Journal of Scientific & Industrial Research
Vol. 85, February 2026, pp. 115-127
DOI: 10.56042/jsir.v85i2.14212

An Insight from A Programmer's Perspective on Cloud Container Security Architecture

Aleksandar Jovanović[1], Dušan Simjanović[1], Vladimir Lukić[1], Petar Milić[2]*, Dragan Savić[1] & Nemanja Zdravković[1]

[1]Belgrade Metropolitan University, Tadeuša Košćuška 63, 11158 Belgrade, Serbia

[2]University of Pristina —Kosovska Mitrovica, Faculty of Technical Sciences, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Serbia

Received 06 October 2024; revised 28 April 2025; accepted 16 January 2026

In today's cloud solution deployments, an important question to consider is whether your development and delivery infrastructure is genuinely secure when using standard security scanning methods and depending on the security protocols of major data providers like AWS, Google, Microsoft Azure, and Kubernetes for application deployment. The references analyzed on common threats in cloud environments and existing solutions, instantiated a need for a survey among 50 software professionals with varied experience in cloud and traditional security to be conducted. The findings highlighted the need for the ranking of common cloud threats in software development for cloud platforms by using Analytic Hierarchy Process (AHP) analysis. The results indicate that safeguarding cloud environments demands a multifaceted approach that addresses the nuanced challenges posed by out-of-date applications, operating system vulnerabilities, and third-party apps, all while remaining vigilant against other miscellaneous threats. Notably, third-party applications, while speeding up software delivery, pose significant security risks. This supports the "shifting left" paradigm, which emphasizes integrating security early in the development cycle. Additionally, the importance of a protective layer between hosts and containers through a common response protocol is determined. Docker accounts for 54.7% of the total deployment, showing that more than half of the respondents deployed container images using Docker. With values of 0.526 with $\lambda = 0.5$ of FAHP (Fuzzy Analytic Hierarchy Process), experts were either not sure or declared containers as non-repudiable, showing that programmers do not know if the container is the one it poses to be.

Keywords: Analytic hierarchy process, Docker, Fuzzy analytic hierarchy process, Fuzzy logic, Security

Introduction

As data storage requirements continue to increase, deployment deadlines, which should accommodate these growing data volumes, are becoming progressively shorter. This highlights the need for a prompt and scalable response. In response to this, the apps' functioning potential becomes more dependent on cloud infrastructure utilization as a fuel. One aspect of this so-called "cloud-native" approach infers application deployment methods' adoption of containerization and Infrastructure-as-Code (IaC) principles, which are characterized by scalability and agility. Zero-trust architectures are based on the principle that no one, whether internal or external to the network, is trusted by default. Access is denied to the system unless explicitly verified, regardless of the individual's known identity.[1]

Shifting left in application development means starting testing like quality assurance and performance checks early in the process, before much of the code is written. This helps identify security issues early, as the code must follow specific rules set by the software architect. However, around 95% of software today is developed without following this approach, which increases the risk of security problems arising later in development.

Containerized structures have their benefits and hindrances. As a benefit, a structure based on containers makes it indicative for quick deployment, patching, and scaling of applications including those with micro services as base principle. Moreover, containerized structures allow for consistent performance while multiple different operating systems are in use or hardware platforms are involved.[2] Containerized approach presents an ideal solution for a cloud-native based app. However, the very structure of containers and the beneficial sides of these systems are in the shadow of creating additional pathways to supply chains where threats can infiltrate the system. To avoid security issues, a starting point would be to indicate software type within the

*Author for Correspondence
E-mail: petar.milic@pr.ac.rs

container. Also, it is necessary to analyze in detail which types of software are more likely and how much more than others, to infiltrate the system and to determine why this happens.

In this paper, the focus is on whether the established data security practices, outside the cloud environment, can provide methodologically significant approaches adaptable to cloud security to ensure an adequate security level in the cloud. To achieve this, an AHP analysis of the results is conducted, allowing for the ranking of common threats based on their importance for software development and its distribution on cloud platforms. By addressing issues of confidentiality, availability, integrity, usability, and non-repudiation, particularly in relation to known and obscure container images, this ranking serves as an introduction to a methodological approach for handling cloud security. This approach aims to identify the security aspects most crucial for cloud security and analyze the interrelations between different parameters. The goal is to highlight key solutions for cloud security methodology for the containers used daily.

Literature Review

According to the definition in the report by Synopsis,[3] the development with containerization of applications, comprises initial base image design, which is in most cases based on an open-source code, and then building an application from it. This further involves layers of additional third-party and custom-based code which developers build on top of the base image.[4] Nevertheless, there are many ways based upon which cloud-related problems and models for its functioning could be defined. As indicated by Pezzella, cloud infrastructures bear similarities to different chemical or mathematical systems.[5] To facilitate virtualization and enable effective container orchestration, servers must function in an automated way, being provisioned based on specifications and instantiated as required. The aforementioned specifications, which are constructed by teams that use software Terraform or Ansible, allow operations to be executed in automated modes. Also, some IaC (Infrastructure as Code) principles include, apart from Ansible, TOSCA procedures and Helm charts, in order to reduce manual process and allow for portability among different surroundings and flexibility as indicated by Giommi *et al*.[6]

Careless behavior of software architects or developers as well as intentional back doors, created by unauthorized privilege escalations or network exposures within the infrastructure configuration, to name but a few, are not a new risk. Nonetheless, when it comes to who is responsible for building and assembling server configuration that indicates a new dimension, as argued by Salaki & Tini.[7] In the past decades, this used to be part of the daily routine job of IT operations engineer – a person trained and experienced in configuring servers securely, anticipating possible threats that may come and dealing with them accordingly to the common protocols of security. In recent times, the level of responsibility role has shifted towards development teams (DevOps).[8] The latter will use IaC principles to easily specify deployment configurations in order to build an app.

Neither scalability nor granularity adaptation of software should be forgotten. For example, Hassan *et al*.[9] discuss the topics concerning scalability dimensions and metrics unique to microservices, along with a methodical approach to scalability assessment for making informed adaptation decisions in modern software implementations. They suggest that establishing a comprehensive set of scalability dimensions and metrics specific to microservices, coupled with innovative applications of scalability analysis tailored to microservice granularity adjustment, can significantly enhance operational efficiency. Likewise, researchers examine and explore the decision-making process involved in microservices through a thorough theoretical analysis, offering notable insights in this domain.[10] However, the theoretical framework presented in this study primarily focuses on application and maintenance aspects, overlooking various other factors which shall be discussed further in the paper and which are to be found within the results of the questionnaire presented in detail. As an important input for the paper flow and aims, de Toledo *et al*.[11] argue that refactoring architectural technical debt, such as lack of communication standards, poor management of dead-letter queues, and the use of inadequate technologies in microservices, reduce the number of critical and high-priority incidents while some low priority incidents may increase. There is a high variety of incident types in microservices and therefore a need for some kind of ranking in terms of their importance for incidence reduction, is evident.

Security issue should also be considered in cloud container orchestration environment as it represents an essential aspect at all stages of the cloud

infrastructure lifecycle.[12] Paladi *et al.*[13] claim that adversaries can leverage misconfigured or maliciously modified orchestrators, as well as forged configuration policies and intents, to conduct attacks on cloud infrastructure. They propose the usage of orchestration security architecture which is flexible in terms of extending security features in an existing orchestration framework. Is this flexibility in line with some hierarchical model that needs to be established? Kulathunga[14] provides an interesting solution for the improvement of security of containerized orchestration environments which is reflected in the application of specific Intrusion Detection Systems (IDS). This research raises the level of security importance of the Kubernetes container orchestration by proposing a comprehensive model for evaluation of security containers for web applications, database applications, multimedia applications etc. Its results strongly indicate a possible path towards a universal approach to the problem of cloud container security methodology. Abdelmassih[15] extends the research in this field by introducing the "closures" which prevents an application from reaching another through the shared hardware, OS or network, given that no such access is required and "traceability", where the system must allow network monitoring of communications such that an IDS, may be utilized to detect network propagated attacks. The study's results help in narrowing the approach down to cloud security and advocating for custom approaches.

In the following chapters of this paper, the conducted research of faults occurrence of mistakes and oversights is examined. It is indicative to happen due to inexperience within the area among software engineer and DevOps team members, which allow for attackers to infiltrate the infrastructure. This puts attackers in a position to affect and compromise the credentials of the apps' users, operators and =data systems and allows for their vulnerability in the future.

Materials and Methods

The aim of the paper is to determine the security aspects most significant for cloud security and to give an inter-relation analysis of different parameters as contribution to current papers that only tackle each or a set of parameters as their main focus. The aim is to point towards the main solutions for cloud security methodology for containers which are used on daily basis. Firstly, the results of the experience levels among the interviewees in terms of different IT aspects were presented. Then, a comparison of the results was conducted within different matrixes and results to convey the most significant results of our findings and determine the ranking criteria.

The second part of the paper focuses on different cloud providers and their importance for creating general overview on cloud container critical circumstances, virtualization types and specifics. The latter analysis is especially important for creating custom approaches in contrast with security apps offered on the market that are generic and ignorant on the criteria mentioned above.

Our methodological approach utilizes an applied fuzzy mathematical model to highlight the most relevant results for cloud containers. Additionally, we propose a model for its application in real-world environments. In the analysis, both AHP and FAHP were used, incorporating five perspectives: pessimistic, semi-pessimistic, balanced, semi-optimistic, and optimistic. This led to the ranking of individual sub-criteria. In the FAHP process, we first calculated the fuzzy comparison matrix and weights for the five primary criteria, which are detailed below. Notably, the matrix shows consistency with a Consistency Ratio (CR) of 0.008117, which is well below the 0.1 threshold.

Survey Structure

The survey consisted of two parts, Part One – general security questions and Part Two – container images deployment using different cloud providers. The list of questions is given in Table 1 and 2.

Applied Fuzzy Logic Methodology for Analysis of Results' Comparison and Analysis

For more than a half of century, one of the useful tools to deal with uncertainty and imprecise linguistic statements, the fuzzy sets theory, has represented a significant support to decision-making problems.[16,17] Primarily, the aim of fuzzy sets and generalization of non-fuzzy sets, was the mathematical presentation of linguistic variables, enabling the decision-maker to make a model for partially unknown or incomplete information.[18,19] In crisp set theory, the element belongs to a set or not, while in the theory of fuzzy sets the membership function (MF) is introduced. It serves to map each element of the universal set into the interval [0,1], determining the degree of belongings of an element to a fuzzy set.

Let all fuzzy sets defined on the set of real numbers $\mathbb{R}$ be denoted as FS ($\mathbb{R}$). The number $G \in FS\ (\mathbb{R})$ is a

Table 1 — Part One – general security questions

Q1.1	How would you best describe your experience with security in general?
Q1.2	In which areas of IT security would you consider yourself most experienced and how much?
Q1.3	Your field and area of expertise and experience working with Cloud technologies?
Q1.4	In your opinion, what it is the importance of the following aspects of cloud containers' security?
Q2.1	What is your opinion on the significance of the software projects for the specific know-how?
Q2.2	Was security an important issue for the software projects you participated in?
Q2.3	Is there any connection between what you used in the past (virtual machines or similar) and cloud environments and what is the difference in their handling challenges?
Q3.1	Were there any security-based critical circumstances of using cloud, and, if yes, have these had some role in the development of the project, to your knowledge?
Q3.2	What were the top breaches causes in your cloud environments?
Q3.3	Please indicate the importance of cloud container images scanning before usage:
Q3.4	Please indicate the importance of cloud container images scanning before usage for known/obscure images.

Table 2 — Part Two – Container images deployment using different cloud providers

Q4.1	What is the great percentage of cloud container images deployed with, within your daily work?
Q4.2	If you specified Other in the previous question, please provide the name of cloud container?
Q4.3	What were the virtualization types?
Q4.4	If you specified Other in the previous question, please provide the name for this virtualization type (desktop, application, server, network, storage virtualization or other type-specific specification)?
Q4.5	Were there any policies regarding security groups, network rules, firewall rules, security rules?
Q4.6	Were cloud containers that you have used in your Cloud environment deployment confidential, available, usable, non-repudiable, with high integrity?
Q4.7	In your opinion, what it is the importance of the following aspects of cloud containers' security?
Q4.8	How important is rapid response for maintaining cloud container security?
Q4.9	Are there any specific mundane conditions under which cloud security is affected normally?
Q5.0	Please enter your e-mail address in case you would like to be informed about the outcome of the survey and be provided with a link to the final research paper: (excluded from the analysis as question).

fuzzy number if there exists$x_0 \in \mathbb{R}$ so it holds $\mu_G(x_0) = 1$ and for every$\lambda \in [0,1]$, $G_\lambda = [x, \mu_{G_\lambda}(x) \geq \lambda]$ is a closed interval.[20]

The fundamental part of triangular fuzzy number (TFN), its membership function, is defined as follows:

$$\mu_{TFN}(x) = \begin{cases} \dfrac{x-l}{m-l}, l \leq x \leq m \\ \dfrac{u-x}{u-m}, m \leq x \leq u \\ 0, otherwise, \end{cases}$$

where, inequality $l \leq m \leq u$ holds. Numbers l, m and u serve as the lower, middle, and upper value of G respectively, while for $l = m = u$, TFN becomes a crisp number. Usual notation of the triangular fuzzy number will be $\tilde{G} = (l, m, u)$.

Left and right side of the membership function $\mu_{TFN}(x)$ of TFN $\tilde{G} = (l, m, u)$, $\mu_{\tilde{G}}^l$ and $\mu_{\tilde{G}}^r$, as well as their matching inverse functions $(\mu_{\tilde{G}}^l)^{-1}$ and $(\mu_{\tilde{G}}^l)^{-1}$are respectively defined as $\mu_{\tilde{G}}^l = \frac{x-l}{m-l}$, $\mu_{\tilde{G}}^r = \frac{u-x}{u-m}$, $(\mu_{\tilde{G}}^l)^{-1} = l + (m-l)y, (\mu_{\tilde{G}}^r)^{-1} = u + (m-u)y, y \in [0,1]$. The total integral value, as a combination of left and right integral values is determined as follows:[21]

$$I_T^\lambda(\tilde{G}) = \lambda I_R(\tilde{G}) + (1-\lambda) I_L(\tilde{G}) = \lambda \int_0^1 (\mu_{\tilde{G}}^r)^{-1} dy + (1-\lambda) \int_0^1 (\mu_{\tilde{G}}^l)^{-1} dy =$$

$$= \frac{1}{2}\lambda(m+u) + \frac{1}{2}(1-\lambda)(m+l) = \frac{1}{2}(\lambda u + m + 1 - \lambda l,$$

where, λ represents an optimism index, i.e. the attitude of an expert during decision-making process. The pessimistic point of view is presented taking the value λ=0, from where it is obtained that $I_T^0(\tilde{G}) = I_L(\tilde{G})$, for the value λ=1, the optimistic point of view is given, and $I_T^1(\tilde{G}) = I_R(\tilde{G})$. For λ=0.5, the balanced (moderate) attitude of the decision-maker is granted, and $I_T^{0.5}(\tilde{G}) = \frac{1}{2}\left(I_L(\tilde{G}) + I_G(\tilde{G})\right)$. There are also, recently introduced, semi-pessimistic, and semi-optimistic points of view, obtained for λ=0.25 and λ=0.75 respectively.[22]

The main unary and binary operations for TFNs $G_1 = (l_1, m_1, u_1)$ and $G_2 = (l_2, m_2, u_2)$ and scalar $k > 0, k \in \mathbb{R}$ are shown below:[23,24]

(Addition): $\tilde{G}_1 \oplus \tilde{G}_2=(l_1,m_1,u_1) \oplus (l_2,m_2,u_2)=(l_1+l2,m1+m2,u1+u2$,
(Subtraction): $\tilde{G}_1 \ominus \tilde{G}_2=(l_1,m_1,u_1) \ominus (l_2,m_2,u_2)=(l_1-u2,m1-m2,u1-l2$,
(Multiplication): $\tilde{G}_1 \otimes \tilde{G}_2 = (l_1,m_1,u_1) \otimes (l_2,m_2,u_2) = (l_1 \cdot l_2, m_1 \cdot m_2, u_1 \cdot u_2)$,
(Scalar multiplication): $k \cdot \tilde{G}_1 = k \cdot (l_1,m_1,u_1) = (k \cdot l_1, k \cdot m_1, k \cdot u_1)$,
(Inverse): $\tilde{G}_1^{-1} = (l_1,m_1,u_1)^{-1} = \left(\frac{1}{u_1},\frac{1}{m_1},\frac{1}{l_1}\right)$.

In the sequel, the steps of the Fuzzy Analytic Hierarchy Process are summarized:[20,25]

Step 1: Establishing the Main Goal and Hierarchical Appearance of Criteria

The hierarchical structure, with the main goal as the most important component, at the top, has been organized vertically. The criteria and sub-criteria affecting the goal are at the intermediate levels, with alternatives at the lowest level (Fig. 1).

Step 2: Setting the Matrix in Terms of Triangular Fuzzy Numbers

Criteria and sub-criteria are used during the pair-wise comparisons, enabling the creation of matrix $\tilde{H} = \left(\tilde{h}_{ij}\right)_{n\times n}$. The total of n(n-1)/2 comparisons of elements from a higher level with elements from a lower level are made. Using triangular fuzzy numbers (TFNs), the hierarchy and comparison are given, where $\tilde{h}_{ij}$ is a fuzzy value representing the relative importance of one criterion to another. It holds that $\tilde{h}_{ii} = (1,1,1)$, when comparing criteria to itself, and $\tilde{h}_{ij} = 1/\tilde{h}_{ji}$ for $i \neq j$.

The fuzzy scale, TFNs, and their explanations used to enable pairwise comparisons are given: $\tilde{1} = (1,1,3)$ (Two criteria are equally important), $\tilde{3} = (1,3,5)$ (One criteria is slightly more important than another), $\tilde{5} = (3,5,7)$ (One criteria is strongly more important than another), $\tilde{7} = (5,7,9)$(One criteria is very strongly more important than another), $\tilde{9} = (7,9,9)$ (One criteria is absolutely strongly more important than another), $\tilde{2} = (1,2,3)$, $\tilde{4} = (3,4,5)$, $\tilde{6} = (5,6,7)$, and $\tilde{8} = (7,8,9)$ are intermediate values used when compromise is needed.[26,27] The graphic representation of the used FAHP scale with all three values (lower, median, and upper) is presented in Fig. 2.

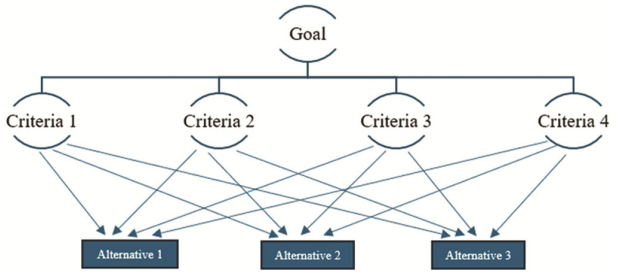

Fig. 1 — The hierarchical structure of the Analytic Hierarchy Process (AHP)

Step 3: Matrix Consistency Calculation

For matrix $H = \left(h_{ij}\right)_{n\times n}$we calculate the consistency index *CI* and consistency ratio *CR* using formulas

$CI = \frac{\lambda_{max}-n}{n-1}$, $CR = \frac{CI}{RI}$,

where, λ_{max}represents maximal eigenvalue of matrices H. The random index RI, determined by the matrix size and corresponding value, is shown as *RI = {(3, 0.58), (4, 0.9), (5, 1.12), (6, 1.24), (7, 1.32), (8, 1,41), (9, 1.45), (10, 1.49)}*. The value $CR<0.1$ verifies the matrix H consistency while differently, the reason for inconsistency should be determined, and all calculations redone.

Step 4: The Fuzzification Process

Applying formulas

$$D = \sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{h}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}(l_{ij},m_{ij},u_{ij})$$

and

$$D^{-1} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{h}_{ij}\right)^{-1}$$

$$= \left(\frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n}u_{ij}},\frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n}m_{ij}},\frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n}l_{ij}}\right)$$

on triangular fuzzy numbers from the matrix $H = \left(h_{ij}\right)_{n\times n}$, the Chang synthetic fuzzy number $\tilde{S}_i = (l_i,m_i,u_i) = \sum_{j=1}^{n}\tilde{h}_{ij} \otimes D^{-1}$, $i = \overline{1,n}$ is obtained.[23]

Step 5: The Defuzzification Process

Applying the formula

$w_i = I_T^{\lambda}(\tilde{S}_i) = 0.5(\lambda u_i + m_i + (1-\lambda)l_i), i = \overline{1,n}, \lambda \in [0,1]$, on obtained TFNs$\tilde{S}_i$, $\overline{1,n}$, the total integral value is calculated.

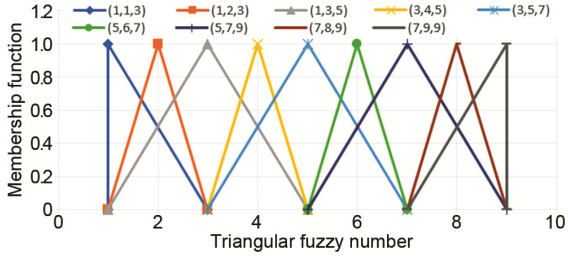

Fig. 2 — Graphic representation of triangular fuzzy numbers

Step 6: Vector Normalization and Criteria Weight Calculation

The weight vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^T$ is normalized using formula

$$w_i^* = w_i \left(\sum_{i=1}^{n} w_i \right)^{-1}$$

After this, criteria ranking and the comparison of alternatives is performed.

As outcomes of this methodology described, the results and discussion of the paper are centered around two parts. The first part of the results, based on the methodology described above, explores the relationship between factors such as years of experience, professional background, and recurring programming challenges related to security. This section discusses how the combination of professional experience and different job roles influences the ability to address common security issues encountered in everyday programming tasks. By analyzing the correlation between these factors, the results provide insights into the evolving cybersecurity expertise of software development practitioners. In the second part, responses from developers and experts were gathered to identify and enhance issues related to cloud security. This section reveals how practitioners navigate specific challenges related to containers and cloud infrastructure in their daily work. By drawing on firsthand experience and industry practices, the paper examines the practical obstacles and innovative solutions emerging from the intersection of modern software development approaches and cloud-native architectures.

Results and Discussion

Part One

The first analysis section is presented below.

Keeping in mind that proficiency (Fig. 3) in cloud technologies is increasingly critical for effective security management in modern IT environments, they may offer diverse perspectives and adaptable problem-solving skills essential for addressing multifaceted security challenges. This is especially important for the dynamic and scalable nature of cloud environments, which can potentially reduce time-to-resolution issues for cloud-related security incidents.

If we consider the server security and Cloud security experience among the interviewees presented in Fig. 4, the even distribution under both of these two criteria in the overall sample, shows that the selected interviewees in the sample were equally competent for server-based aspects and therefore their answers are considered relevant for cloud-based systems, with which they bear similarities in many of the security aspects, including most inner and outer influences to the system, as presented in Fig. 5.

Regarding the question Q3.1., and its focus on specific security-based critical circumstances of using cloud, the analysis follows.

Within the overall sample, 24% of the respondents answered with “no”, and less than 1% answered the question with “not applicable”. Others (75%) gave

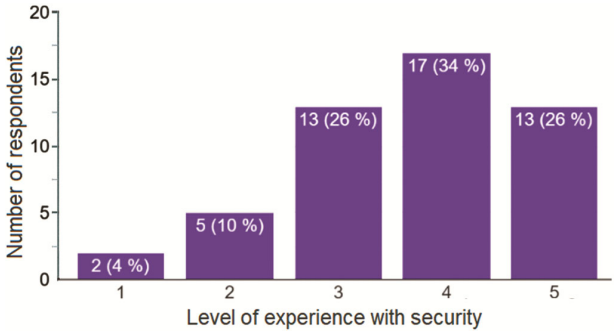

Fig. 3 — Experience with security among the interviewees (1 – little to no experience; 2 – somewhat experienced; 3 – neither experienced nor inexperienced; 4 – experienced; 5 – highly experienced)

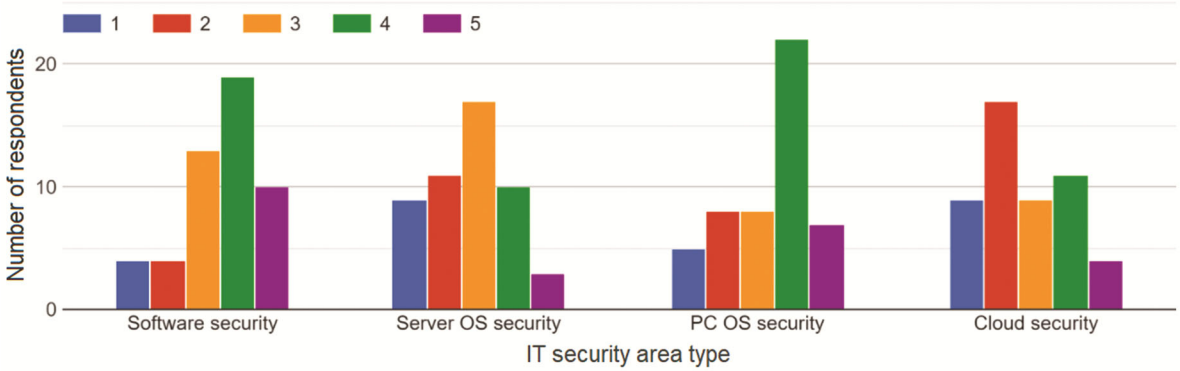

Fig. 4 — Experience with different IT security areas among the interviewees (1 – little to no experience; 2 – somewhat experienced; 3 – neither experienced nor inexperienced; 4 – experienced; 5 – highly experienced)

specific answers. Some answers stated that, "Compared to deployment on local server, deployment to the cloud requires extra care in configuration when connecting different services (APIs, DBs etc.) in order not to expose services needlessly."

Others pointed towards software log4j, regarding it as an applied software solution in terms of cloud globally. Further on, the programmer stated that "We had the similar issue with elastic search service that was relying on that package. We updated the ES and also changed a few things when it comes to storing and communication with other services."

Further answers pointed towards the issues of misconfiguration, communication as cloud critical circumstances aspects. Less than 3% of interviewees stated they had decided to skip programming steps because of security reasons i.e. deciding not to bring some features in production etc. Around 1% of interviewees indicated "regulations if hosted outside of the country" as a potent issue, while the same percentage indicated scaling key management, authentication and "none other than already established secure programming and administration best practices" as relevant for their projects as developers.

Part Two

The second analysis part is presented below. It discusses container images deployment aspects using different cloud providers and virtualization types.

The results of the analysis performed regarding this question represent the usage distribution of different container orchestration platforms among cloud users. Docker and Amazon ECS are the leaders in the area, followed by OpenShift and GKE/Kubernetes (Fig. 6). The presence of "Other" suggests that there are additional platforms being used. These results may place further questions such as why certain platforms are favored over others, what specific features or advantages each platform offers, and how this distribution might change over time as new technologies emerge or the existing ones evolve. Possibly, this is due to their ease of use, portability, ecosystem support, and tight integration with cloud platforms like AWS. The smaller distribution of usage for OpenShift and GKE/Kubernetes compared to Docker and Amazon ECS can be attributed to a combination of factors related to complexity, resource requirements, enterprise focus, licensing and market positioning. However, it is worth mentioning that the landscape is constantly evolving, and adoption

Fig. 5 — Importance of different aspect of cloud container security (Confidentiality, Availability, Integrity, Usability, Non-repudiation) breakdown by different cloud container security aspects

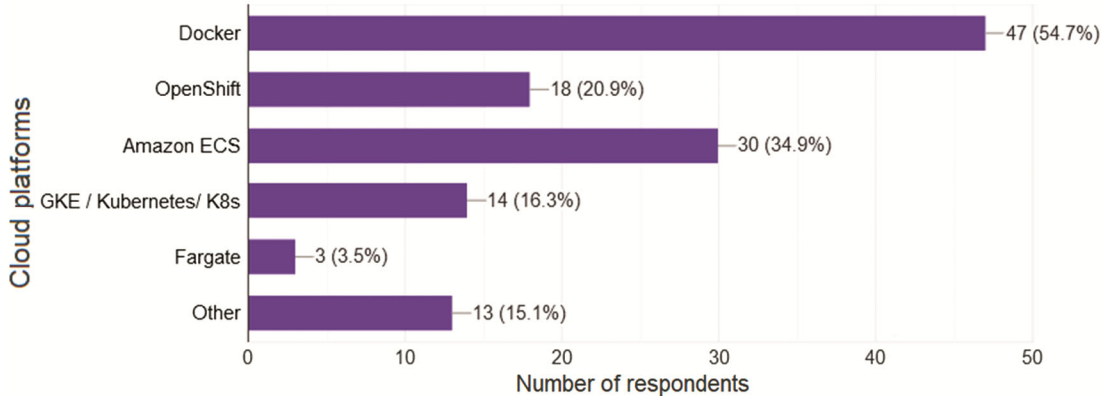

Fig. 6 — Cloud container images deployed within experts' daily work

patterns may shift over time as users become more familiar with container orchestration technologies and new features and improvements are being introduced.

Obtained results for this question encompasses a diverse range of virtualization technologies, such as VMware, QEMU, KVM and Other. VMware emerges as a dominant player in the virtualization space. It captures a significant portion of the total distribution, reflecting their widespread adoption and robust feature sets, via comprehensive suite of management tools and enterprise-grade features, making it a preferred choice for organizations seeking reliable and scalable virtualization solutions. Similarly, KVM's integration with the Linux kernel and support for hardware virtualization acceleration technologies elucidate its position in the results' distribution. QEMU's open-source nature and flexibility appeal to the users seeking lightweight and customizable virtualization options, particularly in development and testing scenarios. Nevertheless, as the demand for virtualization continues to evolve, driven by the trends such as cloud computing, containerization, and edge computing, the distribution of usage across virtualization platforms is expected to undergo further changes.

Results for this question indicate that security policies are very important in establishing and ensuring safe environment against evolving cyber threats and compliance challenges. All of them, security groups, network rules, firewall rules, and security rules, help developers in the implementation of layered security controls, enforcing access controls, and mitigation of risks in cloud environment. By adopting a proactive and holistic approach to cloud security, developers, as well as organizations can enhance trust, maintain data integrity, and ensure the confidentiality of sensitive information in the dynamic and interconnected world of cloud computing.

More details on this survey's results along with breakdowns of answers by different additional parameters can be found in the papers by Jovanovic *et al*.[28] and Jovanovic & Milic.[29]

AHP and FAHP Interpretation of Results

By the application of the AHP in cloud security and prioritizing various security threats and vulnerabilities, a systematic evaluation and ranking of these threats based on their relative importance is performed. The process begins with the identification of security criteria such as confidentiality, integrity, availability, and non-repudiation. These criteria are then broken down into sub-criteria and compared pairwise to establish their relative priorities. Surveyed results from IT and security experts provide the necessary data for these comparisons, resulting in a clear ranking of threats that helps in the formulation of effective security strategies.

FAHP extends the capabilities of AHP by incorporating fuzzy logic to handle the inherent uncertainties and ambiguities in expert judgments. In cloud security, where the evaluation of threats and vulnerabilities often involves subjective assessments, FAHP offers a more robust framework. It allows experts to express their preferences using linguistic variables such as "high", "medium" and "low" which are then converted into fuzzy numbers. This approach ensures that the imprecision in human judgment is adequately addressed. When applied to surveyed results, FAHP provides a more flexible and accurate prioritization of security issues.

By leveraging the strengths of AHP and FAHP, cloud security professionals can adopt a structured and methodical approach to threat assessment and mitigation. The hierarchical structuring of security criteria and the incorporation of fuzzy logic make these techniques particularly well-suited for the dynamic and complex nature of cloud environments. As cloud computing continues to evolve and grow, the application of AHP and FAHP will play a crucial role in maintaining robust security postures, enabling organizations to protect their data and systems effectively against an ever-expanding array of cyber threats.

Regarding this question in the survey which deals with different types of security (software security, server OS security, PC OS security, cloud security), the ranking of criteria for both AHP and FAHP for all established 5 levels of optimism (pessimistic, semi-pessimistic, balanced, semi-optimistic, and optimistic, with corresponding $\lambda = 0$, $\lambda = 0.25$, $\lambda = 0.5$, $\lambda = 0.75$ and $\lambda = 1$, respectively) is presented. In both AHP and all five FAHP cases, element "server OS security" showed the highest CI and CR values among all 4 analyzed. This is somewhat expected, as "server OS security" is a basic factor for ensuring security in network environment,[30] and therefore cloud. This confirms the previous work of Casalicchio and Iannucci[31] who claim that the description of the ways to control specific threats, fix of the vulnerabilities, or provision of the security attributes, makes building secure systems easier. Furthermore, AHP ranking of

our results shows that X2 is most influencing in "cloud security", "software security" and "PC OS security", while "server OS security" has lower results. Using FAHP, engineers and practitioners can fine-tune their actions to increase an aspect of security in cloud environment.

A comprehensive approach to cloud container security involves addressing the issues of confidentiality, availability, integrity, usability, and non-repudiation. Their balancing ensures that containerized environments are resilient against security threats, meet operational requirements, and provide a secure foundation for deploying and managing applications in the cloud. Survey results obtained for this question 1.4 are in line with the results of question 3.4. Known images used in cloud environment for deploying different kinds of infrastructures indicates that they are already tested, provide adequate level of security and ensure correct execution of the software. These findings are in line with the obtained results of AHP & FAHP analysis, where the confidentiality aspect is marked as one of the most important. It has been previously stated and explained why non-repudiation has lowest results. This parameter is crucial for establishing accountability and ensuring that actions taken within a containerized environment can be traced back to their originators. The orientation to using known images in cloud environment, which are at the same time confidential, decreases the need for non-repudiation aspect, as these two elements are in direct relationship, which is confirmed by the analysis and results shown in Table 3.

In the sequel, the calculation of Consistency index and Consistency ratio will be presented.

$$\lambda_{max} = \frac{5.018797 + 5.018797 + 5.011515 + 5.011515 + 5.005765}{5} = \frac{25.006639}{5} = 5.013278$$

$$CI = \frac{5.013278 - 1}{5 - 1} = 0.003319$$

$$CR = \frac{0.003319}{1.12} = 0.002964$$

Next, the calculation of criteria weights (using the comparison matrix) will be given.

$$X1 = X2 = \frac{\frac{1}{3.333} + \frac{1}{3.333} + \frac{2}{6.5} + \frac{2}{6.5} + \frac{3}{11}}{5} = 0.297622$$

$$X3 = X4 = \frac{\frac{0.5}{3.333} + \frac{0.5}{3.333} + \frac{1}{6.5} + \frac{1}{6.5} + \frac{2}{11}}{5} = 0.157902$$

$$X5 = \frac{\frac{0.333}{3.333} + \frac{0.333}{3.333} + \frac{0.5}{6.5} + \frac{0.5}{6.5} + \frac{1}{11}}{5} = 0.088951$$

Applying the elements of fuzzy comparison matrix, the intermediate results and fuzzy synthetic extents for criteria weights calculations, as well as defuzzification process in the case of FAHP are given below.

The sum of all l_i coordinates are equal: $l_1 = 1 + 1 + 1 + 1 + 1 = 5$, $l_2 = \frac{1}{3} + 1 + 1 + 1 + 1 = 4.333$, $l_3 = \frac{1}{3} + \frac{1}{3} + 1 + 1 + 1 = 3.667$, $l_4 = \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + 1 + 1 = 3$, and $l_4 = \frac{1}{5} + \frac{1}{5} + \frac{1}{3} + \frac{1}{3} + 1 = 2.066$.

The sum of all m_i coordinates are equal: $m_1 = 1 + 1 + 2 + 2 + 3 = 9$, $m_2 = 1 + 1 + 2 + 2 + 3 = 9$, $m_3 = \frac{1}{2} + \frac{1}{2} + 1 + 1 + 2 = 5$, $m_4 = \frac{1}{2} + \frac{1}{2} + 1 + 1 + 2 = 5$, and $m_5 = \frac{1}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{2} + 1 = 2.667$.

The sum of all u_i coordinates are equal: $u_1 = 1 + 3 + 3 + 3 + 5 = 15$, $u_2 = 1 + 1 + 3 + 3 + 5 = 13$, $u_3 = 1 + 1 + 1 + 3 + 3 = 9$, $u_4 = 1 + 1 + 1 + 1 + 3 = 7$, and $u_5 = 1 + 1 + 1 + 1 + 1 = 5$.

Their column sums are equal $l_{ij} = 18.0667$, $m_{ij} = 30.667$, and $u_{ij} = 49$, following that their inverse values are respectively equal 0.020408, 0.032609, and 0.055351.

The synthetic fuzzy numbers S_i determined as a product of sums of l_i, m_i, and u_i with the corresponding inverse values could be presented as in Fig. 7.

The defuzzification process in the FAHP pessimistic case could be explained as follows:

$w_1 = 0.5 \cdot (0 \cdot 0.830258 + 0.393478 + 1 \cdot 0.102041)$,
$w_2 = 0.5 \cdot (0 \cdot 0.719557 + 0.393478 + 1 \cdot 0.088435)$,
$w_3 = 0.5 \cdot (0 \cdot 0.498155 + 0.163043 + 1 \cdot 0.07483)$,

Table 3 — Results of the analysis for Q1.4

CI= 0.003CR= 0.003 λ is an optimism index

		AHP	FAHP				
			λ=0	λ=0.25	λ=0.5	λ=0.75	λ=1
X1=Confidentiality	X1	0.297622	0.288973	0.295498	0.299014	0.301212	0.302716
X2=Availability	X2	0.297622	0.279032	0.276118	0.274548	0.273567	0.272895
X3=Integrity	X3	0.157902	0.173794	0.175846	0.176952	0.177643	0.178116
X4=Usability	X4	0.157902	0.163854	0.156466	0.152486	0.149998	0.148295
X5=Non-repudiation	X5	0.088951	0.094347	0.096071	0.096999	0.09758	0.097977

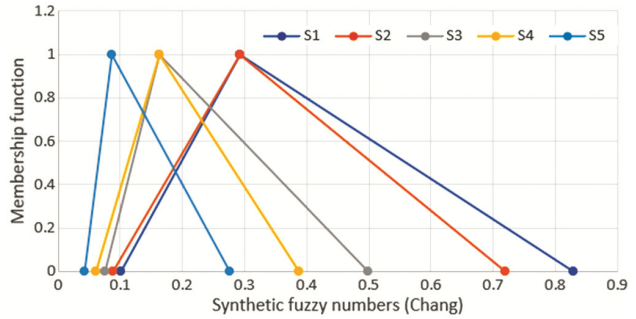

Fig. 7 — Chang synthetic fuzzy numbers

$w_4 = 0.5 \cdot (0 \cdot 0.387454 + 0.163043 + 1 \cdot 0.061244)$, $w_5 = 0.5 \cdot (0 \cdot 0.276753 + 0.086957 + 1 \cdot 0.042177)$, from where, using normalization pprocess, final weights as in Table 3 were obtained. Similar procedure is conducted when $\lambda = 0.5$ (balanced point of view), where $w_1 = 0.5 \cdot (0.5 \cdot 0.830258 + 0.393478 + 0.5 \cdot 0.102041)$, $w_2 = 0.5 \cdot (0.5 \cdot 0.719557 + 0.393478 + 0.5 \cdot 0.088435)$, $w_3 = 0.5 \cdot (0.5 \cdot 0.498155 + 0.163043 + 0.5 \cdot 0.07483)$, $w_4 = 0.5 \cdot (0.5 \cdot 0.387454 + 0.163043 + 0.5 \cdot 0.061244)$, $w_5 = 0.5 \cdot (0.5 \cdot 0.276753 + 0.086957 + 0.5 \cdot 0.042177)$.

A prevalent threat to cloud security stems from the use of out-of-date applications, which is confirmed by our analysis. The AHP values strongly support this claim. Likewise, since λ serves as an index of optimism in expert judgments, a high λ value indicates coherent and reliable expert input on specific security issues. This helps prevent flawed security decisions that could arise from inconsistent assessments. This effect becomes increasingly evident as software evolves, given that outdated applications are more likely to contain known vulnerabilities that have already been resolved in newer versions. The failure to regularly update applications in the cloud environment leaves a potential entry point for malicious actors seeking to exploit these vulnerabilities for unauthorized access or data compromise.

A More Focused Discussion

The choice between VMware, QEMU, and KVM depends on specific use cases and organizational preferences. VMware offers feature-rich proprietary solutions with widespread industry adoption, but comes at a potentially higher cost. Its hypervisor, ESXi, offers excellent performance, advanced management tools, and comprehensive support for various guest-operating systems. QEMU, as an open-source emulator, is versatile but may sacrifice some performance. It is well-suited for development environments and scenarios where emulation of diverse architectures is crucial. KVM, integrated into the Linux kernel, combines the benefits of open-source development, good performance, and cost-effectiveness, particularly for Linux-centric environments. Organizations should carefully evaluate their requirements and priorities to select the virtualization solution that best aligns with their needs.

The responses highlight the critical need for thorough scanning of container images before deployment as a key step in risk mitigation and proactive security management. Scanning commonly used container images helps detect and address known vulnerabilities and threats early in the development process. By using up-to-date vulnerability databases and malware signature repositories, modern scanning tools can effectively identify insecure components or compromised elements within images. Applying validated vulnerability assessment methods enables organizations to strengthen their containerized environments in advance, reducing the risk of exploitation and data breaches in real-world operations.

Additionally, many survey respondents emphasized that scanning should not be limited to well-known container images but must also include obscure or custom-built ones, which may contain hidden vulnerabilities or unauthorized components. Security best practices require that even internally developed or seemingly harmless images undergo thorough inspection to detect possible weaknesses or compliance issues. By using advanced scanning tools that can identify unusual patterns and unauthorized software, organizations can proactively address security risks in lesser-known container images—strengthening the overall security and reliability of their cloud-native applications.

Nevertheless, by monitoring changes in this field, certain emerging trends and shifts in the containerization landscape over time can be revealed. For instance, an increase in the percentage of deployments on Kubernetes may signify a growing preference for container orchestration platforms offering advanced scalability and management capabilities.

It is important to emphasize that the percentages provided are based on the responses collected and may not accurately reflect the entire population. Sampling bias, survey methodology, and respondent demographics could influence the distribution observed.

Based on the results (Tables 4 & 5) of questions Q4.1 and Q4.2, using AHP method, there is a

Table 4 — Results of the analysis for Q4.1

CI = 0.033CR = 0.026			λ is an optimism index				
		AHP	FAHP				
			λ = 0	λ = 0.25	λ = 0.5	λ = 0.75	λ = 1
X1 = Docker	X1	0.4408063	0.404744966	0.392008595	0.384052246	0.378610149	0.374653138
X2 = Amazon ECS	X2	0.219864521	0.228237733	0.231247157	0.233127129	0.234413019	0.235348005
X3 = Openshift	X3	0.14507549	0.160630232	0.162196219	0.163174483	0.16384361	0.164330139
X4 = GKE/Kubernetes K8s	X4	0.093765515	0.100673547	0.107638841	0.111990026	0.114966211	0.11713023
X5 = Other	X5	0.060326725	0.065671707	0.066847709	0.067582352	0.068084843	0.06845021
X6 = Fargate	X6	0.04016145	0.040041815	0.04006148	0.040073765	0.040082168	0.040088278

Table 5 — Results of the analysis for Q4.2

CI = 0 CR = 0			λ is an optimism index				
		AHP	FAHP				
			λ = 0	λ = 0.25	λ = 0.5	λ = 0.75	λ = 1
X1 = Azure	X1	0.6	0.559585492	0.571640055	0.577948401	0.58182843	0.584455959
X2 = iCloud	X2	0.2	0.233160622	0.239690177	0.243107198	0.24520888	0.246632124
X3 = Colima	X3	0.2	0.207253886	0.188669768	0.178944401	0.17296269	0.168911917

relatively uneven distribution of each of the answers given. The use of FAHP method, may indicate the prevalence of Docker, Amazon ECS, OpenShift and GKE to the insignificant set of other and AWS Fargate. However, as in Q4.2, under Other, there are Microsoft Azure, ICloud and Colima, with AHP and FAHP (Fig. 6) with significance of all three cloud provider options to the ones in the major group from Q4.1. This indicated that despite AHP and FAHP values showed in favor of top four, in reality, the likelihood of employing not one but myriad of given cloud provider options for image container deployment, is present in experts' practice. And, as argued before in the introduction,[13,14] the results presented here are in line with creating both a universal and flexible approach for cloud container security maintenance.

The paper ranked the threats and identified obscure images as some of the pathways to security breaches. With values of 0.526 and λ = 0.5 of FAHP, experts were either not sure or declared containers as non-repudiable, showing that programmers do not know if the container is the one it poses to be. The least possible intrusion variant suggests creating a model for threat-safe environment established on the AHP analysis and ranking applied, as a holistic approach based on the experience of programmers, rather than sole data analysis carried out by common software solutions made so far. The identification of obscure images as a significant security risk underscores the need for comprehensive vetting and validation of container images before deployment. Obscure images, which may contain hidden vulnerabilities or a malicious code, pose a substantial threat to cloud security. The analysis recommended adopting stringent image scanning practices and utilizing trusted sources for container images to mitigate this risk.

The analysis revealed that a quick response is crucial for maintaining the security of cloud containers. A rapid response to potential threats is essential to mitigate the risks associated with unauthorized access and to maintain the integrity and confidentiality of data within containers. The survey results highlighted the programmers' emphasis on the need for immediate action to counteract threats, reflecting the real-time nature of cloud environments and the importance of agility in threat management.

Beyond the need for a quick response, the research has identified several other critical criteria for cloud container security. Experts emphasized the importance of thorough monitoring and auditing of container activities, ensuring that all actions are logged and reviewed for any suspicious behavior. This proactive approach enables the early detection of anomalies and potential breaches. Additionally, the implementation of robust authentication and authorization mechanisms was highlighted as a key factor in securing cloud containers. Ensuring that only authorized personnel have access to sensitive data and resources minimizes the risk of insider threats and unauthorized access. The use of encryption for data at rest and in transit was also recognized as a vital measure to protect data integrity and prevent eavesdropping or data tampering.

The study revealed important relationships between containers and the need to prioritize protecting the

host system from potential threats originating within containers. This suggests a complex and layered structure that could benefit from future uncertainty analysis. However, current technological capabilities in cloud security may not yet support such advanced analysis in the near term. Nonetheless, the findings of this paper reinforce the value of adopting the proposed methodology to drive continuous improvement in container security. They also highlight the potential of using FAHP to develop more advanced tools for vulnerability scanning and management. By focusing on these areas, organizations and IT professionals can strengthen their security defenses and reduce the risk of breaches caused by compromised container images.

Conclusions

Due to potential sampling bias, survey methods, and respondent demographics, future research should utilize an updated and more representative dataset to enhance the accuracy of findings. This would clarify whether standard procedures for securing cloud containers are being properly followed and help adjust ranking results where needed. The paper challenges existing practices in cloud container protection and proposes a foundation for a standardized methodology to evaluate security by highlighting the impact of various data breach types. It also incorporates insights from AHP and FAHP analyses, offering a valuable perspective on the evolving landscape of containerization and cloud infrastructure management. These findings contribute to a deeper understanding of current challenges in software engineering. Moving forward, further studies should explore uncertainty analysis models in containerized environments to uncover hidden risks. Expanding the respondent base would improve the generalizability and reliability of survey-driven research on cloud container security.

Acknowledgement

The authors would like to thank the Ministry of Science, Technological Development and Innovation of the Republic of Serbia for funding the scientific research work, contract no. 451-03-34/2026-03/200155, realized by the Faculty of Technical Sciences in Kosovska Mitrovica, University of Pristina.

References

1 Rose S, Borchert O, Mitchell S & Connelly S, Zero trust architecture, U.S. Department of Commerce-National Institute of Standards and Technology, *NIST Spec Publ*, 800-207, (2020), doi: 10.6028/NIST.SP.800-207

2 Thyagaturu A, Shantharama P, Nasrallah A & Reisslein M, Operating systems and hypervisors for network functions: A survey of enabling technologies and research studies, *IEEE Access*, **10** (2022), 79825–79873, doi: 10.1109/ACCESS.2022.3194913.

3 Synopis, Five considerations for supplying your software supply chain, *Synopis*, (2023), Retrieved from *www.synopsis.com*, (18 September 2023).

4 Devi Priya V S, Chakkaravarthy Sethuraman S & S, Khurram Khan M, Container security: precaution levels, mitigation strategies, and research perspectives, *Comput Secur*, **135** (2023) 103490, doi: 10.1016/j.cose.2023.103490.

5 Pezzella M, UniNuvola: the computing portal of the Perugia University, *Int Sym on Grids & Clouds (ISGC)*, Taiwan, (2024), Retrieved from https://indico4.twgrid.org/event/33/contributions/1409/attachments/794/1000/marco_pezzella_26032024.pdf.

6 Giommi L, Vianello E, Miccili R, Agostini F, Fornari F, Costantini A, Antonacci M, Vino G, Savarese G & Donvito G, Efficient management of INDIGO-IAM clients and S3 buckets via Indigo PaaS Orchestrator in INFN Cloud, in *Int Symp on Grids & Clouds (ISGC)*, Academia Sinica, Taiwan (2024), Retrieved from https://indico4.twgrid.org/event/33/contributions/1398/attachments/800/1007/Giommi_orchestrator_ISGC.pdf.

7 Salaki R J & Tini M, Reliability management: Setting-up cloud server in higher education, *Int J Innov Technol Explor Eng (IJITEE)*, **9(1)** (2019), https://doi:10.35940/ijitee.A4534.119119.

8 Pahl C, Jamshidi P & Zimmermann O, Architectural principles for cloud software, *ACM T Internet Techn*, **18(2)** (2017) doi: 10.1145/310402.

9 Hassan S, Bahsoon R & Buyya R, Systematic scalability analysis for microservices granularity adaptation design decisions, *J Soft Pract Exper*, **52(6)** (2022) 1378–1401, doi: 10.1002/spe.3069.

10 Li Y, Wang Ch-Zi, Li Y-Ch & Su J, Granularity decision of microservice splitting in view of maintainability and its innovation effect in government data sharing, *Discrete Dyn Nat Soc*, (2020) 1–11, doi: 10.1155/2020/1057902.

11 De Toledo S, Martini A, Sjoberg D I K, Przybyszenwska A & SkovFrandsen J, Reducing incidents in microservices by repaying architectural technical debt, In *47th Eu Conf Soft Engin Adv App (SEAA)*, IEEE (2021), 196–205, doi: 10.1109/SEAA53835.2021.00033.

12 Butt U A, Amin R, Mehmood M, Aldabbas H, Alharbi M T & Albaqami N, Cloud security threats and solutions: A survey. *Wirel Pers Comm*un, **128(1)** (2023) 387–413, doi: 10.1007/s11277-022-09960-z.

13 Paladi N, Michalas A & Dang H V, Towards secure cloud orchestration for multi-cloud deployments, In *Proc 5th Work Cross Cloud Inf & Platf*, (2018) 1–6, https://dl.acm.org/doi/10.1145/3195870.3195874.

14 Kulathunga R G K P, Dynamic security model for container orchestration platform, MSc thesis, University of Colombo School of Computing, (2020), Retrieved from https://dl.ucsc.cmb.ac.lk/jspui/handle/123456789/4533, (14 January 2026).

15 Abdelmassih C, Container orchestration in security demanding environments at the Swedish Police Authority, KTH Royal Institute of technology, MSc thesis (2018), Retrieved from DiVA, id: diva2:1231856, OAI:

oai:DiVA.org:kth-228531, urn:nbn:se:kth:diva-228531 (14 January 2025).
16 Zadeh L A, The concept of a linguistic variable and its application to approximate reasoning I, *Inf Sci*, **8** (1975) 199–249, doi: 10.1016/0020-0255(75)90036-5.
17 Zadeh L A, The concept of a linguistic variable and its application to approximate reasoning II, *Inf Sci*, **8** (1975) 301–357, doi: 10.1016/0020-0255(75)90046-8.
18 Zadeh L A, The concept of a linguistic variable and its application to approximate reasoning III, *Inf Sci,* **9** (1975) 43–80, doi: 10.1016/0020-0255(75)90017-1.
19 Chou J Sh, Pham A D & Wang H, Bidding strategy to support decision-making by integrating fuzzy AHP and regression-based simulation, *Automat Constr*, **35** (2013) 517–527, doi: 10.1016/j.autcon.2013.06.007.
20 Milošević D, Milošević M & Simjanović D, Implementation of adjusted fuzzy AHP method in the assessment for reuse of industrial buildings, *Math*, **8** (2020) 1697, doi: 10.3390/math8101697.
21 Kulak O, Durmusoglu M B & Kahraman C, Fuzzy multi-attribute equipment selection based on information axiom, *J Mater Process Techn*, **169** (2005) 337–345, doi: 10.1016/j.jmatprotec.2005.03.030.
22 Simjanović D, Zdravković N & Vesić O N, On the factors of successful e-commerce platform design during and after COVID-19 pandemic using extended fuzzy AHP method, *Axioms*, **11(3)** (2022) 105, doi: 10.3390/axioms11030105.
23 Chang D Y, Application of the extent analysis method on fuzzy AHP, *Eur J Oper Res*, **95** (1996) 649–655, doi: 10.1016/j.ejor.2007.01.050.
24 Wang W M, Lee A H I & Chang D T, An integrated FA-FEAHP approach on the social indicators of Taiwan's green building, *Glob Bus Econ Rev*, **11(3)** (2009) 304–316, Retrieved from https://ideas.repec.org/a/ids/gbusec/v11y2009i3-4p304-316.html, (14 January 2026)
25 Kahraman C, Cebeci U & Ruan D, Multi-attribute comparison of catering service companies using fuzzy AHP: The case of Turkey, *Int J Prod Econ*, **87(2)** (2004) 171–184, doi: 10.1016/S0925-5273(03)00099-9.
26 Domínguez S & Carnero M C, Fuzzy multicriteria modelling of decision making in the renewal of healthcare technologies, *Math*, **8(6)** (2020) 944, doi: 10.3390/math8060944.
27 Janackovic G L, Savic S M & Stankovic M S, Selection and ranking of occupational safety indicators based on fuzzy AHP: A case study in road construction companies, *SAJIE*, **24(3)** (2013) 175–189, doi: 10.7166/24-3-463
28 Jovanović A, Milić P & Saravathi V, Towards creating methodology for security assessment of cloud containers-an overview of available tools, *Proc 13th Int Conf Bus Inf Secur BISEC* (Belgrade Metropolitan University) 2022.
29 Jovanović A & Milić P, Custom made approaches to cloud container security: A methodologically sound approach based on international sample results, *Proc 13th Int Con Bus Inf Secur BISEC* (BMU) 2023.
30 Van Dijck J, Governing digital societies: Private platforms, public values, *CLSR*, **36** (2020) 1–4, doi: 10.1016/j.clsr.2019.105377.
31 Casalicchio E & Iannucci S, The state-of-the-art in container technologies: Application, orchestration and security, *Concurr Comput Pract Exp*, **32(17)** (2020) e5668, doi: 10.1002/cpe.5668.