https://doi.org/10.31891/csit-2024-3-8
UDC 621.317

Yuriy SKORIN, Iryna ZOLOTARYOVA, Yuriy LYSTOPAD
Simon Kuznets Kharkiv National University of Economics

THE MANAGEMENT OF SCALABILITY IN CLOUD-BASED APPLICATIONS

The following is an abstract of the article. The article presents an analysis of the challenges associated with monitoring and managing the scalability of a cloud application. To this end, a module for monitoring and managing the scalability of a cloud application has been developed as part of this study. The development process included the introduction of automatic scaling, monitoring using Prometheus and Grafana, which allows for a high level of availability and resource efficiency. The study comprised a series of phases, including requirements analysis, system design, development, testing, and evaluation. Consequently, the system's performance, stability and capacity to scale in response to fluctuating workloads were enhanced. The module exhibits a high degree of adaptability to changes in system requirements and load, which is a crucial attribute for the dynamic development of business applications. This solution assists in optimising the allocation of resources and reducing infrastructure costs. The project has been found to fully meet the set goals and objectives, as well as the requirements for effective resource management of the Amazon Web Services cloud platform using Terraform, Prometheus and Grafana. The practical value of the developed module is evidenced by a significant improvement in resource efficiency, service stability and cost optimisation. The module design has been subjected to rigorous testing and has been successfully implemented in a test environment, thereby demonstrating the sustainability and efficiency of the developed solution. The experience gained in the implementation and operation of this solution may prove useful for further expansion and optimisation of cloud solutions in other projects and companies specialising in the provision of cloud solutions. The findings of this study were validated in a test environment at an IT company with a specialisation in cloud technologies. The objective was to ascertain the functionality and efficiency of the developed module in a real-world context of cloud infrastructure operation. The testing process entailed the configuration of the module on pre-existing cloud infrastructure systems, its integration with Prometheus and Grafana for monitoring purposes, and the execution of a series of stress tests designed to assess the module's scalability. As a result of this testing, a number of critical points were identified that required further optimisation. The results of the study and the issues identified during the project testing have enabled the identification of several areas for further improvement and development of the system. First and foremost, the optimisation of automatic scaling algorithms represents a crucial avenue for improvement. The development of these algorithms should be oriented towards utilising historical monitoring data to anticipate potential shifts in system load. Another pivotal area for enhancement is the precision of monitoring systems. The integration of supplementary tools and the expansion of existing monitoring systems' functionality will facilitate the acquisition of more comprehensive insights into the system's condition. This, in turn, will facilitate the expedient identification and eradication of potential issues.
Keywords: scalability, cloud application, Prometheus, Grafana, Terraform, stress test, monitoring.

Юрій СКОРІН, Ірина ЗОЛОТАРЬОВА, Юрій ЛИСТОПАД
Харківський національний економічний університет імені Семена Кузнеця

УПРАВЛІННЯ МАСШТАБУВАННЯМ ХМАРНИХ ДОДАТКІВ

У статті проаналізовано проблеми моніторингу та управління масштабованістю хмарного застосунку, у рамках даного дослідження був створений модуль моніторингу та управління масштабованістю хмарного застосунку. Розробка включала в себе впровадження автоматичного масштабування, моніторингу за допомогою Prometheus та Grafana, що дозволяє забезпечити високий рівень доступності та ефективності використання ресурсів. Дослідження передбачало проведення аналізу вимог, проєктування системи, розроблення, тестування та оцінку результатів. Завдяки цьому, було досягнуто зростання продуктивності і стабільності системи, а також підвищено здатність до масштабування відповідно до змінюваного навантаження. Модуль демонструє високу адаптивність до змін у вимогах та навантаженні системи, що важливо для бізнес-застосунків, які динамічно розвиваються. Таке рішення сприяє оптимальному розподілу ресурсів і мінімізації витрат на інфраструктуру. Проект повністю відповідає встановленим цілям та завданням, вимогам ефективного управління ресурсами хмарної платформи Amazon Web Services з використанням Terraform, Prometheus та Grafana. Практична цінність розробленого модуля полягає у значному покращенні ефективності використання ресурсів, забезпеченні стабільності сервісів та оптимізації витрат. Проект модулю був апробований та успішно імплементований у тестовому середовищі, де була продемонстрована стійкість та ефективність розробленого рішення. Досвід із впровадження та експлуатації може бути корисним для подальшого розширення та оптимізації хмарних рішень в інших проектах та компаніях, що спеціалізується на наданні хмарних рішень. Результати цього дослідження були апробовані на тестовому середовищі в IT-підприємстві, що спеціалізується на хмарних технологіях. Тестування проходило з метою перевірки функціональності та ефективності розробленого модуля у реальних умовах експлуатації хмарної інфраструктури.
Ключові слова: масштабованість, хмарний застосунок, Prometheus, Grafana, Terraform, стрес-тест, моніторинг.

Introduction

Today's world is characterised by the rapid development of information technology, which is having a significant impact on all aspects of human activity. The automation of business processes is particularly important, as it allows companies to improve their efficiency, optimise costs and increase the speed of response to market changes. The use of cloud technologies is becoming an answer to the need for flexibility and scalability of IT infrastructure, allowing resources to be dynamically scaled up and down according to business needs.

Growing volumes of data and the need to process it require monitoring systems to be highly efficient and able to scale quickly. However, traditional approaches often prove too inflexible or costly to deploy, prompting the search for new solutions. In this context, cloud technologies offer the opportunity to efficiently deploy and scale monitoring and resource management systems, enabling high availability and reliability of IT services. The need to develop systems capable of automatically adapting to changing workloads and optimising resource utilisation is being driven by increasing demands for efficiency in processing large amounts of data and the need to reduce the cost of maintaining IT infrastructure.

The relevance of the study focuses on the critical need to improve and optimise cloud resource scalability management systems. With the rapid development of technology and the increase in data volumes, effective management of cloud infrastructure resources is becoming a critical factor in ensuring high performance and availability of online services. Cloud technologies offer great opportunities for scalability and elasticity, but also require granular control and adaptive management to ensure an optimal performance/cost ratio.

Today's business requirements change frequently, requiring scalability monitoring and management systems to adapt quickly and automatically to these changes. Developing a module that can analyse current performance and resource usage and automatically adjust system scaling based on this analysis meets these requirements. This not only ensures business continuity, but also contributes to a significant reduction in IT infrastructure maintenance costs.

Thus, the relevance of this study lies in the need to develop effective solutions for dynamically managing the scalability of cloud resources that can ensure high availability and performance of services at optimal cost.

The aim of this paper is to implement a software module for monitoring and managing the scalability of a cloud application based on the AWS (Amazon Web Services) platform and managed by Terraform. The main idea is to create a system capable of automatically analysing current resource usage and adaptively adjusting service scaling based on the data obtained. This will help to increase system efficiency, reduce overall maintenance costs and increase end-user satisfaction by maintaining optimal application performance.

Related works

Publications [1-2] show that ensuring high availability and efficiency of cloud applications requires advanced scaling solutions. Scaling is defined as a strategic element to ensure continuous operation of applications even as the load and volume of incoming data increases. AWS was chosen as the cloud infrastructure for the application deployment to provide a wide range of features and high reliability. AWS has an extensive network of data centres to ensure maximum application availability for users in different geographical regions, and AWS offers a number of tools for automation and infrastructure optimisation, which is a key factor for successful scaling. Terraform was chosen as the tool to automate infrastructure tasks because of its ability to describe and manage infrastructure configuration in code.

The author of the publication, Evgeny Brykman [3], proves that Terraform is a real star in the DevOps world. It is a technology that enables the configuration, deployment and management of cloud infrastructure. "Infrastructure as a Code (IaC) allows you to make the most of virtual platforms from tech giants such as AWS, Google Cloud, Azure and others. The authors provide simple, concise examples of the code used in Terraform to deploy and manage infrastructure. Experienced DevOps engineers, and even beginners, will quickly move from the basics of Terraform to working with a full stack capable of supporting large volumes of traffic and ensuring stable conditions for a large team of specialists.

In the article [4], author Jack Dwyer details how Terraform can be used to automate the deployment and management of cloud infrastructure. He analyses how Terraform simplifies resource management through descriptive configuration files, enabling rapid scaling and modification of infrastructure without the need for manual intervention.

Aalok Trivedi's paper [5] shows the potential of AWS in the context of ensuring scalability and high availability of cloud solutions. The article provides examples of using various AWS services such as Elastic Load Balancing, Auto Scaling and Amazon S3 to develop resilient and efficient systems.

The paper [6] analyses methods and tools for monitoring and analysing cloud resources, describing the process of using integrated monitoring tools such as AWS CloudWatch and Prometheus to track the operation of cloud resources and identify potential problems.

The paper [7] defines and outlines the main benefits of cloud scaling, discusses the possibilities of scaling to a large number of subscribers, and analyses the use of load balancing, virtual servers and virtual storage.

The paper [8] discusses two main scaling strategies - horizontal (dispersal) and vertical (infrastructure expansion).

Paper [9] defines the term "cloud", identifies which services are cloud-based, considers cloud deployment models and analyses the benefits of cloud technologies.

Articles [10-11] consider the issue of cloud testing of mobile applications, analyse the advantages and disadvantages of cloud testing of mobile applications, identify the types of testing that can be performed in the cloud and which applications should be tested in the cloud.

In the publication [12], the author, Ilya Reznikov, provides an overview of the history of databases, defines a cloud database, analyses database deployment models, and various cloud database mechanisms.

The author of the publication [13], Alexandra Balykina, considers such issues as types of cloud services, basic models of cloud services, advantages of "clouds, security, selection of a cloud platform, and implementation of clouds in business.

The publication [15] addresses such issues as what is the cloud, what are IaaS and XaaS, how does PaaS differ from IaaS, what tasks does SaaS solve, how to choose a service model, etc.

Based on the literature review, it can be concluded that the integration of tools such as Terraform and AWS services creates a powerful platform for managing cloud resources. Modern solutions provide not only high level of process automation, but also in-depth monitoring and analysis of system health, which is key to maintaining the scalability and resilience of business applications in the cloud. This approach allows companies to quickly adapt to changing market needs and user demands.

The management of scalability in cloud-based applications

In today's cloud world, the importance of effective resource management and monitoring cannot be overstated. There are numerous software solutions designed to optimise the use of cloud resources and ensure their availability and scalability. A review of existing software products provides an opportunity to understand the level of technical sophistication in this area and to identify best practices that can be used in the development of new solutions. This section analyses several leading cloud monitoring and management software products, such as AWS CloudWatch, Datadog and Microsoft Azure Monitor. The aim of the analysis is to compare their key features, including vendor, product versions, operating system, key functionality, user interface, cost of use and availability of free versions or plans. This approach will allow you to gain a detailed overview of the advantages and disadvantages of each of the solutions under consideration, which in turn will help you to formulate a sound strategy for developing your own software product.

In order to achieve this goal, the following objectives have been set

- Analysis of existing solutions, modern methods and tools for monitoring and managing scalability in cloud environments, identify requirements and standards for implementing such systems;
- System design and development of the architecture of the module, identify key components and their interactions, and select technologies to implement the required functionality;
- Module implementation, including metrics collection, data analysis and scaling functions, and ensure integration with AWS and Terraform for resource management;
- Test and tune to verify functionality and reliability, tune module operation to ensure efficiency;
- Documentation for the project, including instructions for using and maintaining the module;
- Implementation and demonstration of the module in a test environment to demonstrate its capabilities and benefits.

This approach will enable a deeper understanding and use of cloud technologies for scaling and monitoring, and contribute to the optimisation of information system resources.

The subject of this study is the scaling and monitoring processes in cloud information systems. These processes form an important part of cloud management, including tracking system operating parameters and automatically adjusting the amount of resources according to current load and user needs.

Scalability helps to adapt the system to changing usage conditions, thereby improving performance and user satisfaction.

Monitoring provides the collection of key metrics that allow you to identify and predict trends, analyse the state of the system and is an integral part of the scaling process.

The development of methods and tools to optimise these processes within AWS cloud systems using Terraform tools formed the basis of the practical part of this study. As a result, the monitoring and scaling system enables real-time response to load changes and optimisation of resource utilisation, leading to an increase in the overall performance and availability of cloud resources.

The subject of research in this thesis is the methods of automatic scaling and monitoring of resources in cloud environments. In particular, this includes the study and application of modern technological solutions that allow optimising the use of computing resources through automated scaling based on changing usage conditions and system requirements.

Emphasis will be placed on methods for dynamic analysis of system performance metrics, including the collection, processing and interpretation of resource usage data to make decisions about the need for scaling.

These processes are supported by the use of monitoring tools such as Prometheus and Grafana, which allow you to visualise key performance indicators and identify potential problems in the cloud infrastructure.

Technologies and platforms such as AWS and Terraform are also being explored to provide tools for the deployment and management of cloud resources, helping to put scaling algorithms into practice. This will allow the techniques developed to be integrated directly into cloud environments, automating resource management processes and improving overall system efficiency.

This study takes an integrated approach to implementing a cloud infrastructure using design methodologies such as Terraform and AWS. Terraform is used for the declarative description of the infrastructure, allowing the required resources and their configurations to be defined in code. This helps to automate the deployment and management of the infrastructure, ensuring accuracy, repeatability and ease of scaling. AWS is used as the primary platform for deploying cloud infrastructure and offers a wide range of services and solutions for efficient cloud hosting and resource management. AWS features such as Elastic Compute Cloud (EC2) and Auto

Scaling provide the project with the necessary tools to dynamically scale resources according to the current load and ensure high availability and reliability of the system.

The use of these methods not only allows for effective management of cloud infrastructure resources, but also ensures a high level of security, stability and scalability of the software, which are critical aspects when working with modern cloud solutions.

The scientific significance of this study is the development and research of effective methods of resource management in cloud infrastructures, which contributes to the development of the field of cloud technologies.

The practical significance of the work is expressed in the possibility of implementing the developed solution to optimise the use of resources in real IT infrastructures of companies. This leads to a significant increase in efficiency, cost reduction and an increase in the overall availability and reliability of information systems.

The implementation of the developed software module in a test phase at an IT company allowed us to verify its effectiveness and identify potential areas for further improvement. This confirms the feasibility of using the development in the field of cloud resource management and provides a basis for its further implementation on an industrial scale.

In the contemporary digital landscape, the significance of efficacious resource administration and observation cannot be overstated. A plethora of software solutions have been developed with the objective of optimising the utilisation of cloud resources, ensuring their availability and scalability.

A review of existing software products allows for an understanding of the level of technical sophistication in this area to be gained, as well as the identification of best practices that can be used in the development of new solutions. This section presents an analysis of several leading cloud monitoring and management software products, including AWS CloudWatch, Datadog, and Microsoft Azure Monitor. The objective of this analysis is to provide a comprehensive comparison of the key features of the selected software products, including vendor, product versions, operating system, primary functionality, user interface, cost of use, and availability of free versions or plans. This approach enables a detailed examination of the advantages and disadvantages of each solution, which can inform the development of a strategic plan for the creation of a software product.

The following is a description of the various forms that the main functionality takes on the screen:

AWS CloudWatch - a dashboard with graphs and indicators for the visualization of resource metrics, as well as the setting up of alarms to notify the user of critical changes in resource usage.

Datadog offers detailed and customizable dashboards for a variety of metrics and logs, as well as integration with numerous cloud services for data collection and application status tracking.

Microsoft Azure Monito - a dashboard that provides visualization of data on the utilization of cloud services, analytical tools for comprehensive data analysis and report generation.

A review of existing software products reveals that contemporary solutions offer flexibility in monitoring and analytics configuration, facilitate integration with diverse cloud platforms, and support the processing of substantial data volumes.

In the development of new software for monitoring and managing cloud resources, it is essential to consider the experience of leading development firms. This entails the creation of intuitive, scalable and efficient solutions that meet the high demands of modern users.

In order to establish a position within the information space of the IT market, it is essential to provide a clear definition of the unique characteristics of the software in question (Table 1).

Table 1

Analysis of existing solutions

Characteristics	AWS CloudWatch	Datadog	Microsoft Azure Monitor
Development company	AmazonWebServices	Datadog, Inc.	Microsoft
Product versions	Constantly updated	Constantly updated	Constantly updated
Operating system	Cloud platform	Cloud platform	Cloud platform
Main functionality	Resource monitoring, alarms	Monitoring, analytics, alarms	Resource monitoring, analytics
User interface	Web-based	Web-based	Web-based
User assistance	Documentation, online support	Documentation, online support	Documentation, online support
Price of use	Depends on the volume of use	Depends on the plan and scope	Depends on the volume of use
Availability of a free version / plan	Basic monitoring is free of charge	Trial period	Basic monitoring is free of charge

A review of existing software products reveals that contemporary solutions offer flexibility in establishing monitoring and analytics, are capable of integrating with diverse cloud platforms, and are equipped to handle substantial volumes of data.

In the development of new software for monitoring and managing cloud resources, it is essential to consider the experience of leading development firms, with a focus on the creation of intuitive, scalable and efficient solutions that meet the high demands of modern users.

In order to establish a position for software within the information space of the IT market, it is essential to provide a clear definition of the unique aspects and the values that it offers to the target audience. This process of definition enables the monitoring and scalability management module for cloud applications, which is currently under development and is based on AWS and Terraform, to identify its niche within the market:

At this stage of cloud technology development, there is an increasing demand for automation and resource optimisation. The module being created addresses these requirements, offering efficient scaling and monitoring of resources with high accuracy and minimal user intervention.

The module is designed to automatically scale and monitor cloud applications in real time, providing up-to-date data on the status of resources to IT departments and system administrators. It allows for intervention only when necessary, thereby increasing the efficiency of infrastructure management and reducing the likelihood of errors.

The necessity for an integrated approach to monitoring and management is becoming increasingly apparent in the context of modern cloud platforms. Inefficient resource management can result in unnecessary costs and reduced productivity. The module under development addresses this issue by providing tools for automating and optimising cloud operations.

These characteristics provide a robust basis for the product's successful positioning in the IT market, rendering it an appealing option for businesses aiming to effectively manage their cloud resources.

In the context of cloud resource management, inefficient scaling and monitoring practices result in unnecessary costs and reduced application performance.

This has a direct impact on IT departments and system administrators, as well as end users of business applications. Insufficiently configured monitoring and auto-scaling systems can result in unstable systems and a suboptimal user experience.

The principal consequences are increased operational costs and a diminished capacity to promptly adapt to evolving market requirements.

The resolution of this issue through the development of an effective monitoring and scaling module can yield substantial business advantages, including diminished expenses and enhanced reliability of the applications in question.

Table 2

Problem solving

Element	Details
Problem.	Inefficient scaling and monitoring of cloud resources, which leads to cost overruns and reduced system performance.
It touches on	IT departments, system administrators, project managers, and end users of business applications.
Its consequences are	Increased costs for cloud services, reduced responsiveness to changing market needs, and possible user dissatisfaction due to lower service quality.
Successful solution	Reduce operational costs, increase the efficiency of cloud resources, ensure high availability and performance of applications, and improve user satisfaction.

The task is to clearly define what unique position the software product under development will occupy in the IT solutions market.

This product is designed to provide companies with a powerful tool for optimising and automating cloud management.

It's important to note that proper positioning helps potential users understand why they should choose this product over numerous alternatives.

Below is a table detailing how this product meets the needs of the market, its key advantages and differences from competitive solutions (Table 3).

Table 3

Position definition

Element	Details
For	IT departments of large and medium-sized enterprises that actively use cloud technologies.
Which is	They are looking for solutions to optimise costs and improve the efficiency of cloud management.
Product name	"The Scalability Monitoring and Management Module is a cloud infrastructure management tool.
Which is	Allows you to automate scaling and monitoring of resources, provides detailed analytics of resource usage.
Unlike the	Traditional solutions such as manual control or less flexible automated systems.
This product	Includes integration with AWS and Terraform, which provides broad compatibility and deep integration with cloud services, automating processes and significantly reducing the need for manual intervention.

In order to implement the module for monitoring and managing the scalability of a cloud application based on AWS and Terraform, a number of main technologies and tools are employed, as outlined below in table 4.

Table 4

Choosing the means of project implementation

Name of the product	Purpose	Alternative options considered	Rationale for the choice
Terraform	Automate infrastructure settings	Ansible, Chef	Terraform provides a clearer and more modular syntax for describing infrastructure as code, which is ideal for large cloud systems.
AWS EC2, AWS Auto Scaling	A framework for creating and scaling virtual machines	Azure Virtual Machines, Google Compute Engine	AWS provides deep integration with other AWS services, making it easy to deploy and scale.
Bash	Scripting language for automating commands and processes	Python, Ruby	Bash has a high execution speed and is widely used for Linux scripting, which provides direct integration with the system shell.
Prometheus and Grafana	System monitoring and metrics visualisation	Zabbix, Nagios	We chose it because of its high flexibility in data collection and visualisation, as well as support for a large number of data sources.
Docker	Containerisation for application isolation and easy deployment	Kubernetes (for orchestration), LXC	Docker simplifies dependency and deployment management, and provides fast creation and destruction of environments.

The table illustrates the rationale behind the selection of each technology, taking into account its suitability for the project's objectives, the alternatives considered and the underlying decision-making process. This demonstrates a balanced approach to the technology stack selection.

During the reporting period, the Group successfully conducted a trial of the software designed to monitor and manage the scalability of its cloud infrastructure.

The software was subjected to rigorous testing in a meticulously crafted test environment that emulates the operational conditions of a real-world system on the AWS platform.

The principal elements of the test environment were Amazon EC2 instances, configured to facilitate automatic scaling and integration with Prometheus and Grafana monitoring systems for the visualisation of metrics.

The testing phase encompassed both manual and automated scenarios.

Manual testing was conducted to evaluate the Grafana user interface and its interaction with Terraform configuration files. Automated testing was performed using scripts that simulated various loads on the system to assess the efficacy of the automatic scaling logic.

During the testing phase, particular attention was paid to the re-integration process following amendments to the code, thereby enabling the early detection of errors during the development stage.

This approach guaranteed the high reliability and stability of the software.

The following data were employed in the construction of the testing schedule:

The importance factor is as follows:

V_h = 3 (coefficient for a high level of importance);

V_m = 2 (coefficient for the average level of importance);

V_l = 1 (coefficient for low importance).

Furthermore, a test of the recovery of the defects that were fixed was conducted to ascertain that previous errors were not repeated and that the new changes did not result in a regression of functionality.

All testing procedures were documented in the form of test cases and results, which were subsequently incorporated into reports (Table 5).

Table 5

Testing Schedule and Procedures for Error Detection

X - test case number	Y - result * importance coefficient
1	1*3
2	1*2
3	0
4	0
5	0
6	0

During the testing phase, particular emphasis was placed on the re-integration process following alterations to the code, thereby facilitating the early detection of errors during the development stage. This approach guaranteed the high reliability and stability of the software product.

Based on the findings of the software tests and the subsequent analysis of the results of defect detection and correction, the following recommendations are put forth with the intention of enhancing the quality of the product and optimising the development process. In light of the identification of several critical defects, it is recommended that the number of tests for the critical path be increased and that additional test scenarios be added to ensure comprehensive coverage of all potential code paths.

The implementation of a more expansive array of automated tests will facilitate the expedient identification of errors at the nascent stages of development, thereby reducing the financial burden associated with their rectification. It is recommended that the project documentation be updated and optimised, particularly with regard to the description of test scenarios and defects. This will facilitate a more comprehensive understanding of the issues and facilitate more effective resolution. The incorporation of static analysis tools can facilitate the identification of errors prior to code execution, thereby enhancing the overall quality of the code. It is recommended that regular training and seminars be organised for developers and testers with the aim of enhancing their skills and familiarising them with the latest technologies and best practices in the industry. It is imperative that more rigorous security guidelines are implemented and that the software in question is shown to comply with the relevant national and international security standards. It is recommended that communication processes between developers, testers and other stakeholders be improved in order to facilitate more efficient interaction and coordination.

It is anticipated that these recommendations will not only enhance the quality of the final product, but also facilitate overall development productivity and ensure a higher level of user satisfaction.

Conclusions

As part of this study, a module for the monitoring and management of the scalability of a cloud application was developed. The development process included the implementation of automatic scaling, monitoring using Prometheus and Grafana, which ensures a high level of availability and resource efficiency. The study encompassed a comprehensive range of activities, including requirements analysis, system design, development, testing, and evaluation. Consequently, the system's performance, stability and capacity to scale in response to fluctuating workloads were enhanced. The module exhibits a high degree of adaptability to changes in system requirements and load, which is a crucial attribute for the dynamic development of business applications. The solution assists in optimising the allocation of resources and reducing infrastructure costs.

The project has been successfully completed in accordance with the initial objectives that were set at the outset of the research. The developed scalability monitoring and management module is fully compliant with the requirements for effective resource management of the AWS cloud platform using Terraform, Prometheus, and Grafana.

In particular, the following key tasks were implemented:

the process of scaling the infrastructure was automated in accordance with the prevailing system load, thus ensuring optimal performance at minimal cost;

a monitoring system was implemented to ensure the collection, analysis and visualisation of system metrics in real time, thereby facilitating the rapid identification and elimination of potential issues.

These measures have led to a notable enhancement in the quality and reliability of the infrastructure, ensuring its stability and availability while guaranteeing the necessary level of data security.

The practical value of the developed module for monitoring and managing scalability for a cloud application is evidenced by significant improvements in resource efficiency, service stability, and cost optimisation.

The automation of scaling processes enables the accurate prediction of resource requirements and their configuration in accordance with these predictions, thereby facilitating the saving of resources and money.

The monitoring system facilitates the rapid identification and rectification of operational issues, thereby enhancing the overall availability and reliability of the cloud application.

The integration of testing, continuous monitoring and scaling tools facilitates the streamlining of development and support processes, thereby enhancing product quality and simplifying project support and development processes.

The deployment of sophisticated security protocols and the continuous observation of the system's status safeguard data and infrastructure from potential threats.

The results of the project, which focused on the development of a monitoring and scalability management module for a cloud application based on AWS and Terraform, were subjected to rigorous testing and successfully implemented in a test environment. This demonstrated the stability and efficiency of the solution that had been developed. The experience gained in the implementation and operation of this solution can be leveraged for further expansion and optimisation of cloud solutions in other projects and companies specialising in the provision of cloud solutions.

The implementation process comprised a number of distinct stages.

The initial phase of the project entailed integrating the monitoring and scalability management module with the company's existing infrastructure. In view of the multiplicity of technology solutions already in use within the company, it was imperative to adapt the module in order for it to function effectively with a variety of server and cloud service configurations. Integration efforts were undertaken with the objective of ensuring a seamless integration process that would not disrupt the operation of existing systems.

The second stage comprised comprehensive testing of the module in a controlled setting to ascertain potential defects and inefficiencies. The testing phase facilitated the identification of several pivotal issues that could potentially impact the efficacy of scaling and monitoring the system. The identified issues were then categorised and the requisite modifications were implemented expeditiously.

The findings of this study were subjected to evaluation within a test environment at an IT company with a specialisation in cloud technologies. The objective of the testing phase was to ascertain the functionality and efficiency of the developed module in a real-world context, specifically in terms of its compatibility with existing cloud infrastructure. The testing process comprised the installation of the module on existing cloud infrastructure systems, its integration with Prometheus and Grafana for monitoring purposes, and the execution of a series of stress tests designed to assess scalability. As a result of this testing, a number of critical points were identified that required further optimisation.

This practical experience facilitated an understanding of the project's actual operational capabilities and utility, and validated its potential for further advancement and implementation in actual productive environments.

In light of the findings of the study and the issues that emerged during the course of the project's testing phase, a number of avenues for further enhancement and the evolution of the system can be discerned.

First and foremost, it is essential to optimise the automatic scaling algorithms. The objective of developing these algorithms should be to utilise historical monitoring data in order to predict potential changes in system load. Such an approach will not only guarantee a high level of system readiness for changes, but will also optimise the utilisation of resources.

Another crucial area for improvement is the accuracy of monitoring systems. The integration of supplementary tools and the enhancement of existing monitoring systems will facilitate the acquisition of more comprehensive data regarding the system's condition. This will facilitate the expedient identification and eradication of potential issues. Furthermore, it is essential to enhance the system's integration capabilities. The incorporation of support for novel cloud platforms and technologies will enhance the scope of the developed solution and facilitate its compatibility with diverse infrastructures. Furthermore, it is imperative to accord special attention to the issue of security. This entails the implementation of supplementary data.

References

1. Didenko O.K. Application of the method of controlled development behaviour for automation of web application testing // O. K. Didenko, D. Y. Holubnychyi // Proceedings of the International Scientific and Practical Conference of Young Scientists, Postgraduates and Students ‘Information Technologies in the Modern World: Research of Young Scientists’ 22 - 23 February 2024 - Kharkiv, 2024.

2. Nystopad Yurii. Development of a module for monitoring and managing the scalability of a cloud application based on AWS and Terraform // Yurii Lystopad, Yurii Skorin // Proceedings of the International Scientific and Practical Conference of Young Scientists, Postgraduate Students and Students ‘Information Technologies in the Modern World: Research of Young Scientists’ 22 - 23 February 2024 - Kharkiv, 2024.

3. Yevhen Brykman. Terraform: infrastructure at the code level, 2024. [Access mode: https://bambooks.com.ua/ua/p1809769909-kniga-terraform-infrastruktura.html).

4. Jack Dwyer. Automating Cloud Infrastructure with Terraform CI CD: A Step-by-Step Guide, 2024. [Access mode: https://zeet.co/blog/terraform-ci-cd).

5. Aalok Trivedi. Leveraging high availability by creating an AWS Auto scaling group & application load balancer, 2023. [Access mode: https://medium.com/@aaloktrivedi/leveraging-high-availability-by-creating-an-aws-auto-scaling-group-application-load-balancer-2ea1f31a746.

6. Real-time Monitoring and Analysis of Edge and Cloud Resources, 2023. [Access mode: https://dl.acm.org/doi/10.1145/3589010.3594892.

7. Scalability as a software requirement, meaning and definition. [Electronic resource]. - Access mode: (https://uk.itpedia.nl/2021/07/20/schaalbaarheid-als-software-requirement-betekenis-en-definitie.

8. Cloud scalability: horizontal and vertical scaling of IT infrastructures. [Electronic resource]. - Access mode: https://cases.media/en/article/masshtabovanist-khmari-gorizontalne-ta-vertikalne-masshtabuvannya-it-infrastruktur.

9. What are cloud technologies? Advantages and disadvantages of cloud services. [Electronic resource]. - Access mode: https://edin.ua/shho-take-xmarni-texnologi%D1%97-i-navishho-voni-potribni.

10. Cloud testing of mobile applications. [Electronic resource]. - Access mode: https://training.qatestlab.com/blog/technical-articles/cloud-testing-mobile-application.

11. Cloud storage is an integral part of the modern information environment. [Electronic resource]. - Access mode: https://buki.com.ua/blogs/xmarni-sxovishha-nevidjemna-castina-sucasnogo-informaciinogo-seredovishha.

12. Ilya Reznikov. Cloud databases. Detailed instructions on how to apply a modern IT approach. [Access mode: https://dou.ua/forums/topic/43820.

13. Oleksandra Balykina. Cloud technologies and computing. Theory and practice. [Electronic resource]. - Access mode: https://www.sim-networks.com/ukr/blog/cloud-technologies.

14. Using cloud services for IoT. [Electronic resource]. - Access mode: https://pupenasan.github.io/TI40/%D0%9B%D0%B5%D0%BA%D1%86/cloud.html.

15. Models of cloud services. [Electronic resource]. - Access mode: https://www.sim-networks.com/ukr/blog/cloud-computing-service-models.

Yuriy Skorin **Юрій Скорін**	PhD, Associate Professor of Information Systems Department, Simon Kuznets Kharkiv National University of Economics, Kharkiv, Ukraine. e-mail: skorin.yuriy@gmail.com https://orcid.org/0000-0002-4613-3154	к.т.н., доц. кафедри інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, Харків, Україна.

Iryna Zolotaryova **Ірина Золотарьова**	PhD, Professor of Information Systems Department, Simon Kuznets Kharkiv National University of Economics, Kharkiv, Ukraine e-mail: iryna.zolotaryova@hneu.net https://orcid.org/0000-0002-1553-2849	к.е.н., професор кафедри інформаційних систем, Харківський національний економічний університет ім. С. Кузнеця, Харків, Україна.
Yuriy Lystopad **Юрій Листопад**	student, Simon Kuznets Kharkiv National University of Economics, Kharkiv, Ukraine. e-mail: listopadyura23@gmail.com https://orcid.org/0009-0001-1398-9925	студент, Харківський національний економічний університет ім. С. Кузнеця, Харків, Україна.