International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Optimization of Cloud-Based Applications using DevOps

Sarthak Srivastava

Email: *sarthaksrivastava44[at]gmail.com*

Abstract: ***The rapid development of software is crucial in meeting company goals and keeping up with competitors in the highly competitive field of IoT infrastructure. Due to the increasing demand for new products and technologies, many organizations struggle with the burden of introducing software quickly while ensuring stability to compete with others. This has resulted in a preference for automated systems for product development and cloud-based applications. To achieve this, organizations leverage tools such as the git version control system for version management, Docker for code packaging and libraries, and AWS services for cloud deployment. Jenkins is used as a CI/CD pipeline to manage various phases of development, and the ELK stack is used for monitoring and visualizing code execution. The study findings indicate that DevOps is an efficient method for cloud application deployment and resource selection, considering value parameters like cost, memory, and CPU capacity. The proposed approach shows a cost reduction of 60% with full weight and 11.3% less with no weight, compared to the benchmark solution's 15.078%. The analysis also shows that DevOps techniques can be tailored to specific application requirements and effectively used for cloud deployment.***

Keywords: DevOps; Cloud pipelines; Continuous integration; Continuous development, Software Development lifecycle.

1. Introduction

The advancement of modern technology has brought numerous tools that contribute to our daily lives and improve working conditions. Organizations are creating software more quickly and regularly than ever before as a result of the rising demand for new software goods and technology. Consequently, many companies opt for automated product development programs that require cloud-based applications. Cloud collaboration using DevOps has emerged as a powerful solution, facilitating not only the development of software products but also the implementation and control of the deployment process. Leited et al. (2023) described DevOps as a collaborative and multidisciplinary effort within an organization to automate the continuous delivery of new software versions while guaranteeing their correctness and reliability. DevOps enables hundreds of tests to be run within a day and provides feedback from clients after each delivery, thereby helping organizations to explore additional features that can be implemented in the project and minimize configuration issues.

DevOps is crucial for cloud-based computing, including automated application deployment, Infrastructure as Code, and delivering servers. It is a vital part of cloud computing that manages infrastructure, application deployment, and application functionalities in various contexts. DevOps aids in making a high-quality product, continuous delivery, and providing quality software to end-users. It enables quick responses to changing client requirements and allows developers and operations to work in a shared environment. Despite its benefits, Ellen et al. (2021) identified poor communication, ingrained organizational culture, market limits, scalability, and diverse ecosystems as the major difficulties for DevOps acceptance in the software business. To resolve the issues that DevOps teams may experience during continuous integration, deployment, and testing, our study uses Jenkins as a helping tool to solve deployment-related problems. We have also employed the Jenkins pipeline to manage various phases of the deployment process and have selected a method for implementing DevOps using Continuous Delivery, building a specific Continuous Delivery system design based on Amazon Web Services (AWS) cloud.

The aim of our study is to propose an algorithm that supports cloud-based applications in employing DevOps techniques and addresses the challenge of developing, deploying, and managing IoT applications in a multi-cloud environment while remaining within the overall parameters of the existing organization ecosystem. The analysis findings revealed that an application can be deployed to the cloud using DevOps techniques (Leited et al., 2023).

2. Literature Survey

2.1 Background

According to Cois et al. [7], an application is deployed in the production environment only after the development team has completed its work and the operations team has independently created and configured the application's deployment environment. Application development involves phases such as coding, designing, integrating, testing, debugging, configuring infrastructure, access control, establishing runtime environment, and deploying experience in different environments, as described by Soni [8]. Barna et al. [9] proposed a method for creating an Autonomous Management System (AMS) for data-intensive containerized applications that are multi-tiered and multi-layered. DevOps plays a crucial role in cloud security when it comes to moving applications to the cloud, as security is a major concern for developers, as described by Jaatun et al. [17].

DevOps offers more comprehensive support for cloud application deployment using a variety of tools that offer automation and continuous integration (CI). Because software and its surroundings are so volatile, managing software change in a DevOps environment has become a crucial skill for a digital organization. Interoperability issues across cloud solutions and difficult maintenance and evolution management of complex cloud apps are the major challenges faced by large-scale and distributed cloud applications [20]. The goal of the DevOps paradigm is to reduce the software delivery cycle while maintaining excellent quality, as highlighted by Basiri et al. [23].

DevOps adoption in software organizations has been studied by several researchers, such as Akbar et al. [24], who discussed DevOps success factors, and Rafi et al. [25], who proposed a DevOps business model for small startups and work from home environment. A model for deploying apps on hybrid clouds and figuring out the optimal hosting match by choosing the ideal cloud to carry out the operation based on the location of the data center was put out by Venkateswaran and Sarkar [26].

According to Guerriero et al., DevOps is an essential part of cloud computing that controls infrastructure, application deployment, and application functions in a range of scenarios. [10]. DevOps facilitates the development of high-quality software at a rapid pace, as described by Arulkumar and Lathamanju [19].

2.2 Available Tools and Related Work

The software development industry widely uses code integration as it reduces the problem of integrating source code when software is built across multiple sites, allowing developers to quickly build, evaluate, analyze, and deploy software [27]. To simplify the process of assembling code, libraries, and configuration files into a single image and run an application as a container, developers use the Docker platform [2].

Figure 1 below shows how the application is containerized within Docker platform.

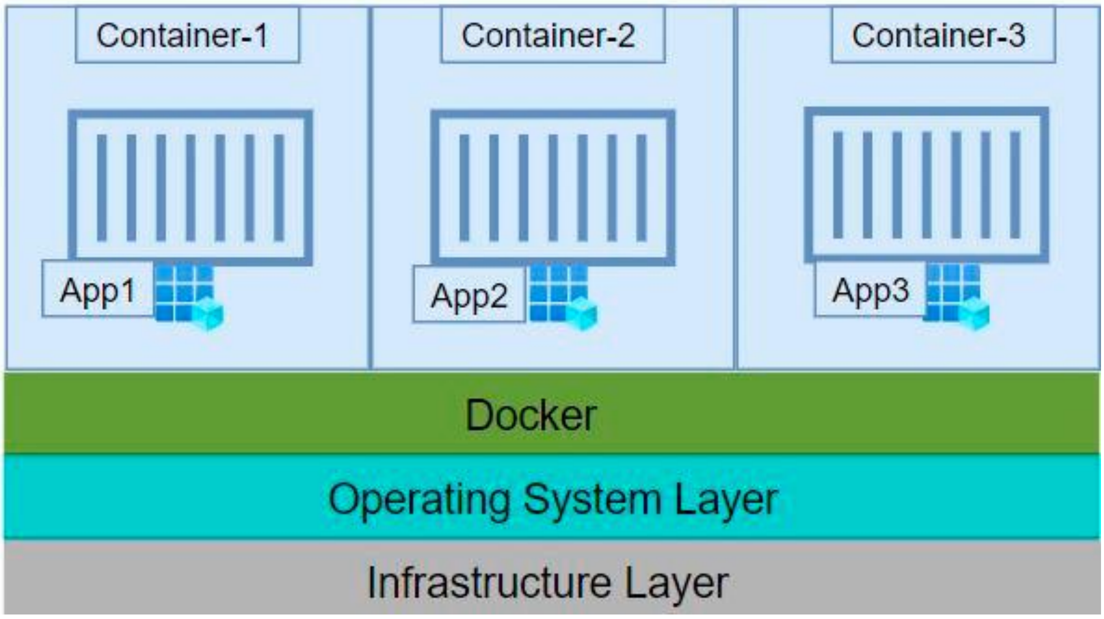

Figure 1: Containerized Application with Docker [2]

The CICD pipeline, which comprises many phases, including Designing, Configuration Management, Production Support, Continuous Integration, Iterative Development, and Continuous Deployment, is used to automate the development of builds from code and their subsequent deployment to the domain controller [28]. Continuous Integration is the first stage of the pipeline, which allows developers to merge their work into the repository frequently. Continuous Deployment is the process of systematically delivering ready code, while Continuous Delivery is a method of planning and preparing code for deployment.

2.3 Git version control

Adopting a version control system (VCS) can provide complete access to code while setting access controls to a limited number of people who work on it, determining when they commit changes and what changes are made [33]. This improves code security and ensures rapid delivery [34]. Popular VCSs include AWS CloudWatch [29], Git [30], Mercurial [31], and SVM [32], with Git being the most widely used and offering hosting services [35]. In fact, Git is used by many new businesses and students [36]. In our work, we also use Git to maintain our source code, tracking, committing, and pushing code to a dedicated storage place like GitHub [33].

2.4 Create Container Using Docker

According to Gupta et al. [35], Docker is a popular open-source tool for building, deploying and running applications by creating containers that provide a level of isolation for software. The containerization approach offered by Docker simplifies the deployment process by packaging the software into a self-contained unit, reducing the likelihood of compatibility issues. Docker also provides a layer of security by isolating the application and its dependencies from the underlying system.

The Docker architecture, as described by Calçado et al. [36], includes Docker CLI, Docker host, and Docker Registry. The Docker Daemon builds Docker images when they are created using the Docker build command on the Docker CLI. Docker Swarm is another important feature of Docker that helps to manage and maintain multiple containers. This is especially useful in software development where micro-services architecture is used to break down software into smaller components [37].

By developing containers that separate the program and its dependencies from the underlying system, Docker offers a quick and secure way to build and deploy applications. Multiple containers can be orchestrated with Docker Swarm, which is helpful when developing software that uses micro-services.

2.5 Utilize Jenkins for Continuous Integration

Because it is flexible, incredibly customization, and free, Jenkins is widely used in the software development sector. It is quite adjustable and can be used with a variety of processes because it supports a lot of plugins. In fact, Jenkins is the only continuous integration tool that supports such a huge number of plugins [36]. Jenkins' primary benefit is that it fully automates all phases of software development, from spotting changes in the source code through testing and delivering the code.

The Jenkins pipeline, as shown in Figure 2, is responsible for automating this process, which begins when a developer pushes their code to the source code manager, and then automatically pulls it through Jenkins with the help of jobs. Jobs are processes or instructions that explain how to perform an action or automate a process [37].

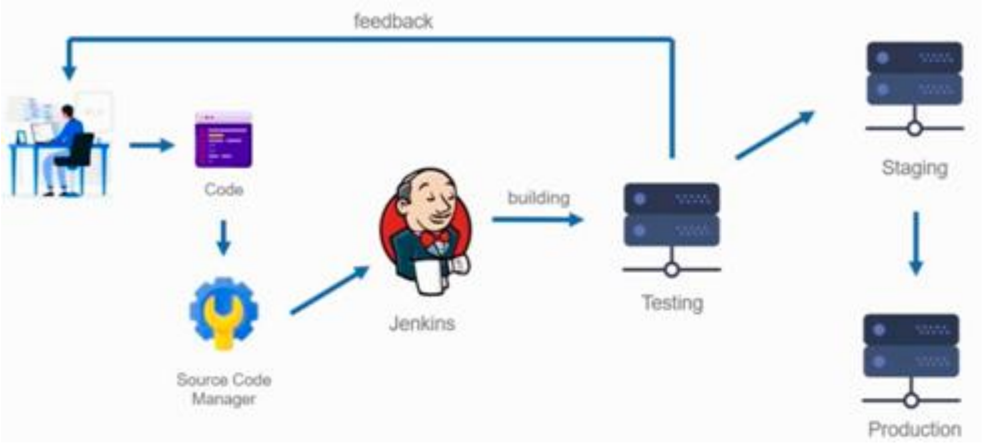

Figure 2: Jenkins pipeline (Li et al. [36])

2.6. Create Cloud Using AWS

AWS is a stage for distributed computing that makes it conceivable to send and work on virtual machines there. AWS EC2 (Elastic Compute Cloud) allows users to create and launch virtual machines, also known as instances, on the AWS cloud platform. These instances can be configured with different operating systems, including Ubuntu [37], and accessed remotely from a user's local system. Users must first register for an account on the AWS dashboard before proceeding to utilize the EC2 service to launch an instance on AWS.

2.7 Continuous Monitoring

The selection of monitoring tools depends on the software requirements, such as whether to monitor metrics or logs. In this study, Prometheus and Grafana were used for metric monitoring, as they are free and open-source tools that provide a dashboard to track various software performance metrics [38]. For collecting and visualizing logs, the ELK stack was utilized in this research [39].

To manage and operate version control systems and provide libraries, a robust and systematic way is needed. AWS services are employed to deploy cloud applications, while Jenkins is utilized as a CI/CD pipeline to manage the different phases of software development, thus ensuring continuous development. The ELK stack is utilized to monitor and visualize the code execution. Based on the findings of this research, it is concluded that DevOps is an effective approach to deploy cloud applications and select resources based on value parameters such as cost, memory, and CPU capacity. This approach can be tailored to meet specific software requirements [40].

3. Methodology

We conducted experiments to evaluate the effectiveness of our proposed algorithm in comparison to standard methods, by formulating a multi-objective optimization solution for creating deployment plans using ANOVA approach to analyze the relationship between two groups. The parts of information change are isolated into mistake or lingering variety and variety between treatments in the ANOVA table [40]. To determine the optimal solution, the Decision Maker takes inputs from various components including vendor services, information hubs, applications, and client footprints. The data-center ID, instance type (determined by memory size, CPU count, and storage), number of available instances, cost per hour, and evaluation inputs regarding vendor capabilities are sent to the resource trader. The Application Repository provides information on application node data and node interdependence, while the Information

Analyzer provides data on client footprint such as the Geo-positioning of POIs and expected traffic volume in the area.

We employed Algorithm 1 for optimizing objectives, which is based on Genetic Algorithm (GA). GA mimics the natural evolution process and each iteration brings the solution closer to the desired answer. This algorithm can also solve multi-objective optimization problems. Our proposed algorithm maps virtual machines to instances and creates an application deployment plan [41].

Algorithm 1. Optimizing objectives

```
begin:
    function createDeploymentPlan(application)
    set i = 1
    while(i ** maximum)
           function generateVirtualMachines()
           #generate virtual machines for the application
           end
    end #end while loop
    function selectInstances()
    #select instances for the virtual machines
    end
    function mapVirtualMachinesToInstances()
    #map virtual machines to the selected instances
    end
    function generateDeploymentPlan()
    #generate the final deployment plan
    end #end function
end #end code
```

4. Results

This study was conducted on distinct clouds utilizing different samples of instance types. The solution individuals (variables) and mapping of virtual machines to instances are illustrated in a hierarchical representation as shown in Figure 3. The instances with different specifications are available at various locations. The application being deployed on the cloud using DevOps required four nodes of virtual machines (VM) with varying specifications. [1]

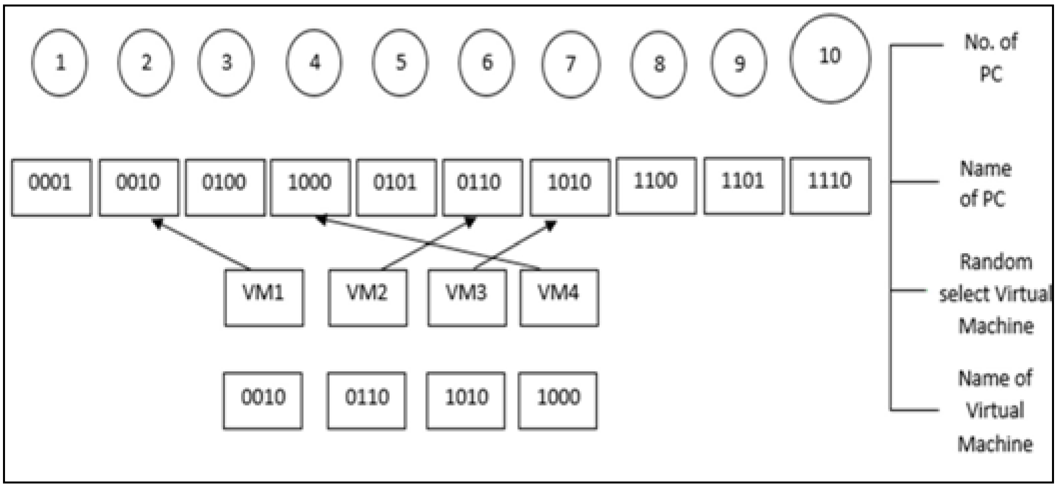

Figure 3: Random selections of V.M machine using algorithm

With the assistance of a automated framework, the challenges related with deployment and continuous integration (CI) in software development were essentially diminished in both time and exertion. The total capacity of virtual machines on three different clouds, AWS, Azure, and Google, is presented in Figure 4 and Table 1. The results in Table 1 indicate that there is a significant difference in capacity between clouds, while time does not show significance. ANOVA analysis in Table 2 shows that the p-value for cloud is 2.1 × 10−09 and for time is 1.7 × 10−10, indicating highly significant results for used space. The model used for predicting the used space shows a multiple R-squared value of 94%, indicating the best fit for used space. The hierarchical representation of solution individuals and mapping of virtual machines to instances is illustrated in Figure 3.

Table 1: ANOVA Table for Total Capacity

	Df	Sum Sq	Mean Sq	F Value	Pr(>F)
Cloud	2	1,040,000	520,000	3.90*10^30	<2 * 10^-16
Time	12	0	0	1.0 * 10^0	0.478
Residuals	24	0	0		

Residual standard error: 3.651 * 10^-13 on 24 degrees of freedom. Multiple R-squared: 1, Adjusted R-squared: F-statistic: 5.572 * 10^29 on 14 and 24 DF, p-value: < 2.2 * 10^-16. Interpretation: p-value = 2 * 10^-16 of cloud shows highly significant results for total. capacity while time is not significant.

The used space of instances on AWS, Azure, and Google clouds is presented in Figure 4, while Table 2 provides the ANOVA table outcomes for cloud and time. The analysis indicates that the p-value of 2.1 x 10^-09 for cloud and 1.7 x 10^-10 for time signify highly significant outcomes for used space.

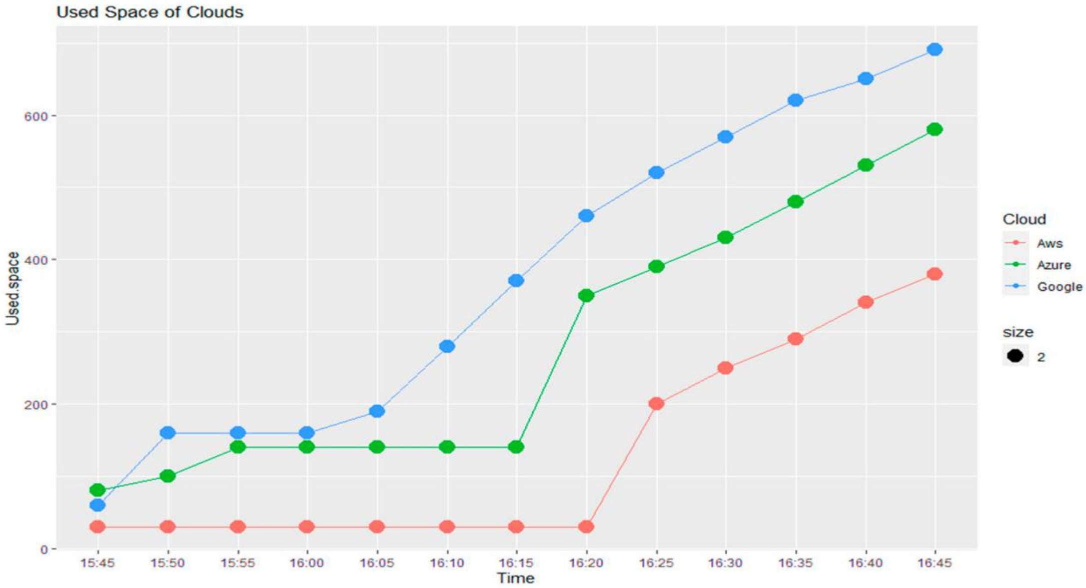

Figure 4: Used space of clouds

Table 2. ANOVA Table for used space

	Df	Sum Sq	Mean Sq	F Value	Pr (>F)
Cloud	2	397,492	198,746	5.13*10^1	2.17*10^-9
Time	12	1,123,159	93,597	2.41*10^01	1.72*10^-10
Residuals	24	93,041	3877		

Residual standard error: 62.26 on 24 degrees of freedom. Multiple R-squared: 0.9423, Adjusted R-squared: 0.9087.F-statistic: 28.02 on 14 and 24 DF, p-value: 1.805 * 10^-11. Interpretation: p-value = 2.1 * 10^-09 of cloud and 1.7 * 10^-10 for time shows highly significant results for used space. Multiple R2 = 94% show that prediction accuracy of this model is best fit for used space.

5. Summary, Conclusions and Future Work

The research has demonstrated that DevOps is a highly effective approach for deploying cloud applications and selecting resources based on value parameters such as cost, memory, and CPU capacity. This altogether affects the quality and speed of software product development, empowering software to be released rapidly and proficiently. Shifting towards cloud environments allows for greater agility and faster time-to-market, providing an optimal deployment environment for DevOps operations. When deploying multi-cloud applications, optimization parameters like price, CPU processing, memory allocation, and user node distance are crucial. The proposed algorithm can generate multiple deployment plans, with costs ranging from 5.96 to 16.10, CPU numbers between 8.43 and 11.99, memory sizes from 30.2 GB to 47.84 GB, user-node distances from 69,680,000 to 96,750,000, and inter-node distances from 69,680,000 to 96,750,000. Properly modifying the weights can achieve the desired trade-off between different parameters of the optimization problem. The suggested algorithmic solution has a cost value of 16.10, which is greater than the default option, when cost has no weight. The second time a weight of 0.2 is applied to cost, the cost value falls to 13.347. The cost value falls to 5.92% when the cost has the full weight of 1.0, resulting in a 60% cost savings compared to the baseline solution. Similar to how the cost parameter lowers, the user-node distance parameter decreases relative to the baseline solution by 9.26% in the case of equal weight and by 27.42% in the case of full weight. This paper provides a foundation for organizations to develop and deploy software on the cloud using DevOps techniques. The suggested algorithm can assist businesses in picking the optimal cloud for their DevOpsbased application deployment. In the future, the research will explore the effects of culture and environment on application deployment using DevOps and investigate DevOps as a cloud provider for application implementation.

References

[1] Leite, L.; Rocha, C.; Kon, F.; Milojicic, D.; Meirelles, P. A Survey of DevOps Concepts and Challenges. ACM Comput. Surv. 2019,52, 1–35.

[2] Jambunathan, B.; Kalpana, Y. Design of devops solution for managing multi cloud distributed environment. Int. J. Eng. Technol.2018, 7, 637–641.
3. Rafi, S.; Yu, W.; Akbar, M.A. RMDevOps: A road map for improvement in DevOps activities in context of software organi-zations.In Proceedings of the Evaluation and Assessment in Software Engineering, Trondheim, Norway, 15–17 April 2020; pp. 413–418.

[3] Zarour, M.; Alhammad, N.; Alenezi, M.; Alsarayrah, K. Devops Process Model Adoption in Saudi Arabia: An Empirical Study.Jordanian J. Comput. Inf. Technol. 2020, 6, 3.

[4] Akbar, M.A.; Rafi, S.; Alsanad, A.A.; Qadri, S.F.; Alsanad, A.; Alothaim, A. Toward Successful DevOps: A Decision-MakingFramework. IEEE Access 2022, 10, 51343–51362.

[5] Ellen, L.; Riungu-Kalliosaari, L.; Mäkinen, S.; Lwakatare, L.E.; Tiihonen, J.; Männistö, T. DevOps Adoption Benefits and Challenges inPractice: A Case Study; Springer: Berlin/Heidelberg, Germany, 2016; pp. 590–597.

[6] Cois, C.A.; Yankel, J.; Connell, A. Modern DevOps: Optimizing software development through effective system interactions.In Proceedings of the 2014 IEEE

International Professional Communication conference (IPCC), Pittsburgh, PA, USA,

[7] 13–15 October 2014.

[8] Soni, M. End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, ContinuousIntegration, Continuous Testing, and Continuous Delivery. In Proceedings of the 2015 IEEE International Conference on CloudComputing in Emerging Markets (CCEM), Bangalore, India, 25–27 November 2015; pp. 85–89.

[9] Barna, C.; Khazaei, H.; Fokaefs, M.; Litoiu, M. Delivering elastic containerized cloud applications to enable DevOps. InProceedings of the 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-ManagingSystems (SEAMS), Buenos Aires, Argentina, 22–23 May 2017; pp. 65–75.

[10] Guerriero, M.; Ciavotta, M.; Gibilisco, G.P.; Ardagna, D. SPACE4 Cloud: A DevOps Environment for Multi-cloud Applications.Short-Pap. In Proceedings of the 1st International Workshop on Quality-Aware DevOps, Bergamo, Italy, 28 May 2015; pp. 29–30.

[11] Kang, H.; Yoonhee, J.K.; Rahm, J. A SLA Driven VM Auto-Scaling Method in Hybrid Cloud Environment. In Proceedings ofthe 2013 15th Asia-Pacific Network Operations and Management Symposium (APNOMS), Hiroshima, Japan, 25–27 September2013; pp. 25–30.

[12] Li, Y.; Xia, Y. Auto-scaling web applications in hybrid cloud based on docker. In Proceedings of the 2016 5th InternationalConference on Computer Science and Network Technology (ICCSNT), Changchun, China, 10–11 December 2016; pp. 75–79.

[13] Mor´n, D.; Vaquero, L.M.; Gal´n, F.; Moran, D.; Galán, F. Elastically Ruling the Cloud: Specifying Application's Behavior inFederated Clouds. In Proceedings of the IEEE 4th International Conference on Cloud Computing,Washington, DC, USA, 4–9 July2011; pp. 89–96.

[14] Ghari, S. Devops for digital business: Optimizing the performance and economic efficiency of software products for digitalbusiness. In Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems, Pittsburgh,PA, USA, 18–23 May 2022; pp. 53–57.

[15] Tsilionis, K.; Sassenus, S.; Wautelet, Y. Determining the Benefits and Drawbacks of Agile (Scrum) and DevOps in Addressingthe Development Challenges of Cloud Applications. In Proceedings of the International Research & Innovation Forum, Athens,Greece, 15–17 April 2021; pp. 109–123.

[16] Akbar, M.A.; Smolander, K.; Mahmood, S.; Alsanad, A. Toward successful DevSecOps in software development organizations: Adecision-making framework. Inf. Softw. Technol. 2022, 147, 106894.

[17] Jaatun, M.G.; Cruzes, D.S.; Luna, J. DevOps for Better Software Security in the Cloud Invited Paper. In Proceedings of the 12thInternational Conference on Availability, Reliability and Security, Reggio Calabria, Italy, 28 August–1 September 2017; p. 69.

[18] Almeida, F.; Simões, J.; Lopes, S. Exploring the Benefits of Combining DevOps and Agile. Futur. Internet 2022, 14, 63.

[19] Arulkumar, V.; Lathamanju, R. Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method. ProcediaComput. Sci. 2019, 165, 399–405.

[20] Ferry, N.; Chauvel, F.; Song, H.; Rossini, A.; Lushpenko, M.; Solberg, A. CloudMF. ACM Trans. Internet Technol. 2018, 18, 1–24.

[21] Wettinger, J.; Breitenbücher, U.; Kopp, O.; Leymann, F. Streamlining DevOps automation for Cloud applications using TOSCA asstandardized metamodel. Futur. Gener. Comput. Syst. 2016, 56, 317–332.

[22] Shin, Y.; Williams, L. Can traditional fault prediction models be used for vulnerability prediction? Empir. Softw. Eng. 2011,18, 25–59.

[23] Blohowiak, A.; Basiri, A.; Hochstein, L.; Rosenthal, C. A Platform for Automating Chaos Experiments. In Proceedings of theIEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Ottawa, ON, Canada, 23–27 October2016; pp. 5–8.

[24] Akbar, M.A.; Mahmood, S.; Shafiq, M.; Alsanad, A.; Alsanad, A.A.A.; Gumaei, A. Identification and prioritization of DevOpssuccess factors using fuzzy-AHP approach. Soft Comput. 2020.

[25] Rafi, S.; Akbar, M.A.; Manzoor, A. DevOps Business Model: Work from Home Environment. In Proceedings of the InternationalConference on Evaluation and Assessment in Software Engineering, Gothenburg, Sweden, 13 June 2022; pp. 408–412.

[26] Venkateswaran, S.; Santonu, S. Architectural partitioning and deployment modeling on hybrid clouds. Softw. Pract. Exp. 2018,48, 345–365.

[27] Singh, V.; Peddoju, S.K. Container-based microservice architecture for cloud applications. In Proceedings of the 2017 InternationalConference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 847–852.

[28] Ghimire, R. Deploying Software in the Cloud with CI / CD Pipelines. 2020. Availableonline:https://www.theseus.fi/bitstream/handle/10024/345618/Thesis_Ramesh_Ghimire_1.pdf?sequence=2 (accessed on 27 October 2022).

[29] AWS- Amazon Web Services, Amazon CloudWatch Developer Guide API Version 2010-08-01 Amazon CloudWatch: DeveloperGuide. 2010. Available online: https://s3.cn-north-1.amazonaws.com.cn/aws-dam-prod/china/pdf/acw-dg.pdf (accessed on27 October 2022).

[30] Knott, M. Version Control with Git; O'Reilly Media: Sebastopol, CA, USA, 2014.

[31] Sullivan, B.O. Mercurial: The Definitive Guide Compiled from c3863298abc7. Availableonline:http://btn1x4.inf.unibayreuth.de/publications/dotor_buchmann/SCM/Mercurial/O%27Sullivan2009%20-%20Mercurial%20The%20defintive%20guide.pdf (accessedon 27 October 2022).

[32] Jakkula, V. Tutorial on Support Vector Machine (SVM). 2011. Available online: http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf (accessed on 27 October 2022).

[33] Uphill, T.; Arundel, J.; Khare, N.; Saito, H.; Lee, H.C.C.; Hsu, K.J.C. DevOps: Puppet, Docker, and

Kubernetes; Packt Publishing Ltd.:Birmingham, UK, 2017.

[34] Jaramillo, D.; Nguyen, D.V.; Smart, R. Leveraging microservices architecture by using Docker technology. In Proceedings of theSoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016.

[35] Bowes, J. Jenkins Continuous Build System Executive summary. 2012. Available online: https://docplayer.net/5686123-Jenkinscontinuous-build-system-jesse-bowes-csci-5828-spring-2012.html (accessed on 27 October 2022).

[36] Li, Z.; Zhang, Y.; Liu, Y. Towards a Full-Stack DevOps Environment (Platform-as-a-Service) for Cloud-Hosted Applications.Tsinghua Sci. Technol. 2017, 22, 1–9.

[37] Shiwani, S. Performance Analysis of IPv4 v / s IPv6 in Virtual Environment Using UBUNTU. In Proceedings of the Interna-tionalConference on Computer Communication and Networks, Valencia, Spain, 9–13 May 2011; pp. 72–76.

[38] Portnoy, J. Systems Monitoring with Prometheus and Grafana. Available online: https://flightaware.engineering/systemsmonitoring-with-prometheus-grafana/(accessed on 27 October 2022).

[39] Beaver, D.; Hutchison, S. Elasticsearch, Logstash, and Kibana (ELK). 2015. Available online: https://resources.sei.cmu.edu/asset_files/presentation/2015_017_001_431205.pdf (accessed on 27 October 2022).

[40] Padmanaban, S.; Khalili, M.; Nasab, M.A.; Zand, M.; Shamim, A.G.; Khan, B. Determination of Power Trans-formers Health IndexUsing Parameters Affecting the Transformer's Life. IETE J. Res. 2022.

[41] Aryal, R.G.; Altmann, J. Dynamic application deployment in federations of clouds and edge resources using a multiobjectiveoptimization AI algorithm. In Proceedings of the 2018 Third International Conference on Fog and Mobile Edge Computing(FMEC), Barcelona, Spain, 23–26 April 2018; pp. 147–154.

Author Profile

Sarthak Srivastava is a seasoned Senior DevOps Engineer with over five years of experience in the field. Currently affiliated with Visa Inc, a leading global financial services corporation, Sarthak has played a pivotal role in optimizing DevOps engineering practices within the fintech industry. With a Master of Science degree in Computer Science, Sarthak brings a strong academic foundation to his work. Sarthak's expertise lies in the intersection of DevOps, testing, and automation. Through his specialization in optimizing testing strategies and leveraging automation techniques, he has successfully streamlined the software development lifecycle, ensuring the delivery of high-quality solutions. His knowledge spans various domains within DevOps, including continuous integration and delivery, infrastructure automation, cloud computing, and security best practices.