because the customer pays only for the resources that are used and pays nothing if the function is not used for a while.

2. Reduced latency for target users due to execution of the function in the data centers which is geographically closest to the requester.

3. Auto-scaling by design. Edge functions are designed in a way to make automatic scaling on demand. At the same time, scaling strategies are flexible because of the provided configuration API. For example, it's possible to always keep 1 or more instances running to avoid cold-start lag.

4. Edge functions are suitable for different use cases. They might be used for the classic API endpoint creation, dynamic content generation, webhooks, etc.

5. Edge functions provide an advanced security level thanks to its isolated nature. Independence from a centralized server environment radically decreases the chance of side effects from malicious requests for the whole system. Also, standalone nature helps to do maintenance more easily disabling only particular features of the application for the maintenance period.

6. For the specific use cases edge functions support Web Assembly, which makes them scalable and increases readiness for really unusual feature requests by businesses.

Supabase provides a big free quota and easy-to-use infrastructure for quick and reliable deployment of the edge functions based on the Deno platform which is a great fit for the development of the small startup or MVP applications. Also because of the usage of the Deno under the hood, it's possible to use a wide range of regular NPM packages since Deno is the JavaScript/TypeScript platform. Deno was designed with a security-first and TypeScript-first vision which gives some advantages against Node.JS such as built-in TypeScript support without the need to transpile, better environment variables support, etc.

To use OpenAI Completions API for different use cases in the scope of the application, the edge function was designed abstracted from concrete features. This approach makes this function reusable and predictable. The Completions API settings were predefined in the const until they needed customization, but because of the clear API of the function, it remains ready for further enhancements on demand (Listing 1).

The authorization token is needed to make the OpenAI Completions API work. Deno's built-in environment variables tool was used since it's considered an anti-pattern to hard-code tokens in the codebase.

This helps to ensure that the token is not accidentally exposed to the codebase [6, 8].

```
const OPEN_AI_API_KEY = Deno.env.get("OPENAI_API_KEY");

const completionsApiSettings: CompletionsApiSettings = {
  model: "gpt-4",
  temperature: 0.7,
  top_p: 1,
  frequency_penalty: 0,
  presence_penalty: 0,
  max_tokens: 1000,
  stream: true,
  n: 1,
};

serve(async (req) => {
  const { messages } = await req.json();

  try {
    const res = await fetch("https://api.openai.com/v1/chat/completions", {
      method: "POST",
      headers: {
        Authorization: `Bearer ${OPEN_AI_API_KEY}`,
        "Content-Type": "application/json",
      },
      body: JSON.stringify({
        ...completionsApiSettings,
        messages: messages,
      }),
    });

    const processedCompletionsApiResponse = processCompletionsApiResponse(res);

    return processedCompletionsApiResponse;
  } catch (error) {
    return new Response(JSON.stringify({ data: null, error: error }));
  }
});
```

Listing 1. Code fragment with OpenAI Completions API usage in Supabase Edge Function

Conclusion

Most information systems require well-configured, secured, and scalable infrastructure. The classic approach to managing infrastructure gradually evolved into cloud-based infrastructure.

Keeping infrastructure in-house is expensive and requires time investment into the establishment and further maintenance. This article presented a chart of computing power growth over the years. This chart represents the amount of floating-point operations per second (GFLOPS)/year which is a standard way to estimate computing resources.

The provided chart proves that cloud technologies have become cheaper and more accessible to businesses.

An analytical review of existing cloud solutions presented in this article provides:

- list of the most popular cloud platform providers, compare;
- comparison of the low-level and high-level cloud solutions;

● compares the pros and cons of two cloud platforms from a single vendor (Google);

● highlights the importance of the project solutions decision-making based on project context and details.

In the main part was provided an example of the cloud platform selection for the implementation of the information system to automate daily tasks in the recruitment workflow based on defined requirements and features.

During the selection of the best-matching cloud solution selection, the level of abstraction of the cloud platform was determined.

A brief comparison of the most popular high-level cloud platforms such as Google Firebase and Supabase was made to make development efficient.

After a high-level comparison of the most popular Backend as a Service providers, pros and cons were analyzed to ensure that the open-source Supabase solution covers the project's requirements.

Also in the scope of the Supabase cons analysis, the fallback strategies possibility was estimated to ensure that Supabase doesn't make our application fully dependent and locked into the vendor.

In order to implement the feature of the feedback generation for candidates after interviews Supabase Edge Functions were utilized. Additionally, before implementation benefits of usage of the Supabase Edge Functions were highlighted. The fragment of the code with usage of the OpenAI Completions API is presented.

In conclusion, fast-growing computing performance leads to better pricing. Better pricing made a great environment for the growth of the new concept – cloud technologies and serverless architectures. Cloud providers in the last decades enhanced their infrastructure and nowadays customers who choose cloud solutions from top cloud providers might be sure of usage under the hood industry-standard infrastructure covered by best practices of top companies, high-security level, and automatic scalability by design. An ability to use cloud solutions is a big benefit for startups and small projects that can't hire experienced DevOps to manage it in-house.

References

1. Wong T. *Introduction to classical and quantum computing* / Thomas Wong. – [S. l.] : Rooted Grove, 2022.
2. Marinescu D. C. *Cloud computing: theory and practice* / Dan C. Marinescu. – [S. l.] : Elsevier Science & Technology, 2022.
3. Hunter T. *Google cloud platform for developers: build highly scalable cloud solutions with the power of google cloud platform* / Ted Hunter, Steven Porter. – [S. l.] : Packt Publishing, 2018. – 506 p.
4. Singh H. *Serverless Web Applications with React and Firebase: develop real-time applications for web and mobile platforms* / Harmeet Singh, Mayur Tanna. – [S. l.] : Packt Publishing, 2018. – 284 p.
5. Khan O. M. A. *Enterprise Application Architecture with .NET Core: An architectural journey into the Microsoft .NET open source platform* / Ovais Mehboob Ahmed Khan, Ganesan Senthilvel, Habib Ahmed Qureshi. – [S. l.] : Packt Publishing, 2017. – 564 p.
6. Sarrion E. *Exploring the Power of ChatGPT* [Electronic resource] / Eric Sarrion. – Berkeley, CA : Apress, 2023. – Mode of access: https://doi.org/10.1007/978-1-4842-9529-8 (date of access: 25.01.2024). – Title from screen.
7. Kolesnikov O. *The objective need to implement an information system to automate daily tasks in recruitment workflow* [Electronic resource] / O. Kolesnikov, G. Golovko // Системи управління, навігації та зв'язку. Збірник наукових праць. – 2023. – Vol. 3, no. 73. – P. 106–110. – Mode of access: https://doi.org/10.26906/sunz.2023.3.106 (date of access: 26.01.2024). – Title from screen.
8. Supabase documentation [Electronic resource] // Supabase Docs. – Mode of access: https://supabase.com/docs (date of access: 22.01.2024). – Title from screen.

Received (Надійшла) 22.11.2023
Accepted for publication (Прийнята до друку) 24.01.2024

Використання хмарних технологій та безсерверної архітектури для ефективної веб-розробки: приклад із реального світу

О. Колеснiков, Г. Головко, В. Ястреба, Є. Пятінцев

Анотація. У цій статті розглянуто сучасні хмарні рішення та безсерверну архітектуру на прикладі використання Backend as a Service (BaaS) та Function as a Service (FaaS) архітектури. Проведено паралель між консистентно зростаючою обчислювальною потужністю та зростаючою популярністю і доступністю хмарних технологій. Здійснено аналітичний огляд в рамках якого наведено список найпопулярніших хмарних провайдерів від провідних корпорацій, порівняно високорівневі та низькорівневі хмарні технології. Представлено переваги та недоліки Google Cloud Platform (GCP) та Google Firebase, де GCP – низькорівневий хмарний провайдер, а Firebase – високорівневий. Підкреслено важливість розуміння контексту та особливостей проєкту при обранні хмарних проєктних рішень задля збереження балансу між гнучкістю та ефективністю розробки. Дослідження знайомить з практичним застосуванням хмарних технологій для розробки сучасного вебдодатку, а саме Supabase. Стаття переконливо доводить доцільність використання Supabase для рорзобки інформаційної системи для оптимізації сучасного процесу рекрутменту персоналу з зазначенням конкретних переваг. Представлено приклад використання Supabase Edge Functions для генерації зворотного зв`язку з використанням OpenAI Completions API та програмної платформи Deno. У статті переконливо доведено, що використання хмарних технологій є сучасною стратегією побудови гнучких, ефективних і масштабованих інформаційних систем. Узагальнено переваги використання інфраструктури, наданої світовими лідерами галузі.

Ключові слова: системи, інформаційні системи, інформаційні технології, хмарні провайдери, безсерверна архітектура, GCP, Firebase, Supabase, ефективність, штучний інтелект.