Observability & Reliability of Microservices Application using DevOps in a Cloud-based Environment

Bhikesh Khute[1]
Dhwani Buddhadev[1]
Srivaramangai R[2]
Department of Information Technology, University of Mumbai, Mumbai, India

Abstract:- In today's extremely competitive commercial environment, organizations must deliver software products that meet customer requirements with the highest quality and reliability possible. After development, deploying the application using various methods or practice is critical. However, ensuring that the application runs smoothly and efficiently, adhering to the Software Level Agreement (SLA) of the project, is equally essential. Moreover, intelligent mitigation of issues without manual intervention can help minimize the workload for DevOps teams and reduce the likelihood of human errors. It also emphasizes the importance of handling code, infrastructure, and integration tools in post-deployment scenarios to maintain reliability and efficiency while operationalizing business projects using DevOps methodologies. This can lead to improved customer satisfaction, increased revenue, and reduced downtime. Customized DevOps methodologies for domains such as IT services, healthcare, and manufacturing can efficiently run information systems with positive business value.

Keywords:- *Devops; Cloud; Reliability Systems; Software.*

I. INTRODUCTION

Microservices architectures have increased popularity in recent years due to their ability to improve scalability and flexibility in application development. However, the distributed nature of microservices can also make them complex to manage and monitor, which can lead to reduced reliability and increased downtime. Exploring how DevOps practices such as continuous integration, delivery, and deployment, along with containerization and orchestration tools, helps to improve the observability and reliability of microservices applications in a cloud -based environment. Also discussing various monitoring techniques and tools that can be used to gain insights into the performance and behavior of microservices applications.

Furthermore, examined the role of automation in maintaining the reliability of microservices applications by automating tasks such as testing, deployment, and scaling. Also, how error budgeting can be used to balance reliability and innovation in microservices applications. Overall, this paper aims to provide insights into how DevOps practices and tools can enhance the observability and reliability of microservices applications in a cloud-based environment, helping organizations to achieve their business goals by delivering high-quality, reliable software at scale.

We will be discussing some key technologies further to know the topics in depth.

- *Cloud Computing*

When computer resources are delivered on demand, users can access servers, storage, databases, and applications online. Cloud computing allows flexibility, quicker innovation, scalability, and the freedom to select a collection of services based on our needs instead of managing actual data centres and servers. Private clouds, public clouds, hybrid clouds, and multi-clouds are some of the several types of cloud computing. Infrastructure-as-a-Service (IaaS), platforms-as-a-service (PaaS), and software-as-a-service (SaaS) are the three primary categories of cloud computing services.

A. DevOps

DevOps is a culture that is practised nowadays in most of the organizations as it has ability to deliver services at high velocity. Compared to organizations employing conventional software development and infrastructure management techniques, it enables superior infrastructure and application development, deployment, delivery, and maintenance at a quicker rate. Benefits of DevOps include automation, security, scalability, speed, rapid delivery, and strategies for configuration management.

B. CI/CD

Continuous development and deployment of new features are required. CI/CD tools available in market are very useful for designing, deployment and release of pipeline. It allows to design, control security aspects, approvals, notifications, audit etc. Gitlab has mostly yaml based configurations. Jenkins is an open-source tool having variety of plugins and Features which are built and updated regularly. Also, other few examples of CI/CD offerings are logic. Bamboo, TeamCity, Circle CI, Travis CI etc.

C. Microservices

Concept of Microservices is that an application or software can be deployed as multiple different modules and

ISSN No:-2456-2165

services. So that if any of the module or service goes down it won't affect the other modules of the same application. Using microservices helps applications to scale up and develop faster, easy to do enhancements separately for services, accelerates releasing of new features in market.

D. Node.js

JavaScript is used by the free, open-source server environment Node.js. It functions on a number of operating systems, including Windows, MacOS, Linux, and Unix. JavaScript code is executed outside of a web browser to create scalable network applications. ".js" is the extension for Node.js files. Web servers and networking utilities can be built using node.js and a group of "modules" that each handle a different basic functionality. Single-threaded, non-blocking, asynchronous programming, which is incredibly memory efficient, is used by Node.js code.

E. AWS Lambda

AWS Lambda runs code in response to events and automatically manages resources, it is a serverless compute service. Events may include changes or some actions that are performed on an application. It is used to extend AWS services to create backend services that operate at AWS to scale up, performance and security. Examples may include HTTPS requests coming via Amazon API Gateway, S3 bucket modifications, data update and structure modifications in Amazon DynamoDB, state transitions in AWS step functions.

F. Docker

Docker is an open platform which helps developers to build, share, run applications. Docker is a container that is used to host applications. It allows to run multiple containers. We need to make docker image run in an organised manner as it holds out application. Kubernetes and Docker Swarm are Container Orchestrator tools which can be used to scale and expose our application to the world.

G. Kubernetes

Google created the open-source container-orchestration tool known as Kubernetes, commonly referred to as K8s. Clusters of containerized applications are controlled and protect resources. useful for controlling application-running containers to prevent downtime. Another container must be used in the event that the first one fails. When a failed container fails to respond to a health check, Kubernetes automatically and effectively handles this switchover by restarting, replacing, and destroying the failed container. It is helpful for scaling, deployment, and all-around application management because it also keeps track of clusters and chooses where to launch containers based on the resources that are currently being used.

H. Monitoring

Monitoring is a practice to check, record and test particular services running and their availability. Using monitoring tool, we can do logging of the events and actions performed on the servers. We will be using Prometheus & Grafana tool to monitor of applications and their impact on the infrastructure. All the errors, warnings & bottlenecks will be detected and mitigated in the provided SLA.

II. BACKGROUND

All the research in this domain speaks about the problems in software life cycle management and how it can be enhanced by using best practices. Not only technology but skilled people is the current crunch in this domain. Not enough skilled people are present in the industry who can understand and develop strategies to mitigate the same. The papers discuss about characteristics of application such as performance, efficiency and quality of the system and how to enhance them. Also, it also throws some light on the lack of documentation while building applications & products & new methodologies to avoid the limitation. Developing a product or an application requires to follow a standard model which is recognized irrespective of the domains. They also speak about the no full utilization of the modern resources which can fine tune the efficiency of the system. Some of the strategies also play a key role in finding the scope and limitations of a system which is further discussed by these researchers in these research works.

III. RESEARCH WORK

Jayakody et al. [1] discusses the challenges and prioritization of challenges faced by organization while adopting DevOps. They have identifies & provides mitigating strategies in adopting DevOps in project development. Some of the challenges are consumption of more time, less expertise, no interesting tools available and lack of IT infrastructure. However, mitigation strategies include establishing communication platform procedure between software development and operations team, rearrange development of group to include people who have good experience with DevOps, improve knowledge about DevOps adoption through recent research findings. Pooja Batra et al. [2] explains about the performance & evaluation of the issue of the application/software while adopting devops framework. Productivity and Efficiency is considered as essential quality drivers to assess the performance of a project. DevOps provide more transparency to the customer while the traditional model lacks the same. The efforts taken in devops is far less as compared to the other models. The authors of Pulasthi Perera et al. [3] discuss how DevOps practises have affected software quality and how to effectively increase quality. The rationale for why big businesses have shifted to using DevOps is presented. The conceptual framework is split into two sections, each of which covers the elements required for implementing DevOps and their effects on the calibre of software. The research study's assumptions have been developed to confirm the relationship between elements such as practise, culture, automation, sharing, and measurement in devops. Information has been acquired through online surveys and interviews with DevOps professionals in the software development sector. After gathering and assessing the data, it was found that there is a link between the DevOps methodology and the calibre of software. DevOps practise will improve the quality of software. Additionally, there is a significant positive correlation between culture, automation, measurement, and sharing, which implies that putting automation into practise and sharing knowledge will raise software quality. The findings of this study can help businesses that use DevOps and quality engineering teams

ISSN No:-2456-2165

make decisions on how to enhance testing procedures. Hugo da Giao et al. [4] explore some of the difficulties that businesses and software engineers encounter when implementing and using DevOps. Model-driven engineering (MDE) has been used to address a variety of issues in computer science, including complexity, the need for greater system interoperability, aiding in the design process, and enhancing teamwork among members of various teams. The three components of the DevOps process are development, build management, and deployment management. The author's goal is to provide a framework that can produce the necessary configuration files or code based on the model for tools that are already in use or even tools that could be developed in the future, with the only requirement being to create the necessary mapping between the tools. Guoping Rong et al. [5] explains about the lack of documentation implied while building software products using DevOps approach. A new approach, DevDocOps is a continuous automated documentation aim to create documents parallelly with the working versions of the software in a very seamless manner. It reduces the time frame from one-two months to two days which improves customer(client) success. In their discussion of a case study, He Zhang et al. [6] assess the viability of using the CMMI models to direct process improvement for DevOps projects and pinpoint any potential shortcomings. It is explained how CMMI models can be used to evaluate the DevOps project's present processes and direct future improvements. To achieve the goal of this research, a case study that involves a real-world software project is incorporated. DevOps contains three interconnected aspects, i.e., capabilities, cultural enablers, and technological enablers. Project background, research-related Q&A, study scope, subjects, interviewing techniques, data gathering and consolidation, and study validity are all taken into account while designing a case study. Hui Kang et al. [7] discusses t the containers and microservices design which is the modern approach in today's world. The author criticises about the full utilization of the containers services that it offers. It investigated the difficulties and various approaches to managing services in containers using Open stack. Contrary to virtual machines, several containers can be set up on a single server to maximise resource usage and improve isolation while accessing memory and I/O devices. Active: In a container-based architecture, the active state is considerably more appropriate. Johannes Wettinger et al. [8] explains that DevOps when integrated with cloud provides a very affordable cost to build and run to the meet the customer requirements. In order to choose the right approach to do so, right decision making is taken with a help of knowledgebase. This shows how to utilize, deploy and operate cloud applications in a very organized and affordable fashion. It captures the knowledge from experts, crawlers and crowdsourcing. The knowledgebase contains implementation, cookbooks & modules. Johannes Wettinger et al. [9] reviews about the systematic and standard classification of artifacts and how it can be transformed towards Topology and Orchestration Specification for Cloud Applications (TOSCA). It enables seamless orchestration of artifacts to model and deploy. The artifacts are further classified into node-centric & environment-centric. manual selection of artifacts of different applications is done and further retrieval of additional artifacts to resolve dependencies. Further, TOSCA node and relationship types are generated based on artifacts and templates are created. In order to get a somewhat thorough understanding of what employers demand from DevOps roles in a New Zealand (NZ) setting, Waqar Hussain et al. [10] describe and present a picture of Knowledge areas, Skills and Capabilities (KSCs). The study's conclusions show that DevOps is becoming popular as a concept in which the duties of individual team members are increasingly divided between Development and Operations teams. This study will outline the research and practise in the DevOps field's ramifications and future directions, with potential applications to other small advanced economies. The findings raise awareness of the DevOps job market in New Zealand and assist graduates and job seekers in determining their employability in this field. Thomas F. Düllmann et al. [11] discusses about the continuous delivery pipelines which provides short and high-frequency commits and bringing it into production. The goal is to detect, diagnose and resolve dependability and security in the Continuous Deployment (CD) pipeline behaviour. To overcome the same, it uses devops practices such as canary release, Alpha and Beta (A/B) In their discussion of a case study, He Zhang et al. [6] assess the viability of using the CMMI models to direct process improvement for DevOps projects and pinpoint any potential shortcomings. It is explained how CMMI models can be used to evaluate the DevOps project's present processes and direct future improvements. To achieve the goal of this research, a case study that involves a real-world software project is incorporated. DevOps contains three interconnected aspects, i.e., capabilities, cultural enablers, and technological enablers. Project background, research-related Q&A, study scope, subjects, interviewing techniques, data gathering and consolidation, and study validity are all taken into account while designing a case study. Hui Kang et al. [7] discusses t the containers and microservices design which is the modern approach in today's world. The author criticises about the full utilization of the containers services that it offers. It investigated the difficulties and various approaches to managing services in containers using Open stack. Contrary to virtual machines, several containers can be set up on a single server to maximise resource usage and improve isolation while accessing memory and I/O devices. Active: In a container-based architecture, the active state is considerably more appropriate. Johannes Wettinger et al. [8] explains that DevOps when integrated with cloud provides a very affordable cost to build and run to the meet the customer requirements. In order to choose the right approach to do so, right decision making is taken with a help of knowledgebase. This shows how to utilize, deploy and operate cloud applications in a very organized and affordable fashion. It captures the knowledge from experts, crawlers and crowdsourcing. The knowledgebase contains implementation, cookbooks & modules. Johannes Wettinger et al. [9] reviews about the systematic and standard classification of artifacts and how it can be transformed towards Topology and Orchestration Specification for Cloud Applications (TOSCA). It enables seamless orchestration of artifacts to model and deploy. The artifacts are further classified into node-centric & environment-centric. manual selection of artifacts of different applications is done and further retrieval of additional artifacts to resolve dependencies. Further, TOSCA node and relationship types are generated based on artifacts and

ISSN No:-2456-2165

templates are created. In order to get a somewhat thorough understanding of what employers demand from DevOps roles in a New Zealand (NZ) setting, Waqar Hussain et al. [10] describe and present a picture of Knowledge areas, Skills and Capabilities (KSCs). The study's conclusions show that DevOps is becoming popular as a concept in which the duties of individual team members are increasingly divided between Development and Operations teams. This study will outline the research and practise in the DevOps field's ramifications and future directions, with potential applications to other small advanced economies. The findings raise awareness of the DevOps job market in New Zealand and assist graduates and job seekers in determining their employability in this field. Thomas F. Düllmann et al. [11] discusses about the continuous delivery pipelines which provides short and high frequency commits and bringing it into production. The goal is to detect, diagnose and resolve dependability and security in the Continuous Deployment (CD) pipeline behaviour. To overcome the same, it uses devops practices such as canary release, Alpha and Beta (A/B) testing and chaos engineering to the CD pipelines. It focuses on the security of the CD pipeline and uses Spoofing identity, tampering with data, Repudiation threats, Information disclosure, Denial of service and Elevation of privileges (STRIDE) threat modelling approach. Nikhil Govil et al. [12] discusses about some of the functional facets of agile methodologies and DevOps processes are acquainted in e-commerce application. The prime challenge mentioned is regarding updating of e-commerce application on frequent basis for which there is a need of continuous development and continuous integration which can be fulfilled with the help of devops. The project is on an e-commercial web application of jewellery firm in which products are listed online and made available to buy online or by visiting onsite. use of scrum framework is explained and how it supports development teams to work collectively. Development and maintenance of a product using devops tools is defined in detail. A report is shared which explains benefits of using devops and agile mechanism combined together helps in achieving employee engagement and customer satisfaction both leading to tremendous growth in IT industries business. Also, some of the major challenges of using agile and devops together are highlighted. Rajeev Kumar Gupta et al. [13] reviews successful implementation of DevOps in three different countries with timeboxed release strategies. Also, they have transformed from traditional scrum into a DevOps team. It helped them stabilizing processes and methods to an extent where they have released several products within a year. The quality process followed is unique in different geographical regions because following one QP makes the project cumbersome, complicated and block short release cycles. Mindset is the biggest blocker for a transforming team. Continuous learning is an essential part of learning cycle. Prashant Agrawal et al. [14] discuss the extension of DevOps processes and automation into public and/or private clouds as well as how to become more agile in software development and operations. The main objective is to comprehend how DevOps and Cloud cooperate to assist firms in achieving their transformational objectives. includes setting up a DevOps process on the cloud and using services. DevOps approaches applied for pilot projects with Cloud Computing prove to be a great success, as evidenced by the DevOps Life Cycle Model, DevOps Framework as specified collection of cascade-less phases, Process of Continuous Testing, Process of Continuous Deployment, and DevOps-as-a-service results. In his article, Manish Virmani [15] discusses all the DevOps-related factors that are pertinent to the various stages of the SDLC, with a particular emphasis on the business need and the switch from continuous integration to continuous delivery of software products. It is challenging to alter the same in order to satisfy the swift changes demanded by the client or market. Continuous Planning, Continuous Integration, Continuous Deployment, and Continuous Monitoring are the sequential steps to accomplish the same. Benefits include reliability, time savings, and the elimination of specific hardware. DevOps culture and practises present a number of difficulties, which Shoaib Khan et al. [16] analyse and address. Additionally, it explains how DevOps functions within an organisation, gives a thorough explanation of DevOps, and looks into the cultural difficulties that organisations encounter when implementing DevOps. The proposed study identifies ten significant obstacles to implementing the DevOps culture. The difficulties are further examined based on the different continents. The research revealed that the following significant difficulties are taken into account while implementing a DevOps culture: lack of cooperation and communication, knowledge and skill deficiencies. According to Azeem Akbar et al. [17], the implementation of DevOps practises is complicated, which inspires us to investigate the best practises that are crucial for the DevOps paradigm's success in software organisations. A total of 48 best practises were discovered as a consequence of the thorough literature review. The CAMS model's basic categories of culture, automation, measurement, and sharing (CAMS) were further matched to the best practises that had been found. In addition, a questionnaire survey study was done to collect the opinions of experts on the best practises that had been found. The findings of the questionnaire survey study showed that the best practises identified are consistent with actual practises. Finally, using fuzzy-AHP (Analytic Hierarchy Process), the investigated best practises were further prioritised according to their importance for DevOps practises. Results of the prioritisation reveal crucial best practises. Future work will include creating a readiness model that will help practitioners evaluate and enhance their DevOps implementation plans. With the help of the build pipeline concept, Mitesh Soni [18] presents a proof of concept for creating a successful framework for continuous integration, continuous testing, and continuous delivery that automates the compilation of source code, code analysis, test execution, packaging, infrastructure provisioning, deployment, and notifications. The insurance business is used as an example of how to better respond to changing market demands, has to launch new projects and services more quickly, and must support cutting-edge consumer engagement strategies. The issues facing the insurance business are discussed in this study, along with the many Proof of Concept (PoC) use cases that were implemented, the processes of artifacts from Continuous Integration to Continuous Deployment, the build pipeline for orchestration, end-to-end automation, and the advantages realized after deployment. Monitoring, security, quantifying the benefits of end-to-end automation for continuous delivery, and integrating open-source Application Life cycle Management tools (ALM Tools) are the main areas that need to be the focus of future effort. Giuseppe Vergori et al. [19] discusses the dimensions of the devops organisational scenarios that can be addressed with performance engineering. This is done with in a real-life industrial DevOps scenario. The key consideration in the DevOps workflow is – Actors, Actions, Tools, Variables and Measurable Quantities. A DevOps environment can be used to ensure compliance and build trust, according to John R. Michener et al. [20]. demonstrated how a lack of strict compliance controls can result in errors and a lack of confidence in organisations like Microsoft and Jupiter Networks. Compliance controls for operations are covered in detail separately from compliance controls for development. This study comes to the conclusion that as customer expectations rise and businesses need to continually match those demands, DevOps as a methodology will flourish. While poorly executed DevOps implementations are currently incompatible with compliance environments, hybrid environments can be set up that offer many of the benefits of the DevOps environment while still upholding the crucial security controls, rigour, and trust needed by compliance environments.

IV. CONCLUSION

This paper discusses the observability and reliability of microservices applications in a cloud- based environment using customized DevOps methodologies. It emphasizes the need for deploying applications using suitable methods and adhering to Software Level Agreements to ensure reliability and efficiency. Key technologies such as cloud computing, DevOps, CI/CD, microservices, Node.js, AWS Lambda, Docker, Kubernetes, and Monitoring are discussed. Additionally, the paper addresses performance bottlenecks in traditional web applications and presents a methodology involving analyzing existing system implementations, implementing cost- effective solutions, performing chaos engineering, and optimizing performance based on SLAs. The results of this study help improve the efficiency of legacy systems and can serve as a case study for similar projects around the world. Furthermore, the paper concludes by discussing the problems in software lifecycle management and how best practices can enhance it.

REFERENCES

[1]. Jayakody, J. A. V. M. K., and W. M. J. I. Wijayanayake. "Challenges for adopting DevOps in information technology projects." In 2021 International Research Conference on Smart Computing and Systems Engineering (SCSE), vol. 4, pp. 203-210. IEEE, 2021.

[2]. Batra, Pooja, and Aman Jatain. "Measurement Based Performance Evaluation of DevOps." In 2020 International Conference on Computational Performance Evaluation (ComPE), pp. 757- 760. IEEE, 2020.

[3]. Mohammad, Sikender Mohsienuddin. "Improve software quality through practicing devops automation." Sikender Mohsienuddin Mohammad," IMPROVE SOFTWARE QUALITY THROUGH PRACTICING DEVOPS AUTOMATION", International Journal of Creative Research Thoughts (IJCRT), ISSN (2018): 2320-2882.

[4]. Giao, Hugo. "A model-driven approach for DevOps." In 2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 1-3. IEEE, 2022.

[5]. Rong, Guoping, Zefeng Jin, He Zhang, Youwen Zhang, Wenhua Ye, and Dong Shao. "DevDocOps: Towards automated documentation for DevOps." In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp. 243-252. IEEE, 2019.

[6]. Rong, Guoping, He Zhang, and Dong Shao. "CMMI guided process improvement for DevOps projects: an exploratory case study." In Proceedings of the International Conference on Software and Systems Process, pp. 76-85. 2016.

[7]. Kang, Hui, Michael Le, and Shu Tao. "Container and microservice driven design for cloud infrastructure devops." In 2016 IEEE International Conference on Cloud Engineering (IC2E), pp. 202-211. IEEE, 2016.

[8]. Wettinger, Johannes, Vasilios Andrikopoulos, and Frank Leymann. "Automated capturing and systematic usage of devops knowledge for cloud applications." In 2015 IEEE International Conference on Cloud Engineering, pp. 60-65. IEEE, 2015.

[9]. Wettinger, Johannes, Uwe Breitenbücher, and Frank Leymann. "Standards-based DevOps automation and integration using TOSCA." In 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 59-68. IEEE, 2014.

[10]. Hussain, Waqar, Tony Clear, and Stephen MacDonell. "Emerging trends for global DevOps: a New Zealand perspective." In 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), pp. 21-30. IEEE, 2017.

[11]. Düllmann, Thomas F., Christina Paule, and André van Hoorn. "Exploiting devops practices for dependable and secure continuous delivery pipelines." In 2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE), pp. 27-30. IEEE, 2018.

[12]. Govil, Nikhil, Mayank Saurakhia, Pradyumna Agnihotri, Sachin Shukla, and Shivam Agarwal. "Analyzing the behaviour of applying agile methodologies & DevOps culture in e- Commerce web application." In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), pp. 899-902. IEEE, 2020.

[13]. Gupta, Rajeev Kumar, Mekanathan Venkatachalapathy, and Ferose Khan Jeberla. "Challenges in adopting continuous delivery and DevOps in a globally distributed product team: a case study of a healthcare organization." In 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), pp. 30-34. IEEE, 2019.

[14]. Agrawal, Prashant, and Neelam Rawat. "Devops, a new approach to cloud development & testing." In 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), vol. 1, pp. 1-4. IEEE, 2019.

[15]. Virmani, Manish. "Understanding DevOps & bridging the gap from continuous integration to continuous delivery." In Fifth international conference on the

ISSN No:-2456-2165

innovative computing technology (intech 2015), pp. 78-82. IEEE, 2015.

[16]. Khan, Muhammad Shoaib, Abudul Wahid Khan, Faheem Khan, Muhammad Adnan Khan, and Taeg Keun Whangbo. "Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review." IEEE Access 10 (2022): 14339-14349.

[17]. Khan, Muhammad Shoaib, Abudul Wahid Khan, Faheem Khan, Muhammad Adnan Khan, and Taeg Keun Whangbo. "Critical Challenges to Adopt DevOps Culture in Software Organizations: A Systematic Review." IEEE Access 10 (2022): 14339-14349.

[18]. Soni, Mitesh. "End to end automation on cloud with build pipeline: the case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery." In 2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 85-89. IEEE, 2015.

[19]. Vergori, Giuseppe, Damian A. Tamburri, Diego Perez-Palacin, and Raffaela Mirandola. "Devops performance engineering: A quasi-ethnographical study." In Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion, pp. 127-132. 2017.

[20]. Michener, John R., and Aaron T. Clager. "Mitigating an oxymoron: compliance in a DevOps environments." In 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), vol. 1, pp. 396-398. IEEE, 2016.