National College of Ireland

Title

Integration of Security Vulnerability Tools and Kubernetes Deployment to Obtain an Enhanced CI/CD Pipeline for a Blockchain Based Decentralized Application (DApp)

MSc Research Project
MSc Cloud Computing

Rachana Poonacha
Student ID: 22217029

School of Computing
National College of Ireland

Supervisor: Jitendra Kumar Sharma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Rachana Kottangada Poonacha ……………………………………………………………………………………………………

Student ID: 22217029 ……………………………………………………………………………………………………..……

Programme: MSc Cloud Computing …………………………………………………………… **Year:** 2023-24 ………………………..

Module: MSCCLOUD Research Project ………………………………………………………………………………………….………

Supervisor: Jitendra Kumar Sharma ………………………………………………………………………………………….………

Submission Due Date: 16-09-2024 ………………………………………………………………………………………….………

Project Title: Integration of Security Vulnerability Tools and Kubernetes Deployment to Obtain an Enhanced CI/CD Pipeline for A Blockchain-based Decentralized Application (DApp) ……………………………………………………………………………………………….………

Word Count: 13043 ……………………………………… **Page Count** 37 ………………………………………………….……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: K P Rachana ……………………………………………………………………………………………………

Date: 16-09-2024 ……………………………………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

INTEGRATION OF SECURITY VULNERABILITY TOOLS AND KUBERNETES DEPLOYMENT TO OBTAIN AN ENHANCED CI/CD PIPELINE FOR A BLOCKCHAIN BASED DECENTRALIZED APPLICATION (DApp)

Rachana Kottangada Poonacha
Student ID: 22217029

Abstract

Blockchain-based technology has revolutionized several industry sectors due to its ability to enable trustless systems by giving greater control over the data to the users and reducing the dependency on third party centralized authorities. However, there is a significant gap identified in the development lifecycle of the applications that are supported by Blockchain. One such application is called the Decentralized Application or the DApp that runs on Web3. This research fundamentally explores the integration of one of the modern and latest software engineering technique available in the market called "Devops" into the development lifecycle of Blockchain-based Decentralized Application (DApp) and showcases significant advancement over the existing literature in terms of security, scalability, availability and integrity. The application developed for the purpose of this research is called "The Yoga Studio" DApp. The implemented CI/CD pipeline automated various stages of the development process through "Github Actions" enabling Continuous Integration and Continuous Deployment. The stages included in the CI/CD pipeline are: Source stage, Lint stage using ESLint, SonarCloud scan using SonarCloud, Build using Docker, Deploy stage using AWS Elastic Kubernetes Service. The results from the implementation of the proposed solution have provided insights into how the integration of Devops principles has improved application development process of DApps. The factors that evaluate the conducted experiment includes: The speed of deployment that is accelerated due to end-to-end automation, enhanced security due to vulnerability scan at the early stage of the development process, versioning of the code to keep track of latest changes, improvement in the overall quality of the code due to the integration of linting tools, and deployment of the application into AWS EKS by leveraging the benefits offered by the containerization and orchestration techniques.

***Keywords:* Blockchain, DApp, Devops, security, availability, containerization, orchestration, AWS EKS, Docker, ESLint, SonarCloud**

1 Introduction

A key area of research in contemporary software engineering is the integration of Devops practices into the Blockchain application development lifecycle. Devops framework, with its emphasis on continuous integration, continuous delivery, continuous deployment, collaboration, continuous communication, visibility, automation and transparency, has the ability to significantly enhance the development process of Blockchain-based applications such as a Decentralized Application (DApp) or Hyperledger Fabric application. On the other hand, Blockchain technology is set to transform the way current business is conducted due to its advantages of being immutable, secure, decentralized peer-to-peer networks tagged with distributed consensus, smart contracts and distributed ledger mechanisms. However, these same benefits introduce a number of additional factors to be taken into account for Blockchain-based applications that contributes to greater challenges in testing and deployments. Organisations needs to address this plethora of challenges in 3 major areas: process, culture and tools. On the process front, Blockchain technology is a new idea without mature design guidelines and principles at the management level, with uncertainty and lack of knowledge of the Blockchain development stack and environment. On the culture perspective, Blockchain's success depends on the backing at executive level, adequate funding and sufficient resources for efficient deployments. Regardless of whether you are trying to build a Blockchain network with nodes outside the organisation or within a single business, obtaining consensus on the Blockchain technology procedures is currently an issue. It is an added complexity due to lack of standard tools and processes in place. Another major obstacle for Blockchain is the absence of development tools. The ones that are available currently are mostly unreliable. Hence, the Blockchain ecosystem requires a high-quality Integrated Development Environment (IDE) with suitable plug-ins and linters, compilers and building tools, tools for debugging, testing methodologies, proper documentation, security auditing tools, monitoring/logging tools and deployment tools. As a result, Devops practices can provide the required efficiency, quality and consistency that is exactly required for organisations looking to embrace the cutting-edge Blockchain technology.

Furthermore, the Devops approach not only incorporates code integration, testing, and deployments, but also encompasses critical steps to elevate "Security" to the top of the Devops pipeline, resulting in a more refined SDLC process known as "DevOpsSec" which also forms the base for this research. Integrating security in the early stages of the development lifecycle (called as the shift-left philosophy) is essential to maximize productivity and efficiency. A system that supports innovation and that supports a team of individuals collectively working towards fixing issues is required to implement Blockchain successfully.

Having said that, Blockchain technology, has revolutionized several industry sectors ever since its inception in 1991. Its potential use cases spans across numerous industries such as healthcare, retail, financial services, marketing and advertising. The technology offers distributed ledgers that gets automatically updated, allowing for decentralized and immutable records of transactions that are exposed publicly for all the users on the network. In the banking industry, Blockchain technology has been offering highest level of security with minimum transaction fees due to the peer-to-peer mechanism of payments. It enables faster settlements by eliminating the risk of exchange rates in cross-currency transactions. Additionally, it can be utilized to streamline processes by implementing smart contracts, compliance automation, simplification of operations by tracing credit letters and bank guarantees etc. Cyberattacks, were earlier posing enormous number of risks to the society and organisations started finding ways to develop solutions to protect data against manipulation and misuse. However, with the development of Blockchain technology, it has now become relatively easier to identify malicious attacks as the Blockchain operates in a peer-to-peer network where data is non-erasable and encrypted using cryptographic algorithms. In the field of Supply chain management, there has always been irregularities between departments due to lack of coordination, unreliable environments and service redundancy. Following the advent of Blockchain, tracking of products became extremely simple and anybody could validate the authenticity of the transactions as they are recorded into the Blockchain. In healthcare systems, patient health records were retrieved by connecting with a specific hospital which resulted in unnecessary delays and the risk of data corruption increased as they were stored locally in physical memory. But, with the increased popularity of Blockchain technology, the medical records were electronically saved, retrieved quickly without third-party intervention, improved processing of payments and eliminated the risk of data corruption as each block of data are linked together in the Blockchain. In the field of marketing, Blockchain plays an important role in improving the security and transparency around customer data sharing, either between two companies or between a company and a customer. The government sector has also greatly benefited from the Blockchain technology in terms of casting votes. Once a vote is cast, the details are entered into the public ledger and is cannot be altered or removed. Due to the range of industry sectors benefiting from Blockchain technology, it is extremely important to secure and securely deploy the Blockchain applications with rigorous measures. This process should ideally begin at the early stages of development by aligning with the best possible software practices such as integration of Devops. Given the lack of research done in this area, exploring how Devops can support Blockchain application development is a valuable and crucial topic for consideration.

The significance of the research question has been emphasized in the “Related Work (Literature Review)” section that follows by carefully reviewing a considerable number of research papers and examining each one thoroughly. It has been observed that there is a lack of standard SDLC practices being implemented for Blockchain related projects and hence the primary goal of this research is to close the gap between Blockchain technology and Devops methodology. As mentioned earlier, Devops will help in streamlining the Blockchain development process by improving the collaboration between different teams, rapid iterations, faster code release cycles, integration of static and dynamic code analysis tools, security mechanisms etc thereby enhancing the overall efficiency. The Continuous Integration (CI) and Continuous Delivery (CD) pipelines will help in accomplishing the process starting from code integration to deployment of the application into staging or production environments, thereby saving time by automating the repetitive tasks. Another important part of this research is to evaluate the significance of integrating code and security analysis tools after the Build stage in the Devops pipeline to demonstrate the security issues encountered during the early stages of development cycle so that security related threats and issues can be resolved before deploying the application. In addition to this, containerization and orchestration techniques have become a compute method of choice in modern software development and cloud-native application architectures as they are more resource-efficient and portable as compared to Virtual Machines. Application of these techniques into traditional applications have proved greater benefits. Hence, as part of this research, an effort has been made to containerize the Blockchain specific DApp using tools like Docker and

Kubernetes. This aims to analyse the capabilities these tools offer and evaluate their effectiveness for Blockchain applications, given the vast number of differences between Blockchain applications and traditional applications. Hence, a decentralized application called “The Yoga Studio App” (DApp) has been developed for demonstration purposes, with the intention of applying Devops practices to evaluate the operational efficiency and enhance the deployment process.

Having said that, the following research question serves as the study’s roadmap: “Can the integration of advanced security mechanisms for vulnerability scanning triggered at the release build time of a CI/CD pipeline and orchestration technology using Kubernetes enhance the resilience, integrity, scalability and availability of Blockchain based applications such as DApps where the results of the scan can be recorded and evaluated for security threats and issues at the early stages of the Blockchain Software Development Life Cycle?”.

The report comprises of the following sections: Section 2 provides a detailed study of the “Related works” (Literature Review) in the areas pertaining to Devops integration for Blockchain applications, security challenges in Blockchain based applications followed by the evaluation of containerization and orchestration techniques for Blockchain application deployments. Section 3 illustrates the “Research methodology” outlining the overall process and research procedures used to conduct the research. Section 4 outlines the “Design specifications” presenting the underlying technical frameworks and architecture that is supporting the solution. Section 5 illustrates the detailed “Implementation” steps describing how the solution was built and executed. Section 6 pertains to the “Evaluation” phase describing the primary findings and results from the overall research and providing a comprehensive analysis of the implemented solution. Section 7 summarizes the “Conclusions derived and Future work” with detailed description of the overall outcomes of the study and future objectives.

2 Related Work

The “Related Work” section examines a number of research papers in the field of “Devops for Blockchain application development” and investigates the range of suggested approaches that have been discovered thus far for integrating the two paradigms. The primary goal of the Literature review is to critically identify and evaluate any gaps and opportunities that could improve the integration of these two frameworks. Both the domains have independently contributed towards the transformation of how a particular software is built, maintained and delivered and hence there is a growing demand to explore how well they can work together to deal with challenges pertaining to automation, security and scalability. By thoroughly examining the existing state of the art in “Devops for Blockchain”, the literature contributes to a comprehensive and effective analysis of Devops pipeline for Blockchain application development. The review performed is broken down into 3 sub-sections which are “2.1: Devops techniques in Blockchain: a detailed analysis of related literature”, “2.2: Security related issues pertaining to Blockchain applications such as DApps”, and “2.3: Incorporating Orchestration and containerization tools (Docker and Kubernetes) into the Devops pipeline (CI/CD)”.

2.1 Devops techniques in Blockchain: a detailed analysis of related literature

This section delves into the various research undertakings on Devops practices applied in Blockchain, and to highlight the noticeable gap in the existing literature in this field. While the existence of Devops methodology has revolutionized software development practices, its application within the Blockchain domain remains unexplored. In reviewing the current state of the research, I have found that key Devops techniques such as continuous integration, continuous delivery, continuous deployment, continuous testing etc. have not yet been analysed or applied on Blockchain domain. Hence, an effort has been made to shed light on the present body of research, addressing the new challenges and opportunities to contribute to the field and to begin with the practical implementation so as to validate the results.

Maximilian Wohrer and Uwe Zdun (2021) has focused on the application of Devops principles in the field of software development for Blockchain, specifically targeting those Blockchain platforms that are utilizing Smart Contracts which are part of the Ethereum network. The Literature Review highlighted in the paper discusses about the lack of standard and structured approach for SDLC process of Blockchain-based applications. Researchers have applied techniques such as “Ground Theory” procedures to extract information related to the best Devops practices for Blockchain based applications, and has indicated that there are no advanced tools and practices yet in the space of Blockchain. They have also indicated that there is a potential scope for improvement in the domain to facilitate updates to smart contracts in a more seamless manner. Overall, as part of the future work, researchers have highlighted for integration of more automated and flexible deployments for Blockchain-based software development. As a result, an effort has been made to develop an end-to-end pipeline for the dApp called “The Yoga Studio” where continuous integration and continuous deployment tools such as “Github Actions” and “Kubernetes” have been integrated into the pipeline.

Sandip Bankar and Deven Shah (2021) emphasizes the importance of Blockchain framework for improving the Devops process in itself using the core properties of Blockchain technology such as decentralization, transparency, immutability and so on in order to secure the development environment and manage the project artifacts. The research highlights how Blockchain can transform the Devops process by enhancing the collaboration among stakeholders, reducing the redundancies in the development process and by promoting auditability. Researchers have utilized a permissioned Blockchain network such as Hyperledger Fabric as the main component of the framework proposed along with the integration of a decentralized database called the “Inter Planetary File System (IPFS)” to maintain project artifacts securely. Researchers have also suggested a robust CI/CD pipeline for improving the development process. As part of the future work, researchers have suggested the application of the framework across different Blockchain platforms in order to evaluate the scalability and flexibility of the environments. As a result, an effort has been made to implement the Devops process into the software development practice of a Decentralized Application (DApp) called the “The Yoga Studio” and the findings have been explained in detail to illustrate the advantages obtained in terms of deployments, compliance and overall efficiency.

M. J. H. Faruk, S. Subramanian, H. Shahriar, M. Valero, X. Li and M. Tasnim. (2022), in their paper, conducted a thorough investigation to determine the feasibility of combining software engineering techniques such as Agile and Devops methodologies for Blockchain-based application development. As the Blockchain technology is inherently complex due its core nature of being immutable combined with cryptographic mechanisms and consensus principles, there are certain challenges involved which prevents the smooth integration of Devops techniques to Blockchain based application development. Another important factor highlighted in this research paper is the aspect of “security” that requires careful consideration while integrating CI/CD pipeline for Blockchain development so that data breaches can be avoided. Furthermore, the study emphasizes the difficulties involved in the deployment of Blockchain applications due to the presence of sensitive user information, transaction details and so on. The main conclusion of the study indicates that the implementation of a successful Devops process for Blockchain domain is limited by the issues with respect to scalability. Hence, the implemented design and solution resolves the issue of scalability by considering the deployment of the DApp into container and orchestration tools such as Docker and Kubernetes which will help in enhancing the overall Devops process in terms of security, scalability and integrity.

A. Reyes, M. Jimeno, R. Villanueva-Polanco (2023) has contributed to the existing literature by discussing the significance of a structured approach that is required for the development of Ethereum supported smart contracts through the integration of Devops methodology. Authors have suggested the integration of Devops tools especially for continuous testing of the code for smart contracts along with source code management tools (SCMs), orchestration tools, cloud-based tools and so on. They have also highlighted the risks associated with the creation of Smart Contracts as they are subjected to vulnerabilities and bugs. Therefore, a comprehensive solution that can include security factor as a key component in the pipeline has been suggested to improve the overall efficiency of Smart Contracts development. However, in the paper, there has been no mention about a practical implementation of a robust end-to-end automated pipeline to improve the quality of the software development process for Blockchain enabled DApps. This aligns with the objective of the proposed research question and a suitable solution for the same has been implemented. The implemented solution deals with stringent security policies and measures introduced at the “Build” stage of the CI/CD pipeline using tools like Github actions, SonarCloud, Eslint, Docker and Kubernetes.

M. Shoaib Farooq and U. Ali (2023) make a compelling argument for combining Devops practices and Blockchain technology to create an efficient software development system. In this regard, a 6-layered framework has been proposed by the authors that defines the integration of Blockchain technology with distributed Devops methods. Some of the technology stack introduced as part of the implementation would be the use of permissioned Blockchain platform called the Hyperledger Fabric, decentralized storage system such as IPFS, smart contracts for automation, couch DB which is part of the Hyperledger network, Ethereum platform and so on. The benefits of the framework proposed is in the direction of enhancing the security, transparency, efficiency, automation etc. of the Devops based system. While their primary focus is improving the practices of Devops using Blockchain, their research offers important insights into how Devops methods might help in overcoming the challenges associated with the Blockchain-based applications specifically with respect to security. Hence, this research contributes to the existing body of literature by introducing “DevOpsSec” as an important factor into the CI/CD pipeline for “The Yoga Studio” DApp demonstrating different stages and evaluation criteria to make the overall system more secure.

K. Duan, J. M. Caballero and X. Jing. (2023) discusses the implementation of Scrum-based agile development practice and the Devops model to enhance the software engineering methods for Blockchain-based applications. Along with conveying the importance of CI/CD pipelines for Blockchain automated releases, the authors have also suggested the significance of a rigorous testing pipelines in place for conducting regression testing, smoke testing, performance testing and integration testing. Additionally, a conceptual framework called the Rivtower Inter-Enterprise Trust Automation (CITA) has been introduced outlining the security features such as the importance of decentralization for authentication, fine-grained access tokens and access control, Role-based Access Control (RBAC) and so on. However, the study does not exclusively mention the integration of such security mechanisms into the Devops pipeline to identify security related issues and security hotspots. Hence, in order to raise the standards of the DApp development process, an attempt has been made through this research to incorporate security tools such as SonarCloud at the build stage of the Devops pipeline to resolve security related concerns before the deployment of the application.

N. Sanchez-Gomez, J. Torres-Valderrama, J. A. García-García, J. J. Gutiérrez and M. J. Escalona (2020) emphasizes the critical importance of a systematic software design mechanism in place for the development and deployment of smart contracts. The study identifies the important shortcomings in the currently implemented approaches for analysing and verifying smart contracts, particularly during the “requirements gathering” and “testing” phases. Authors have promoted the adoption of a Devops framework that will allow for the collaboration of various teams and to streamline the overall development workflow, thereby facilitating continuous integration and continuous deployment of smart contracts into the Blockchain. It also mentions on the importance of identifying security vulnerabilities and bugs in the smart contracts code before being deployed into the Blockchain. Therefore, as the paper talks about the integration of Devops methodology as part of the future work for the deployment of smart contracts, an attempt has been made to accommodate the Devops framework by creating different stages in the pipeline such as build, security, and deploy to improve security assessments and automated deployments using tools such as Github actions, SonarCloud and Kubernetes.

2.2 Security issues pertaining to Blockchain based applications such as DApps

This section explores the existing literature on the various security related issues with Blockchain-based DApps and arrives at a conclusion recommending to introduce “Security” stage early in the development process. By doing so, security related data breaches and vulnerabilities can be mitigated and resolved before deploying the application to staging or production environments.

Hongsong Chen, Xietian Luo, Lei Shi, Yongrui Cao, Yongpeng Zhang (2023), in their paper, has highlighted the different security challenges associated with Blockchain-based applications. They have developed an entire framework of security architecture to address the security vulnerabilities in “Smart Contracts” such as DDoS attacks, trojan attacks, authorization and authentication attacks, weak passwords, issues in code and so on. Although the study implies the incorporation of Blockchain based security mechanisms such as zero-knowledge-proofs, consensus algorithms and hashed storage, it does not mention the need to introduce the latest software techniques like Devops to enhance the security of the Blockchain-based DApps. Therefore, as part of the research, an effort has been made to implement “Security” before the build stage of the CI/CD pipeline to identify bugs and security vulnerabilities in the Decentralized application called “The Yoga Studio”.

Noama Fatima Samreen, Manar H. Alalfi (2023), has provided a comprehensive analysis of the security, maintainability and complexity involved in Ethereum-based Blockchain applications. They begin by discussing the evolution of Blockchain technology and the relevance of “Smart Contracts” based DApps in the present world. Authors have investigated the current trends in Dpps development process and analysed the underlying “Smart Contracts” for security vulnerabilities. The tools used as part of the DApp development are Ganache, Truffle, Remix and Drizzle. Key findings from their research reveal that the critical security related issues in “Smart Contracts” are mostly related to Denial-of-Service (DoS) attacks and integer underflow and overflow, highlighting the need for a robust security mechanism in place during the development process. Therefore, an effort has been made to create a CI/CD pipeline integrating “SonarCloud” security vulnerabilities detecting tool to enhance the security features of the DApp.

M. R. Islam, M. M. Rahman, M. Mahmud, M. A. Rahman, in their 2021 work have discussed various security concerns with respect to Blockchain technology. Some of the Blockchain related security issues highlighted in the paper includes the 51% attack issue where the transactions are altered by an attacker, forking issues where the Blockchain network is broken, eclipse issue where individual users are targeted rather than the entire network and so on. Some of the solutions proposed in this paper for these issues include the investigation of the double-spending problem which poses a serious concern for Blockchain systems, potential solutions for bugs identified in the network or code and measures to protect sensitive data. Devops, being the methodology for modern software development process, can help building a robust and resilient system where such security issues can be identified and resolved much earlier in the development process. This brings us to the identified research question and in order to validate the same, a Devops pipeline has been configured with “Security” stage to scan for security issues using SonarCloud. The results of the experiment have been discussed in the following sections.

S. M. Idrees, M. Nowostawski, R. Jameel and A.K. Mourya (2021), provides an executive summary of the existing research in the filed of security for Blockchain powered applications. The authors have tried to analyse the various security threats that the Blockchain applications might get subjected to in case of improper security system in place. They have also emphasized the need to have a secure system considering how the Blockchain DApps have revolutionised several industry sectors such as healthcare, government, banking, supply chain management, retail and so on. The paper mentions about the peer-to-peer network architecture of Blockchain applications being the reason for the system to be easily susceptible to unauthorized accesses and data breaches. Authors have conceptually suggested a few solutions to mitigate these kind of security issues. However, there is no mention about a standard software development practice such as Devops that can be put in place to strength the security of the Blockchain system. An effort has been made to apply the principles of Devops into the Blockchain development by configuring CI/CD pipelines that includes a “Security” stage.

P. Zheng, Z. Jiang, J. Wu and Z. Zheng (2023), has conducted a survey that was aimed to provide a comprehensive overview of various challenges associated with DApps that included risks associated with economic policies, performance and security risks. The paper emphasizes on the vulnerabilities present in the multi-layered Blockchain architecture including the smart contracts, web and other components. It brings attention to the financial losses that comes along with DApps if the system is not secured. Some of the solutions recommended by the authors to mitigate these risks would be the use of suitable and safe programming languages and usage of formal verification and validation methods for smart contracts. To further improve DApp security, authors have reiterated the importance of having advanced vulnerability detection tools and trusted application development practice in place. As a result, my research underscores the importance of integrating Devops and Blockchain development to secure the Blockchain applications from potential security breaches.

W. Fan, H. J. Hong, X. Zhou and S. Y. Chang (2021) have emphasized the need for a standardized and structured approach towards securing the Decentralized applications. The developed framework is called as the “Generic Blockchain Framework” which can be applied to any kind of Blockchain-based DApps across all industry sectors and can be considered as the starting point for developing secure DApps. However, the paper does not explicitly address how the developed framework can be incorporated to increase the security of the DApps. As we already know, the software development methodology plays an important role in securing the Blockchain-based DApps as they provide guidance and structured approaches for building, testing, planning and maintaining the software, which helps the development teams to enhance the quality of the application. In this regard, the research aims to bridge the gap between one such software development strategy called the Devops with DApps, illustrating through the implementation how connecting these two paradigms has improved the security of the application early in the development process.

R. Hegadi, S. S. K. Akella, K. Reddy and P. Kumar C (2023), have thoroughly examined the security vulnerabilities in the Web3 ecosystem specifically with respect to Ethereum based “Smart Contracts”. “Smart Contracts” are programs enabling user interaction from outside the Blockchain network and has become widely popular. Therefore, the security of these programs becomes increasingly important. Some of the vulnerabilities mentioned in the paper that are commonly encountered in “Smart Contracts” includes integer overflow, improper access controls, continuous re-execution of functions, unauthorized actions like reusing the digital signatures and so on. Authors have proposed a strong framework in order to mitigate these security risks by combining secure coding practices with end-to-end testing along with testing tools like Solidity scan and Slither that help in scanning “Smart Contracts” for vulnerabilities. As part of the future work, authors have specified the need for integrating

Devops with security mechanisms for continuous security analysis and to build a resilient system for Blockchain enabled Decentralized applications. Hence, the objective of the research question was to develop an automated pipeline to integrate a “Security” stage to obtain frequent feedbacks related to security issues and to resolve them at the earliest. The same has been implemented which will be discussed in later sections.

2.3 Incorporating Orchestration and containerization tools (Docker and Kubernetes) into the Devops pipeline (CI/CD) for DApp

This section outlines in detail the existing Literature on the orchestration and containerization techniques that have been utilized for the infrastructure management of Blockchain-based applications. It discusses the gap existing in the development, testing and deployment of the Blockchain applications, leading to research aimed in incorporating tools like Docker and Kubernetes to enhance the overall scalability and availability of the applications.

Rupsingh Mathwale (2023), in his paper, has discussed the development of an automated framework to deploy Hyperledger Fabric applications, which are nothing but permissioned Blockchain applications for enterprise settings. The author mainly aimed to automate the configurations that are required to setup the framework’s orderers, certificates and peer components that are generally difficult to integrate and deploy the same using Kubernetes. However, the same process can be improved in terms of speed of deployment, versioning, or enhanced code quality if aligned with the principles of Devops. Therefore, my research primarily aimed to automate the entire SDLC process including the “Deploy” stage for a Blockchain-based Decentralized application (DApp), which can later be utilized to test the effectiveness of the Devops principles on Hyperledger-based applications as well.

Zhenwu Shi, Chenming Jiang, Landu Jiang, Xue Liu (2021), addresses the issues involved in migrating the Blockchain-based applications to cloud environments. As the Blockchain applications function on the concept on Proof-of-Stake (POS) and Proof-of-Work (POW) consensus mechanisms, there has always been challenges in deploying these kinds of applications onto the cloud which needs to be addressed. Therefore, the authors have developed strategies to minimize the migration costs and keep the cloud prices within budget using Kubernetes scheduling. However, there has been no mention of automating the scheduling process using the latest software development methodologies, which could make the migration processes much easier as compared to using standalone methods. Therefore, an end-to-end Devops pipeline has been implemented as part of my research to convey the importance of integrating Devops practices into the software development life cycle of Blockchain based Decentralized applications.

Tomasz Gorski (2021), has addressed the absence of research in the field of continuous practices in relation to the design and development of Blockchain-based applications. The author aimed to create a pipeline using Jenkins Continuous Integration server to automate the deployment of the Blockchain-based applications. The deployment strategy utilized the concept of model-to-code transformation in order to deploy a set of yml files were configured to be deployed into Kubernetes cluster. Despite the integration of these design components, a pipeline can only be considered robust if it includes advanced security mechanisms and other code quality analysis tools that enhance the overall deployment of these applications. Therefore, an effort has been made to implement one such solution using Devops for a Blockchain-based DApp, which has proven to be efficient and the same is discussed in the following sections in this report.

Sultan. N.A. and Qasha. R.P (2023), in their work, provides a comprehensive overview of the integration of containerization and orchestration tools like Docker and Kubernetes for the deployment of Blockchain-based applications. The paper discusses several advantages offered by these tools in terms of flexibility, scalability, isolated environments and so on. However, authors have also discussed in detail regarding the challenges in containerization and opens the platform for discussion on the integration of Devops principles to automate the deployment process. The challenges and limitations are specifically with respect to resource utilization and scalability where the authors have emphasized on the need to perform a detailed analysis of how the resources can be utilized efficiently in Kubernetes for Blockchain applications. The paper also addresses the challenges involved in the security of containerized applications in terms of potential vulnerabilities and risks associated during the deployment of Blockchain applications into Kubernetes. Therefore, in order to overcome these challenges, an end-to-end automated Devops pipeline has been created along with the integration of advanced security mechanisms for a Blockchain-based DApp called “The Yoga Studio”, the results of which has been discussed in the later part of the report.

Eranga Bandara, Xueping Liang, Peter Foytik, Sachin Shetty, Nalin Ranasinghe, Kasun De Zoysa (2021), in their paper, “Rahasak—Scalable blockchain architecture for enterprise applications” identifies potential challenges faced by existing blockchain platforms, in terms of handling large amounts of workloads in environments pertaining to Internet-Of-Things (IOT) and Big-data. Through “Validate-Execute-Group” architecture and by using an Apache-Kafka based consensus mechanism, the authors have developed a framework called the “Rahasak” that addresses the challenges associated with the inability of Blockchain applications to manage scalability during varying workloads and has allowed for improved functionalities in data analytics. Although, frameworks like “Rahasak” have been developed for specific use-cases, they cannot be applied for all types of Blockchain-based applications such as DApps or cryptocurrencies. This existing work is relevant in my current research, highlighting the importance of having a robust and scalable system for Blockchain based application, as they often face scalability issues. Therefore, the integration of modern software development techniques like Devops can help mitigate these challenges and provide a platform for efficient development and deployment of Blockchain-based applications. In this regard, an automated deployment pipeline has been developed for deploying a DApp into Kubernetes, addressing the scalability challenges in Blockchain-based applications.

Khan. D, Jung. L.T and Hashmani. M.A (2021), provides an extensive overview on the scalability challenges faced by Ethereum and Bitcoin based public Blockchain platforms. Sone of the factors recognized by authors as having impact on scalability includes transaction throughput, storage, energy utilization, latency and so on. These variables are mentioned to have trade-offs between decentralization, scalability and security. They have also mentioned references to off-chain and on-chain resolutions, such as lightning networks and expansion of Block sizes, but highlighting the fact that these existing solutions are not enough to solve the scalability issues. Therefore, there is a need to implement a design that can combine the modern software development practices to build a robust solution to solve issues related to scalability. In this regard, the solution proposed and implemented through my research is the integration of Devops practices and Kubernetes that will help in addressing the problems associated with scalability. Kubernetes enables dynamic scaling of nodes based on demand and efficiently manages the computational workloads.

Q. Zhou, H. Huang, Z. Zheng and J. Bian (2020), in their work, provides a detailed analysis of the various scalability issues faced by Blockchain technology platforms such as Bitcoin and Ethereum. The authors have identified 3 layers of scalability solutions: Layer 0 includes improvements to hardware and network; Layer 1 suggests on-chain solutions to strength the consensus algorithms and Layer 2 recommends off-chain solutions to support payments during transactions. The study highlights that although there are numerous solutions that have been proposed, each solution has a trade-off in terms of preserving decentralization alongside scalability and security. Therefore, a comprehensive solution has been proposed and implemented through my research highlighting the need to build solutions that can bridge the gap between scalability and security. The solution implemented is the integration of Devops practices into Blockchain application development where it can help to address some of the challenges mentioned in this paper. Devops and Kubernetes collectively can provide a scalable environment along with automated infrastructure for application deployments with faster iteration cycles. This enhances fault tolerance of the system ensuring efficient utilization of resources.

3 Research Methodology

This research methodology outlines the importance of integrating Devops principles into the Software Development Life Cycle of Blockchain-based decentralized applications (DApp). The primary goal of the research conducted was to enhance the utilization of Devops principles in order to evaluate the various stages of the SDLC process using the core techniques of Devops methodology such as continuous integration, continuous delivery and continuous deployment. This enabled end-to-end automation of the various stages of the SDLC process that helped to assess the overall efficiency of the development procedure, ultimately improving the security, integrity and scalability of the Blockchain system. This particular section provides meaningful insights into the various steps followed to perform this research, materials and resources used, techniques employed to finalise the data/code on which the experiment has been conducted on, evaluation criteria and the ethical challenges that were considered.

3.1 Steps followed in the research

This section explains the various steps taken to conduct this research comprehensively and to derive at the conclusion that the “Devops integration for Blockchain application development” is a crucial area of focus.

3.1.1 Problem definition

The research process began with a thorough assessment of the existing literature to identify the key obstacles in integrating Devops methodology in Blockchain application development. The Literature Review that was carried out in this regard highlighted some important issues.

- *Speed of deployment:* Blockchain application developers make use of public testnets to validate the behaviour of Blockchain applications. For instance, in the case of DApps, such as "The Yoga Studio", extensive testing needs to be performed before deploying the smart contracts into the Ethereum mainnet. Hence, there is a need for faster deployment mechanisms to be employed so that developers can quickly evaluate the impact of the changes being made on the test networks that imitates the main network.
- *Versioning and release management:* The review highlighted the need to have a systematic and auditable release process in place in order to ensure the careful versioning of the Blockchain code to manage upgrades when necessary. As the Blockchain is immutable in nature, it is important to have an efficient versioning system in place.
- *Security*: Ensuring strong security related mechanisms to prevent vulnerabilities in the Blockchain code is another crucial issue that was pointed out in the review. As the decentralized applications consists of financial transactions and sensitive data logged into the peer-to-peer Blockchain network, it is required to make security as the top priority.
- *Scalability, consistency and availability*: The literature review conducted has emphasized on the importance of leveraging the benefits offered by containerization and orchestration tools to maintain the balance between scalability of Blockchain network with high availability, especially in the face of fluctuating loads.

These findings led to the formation of the research question and the need to practically implement the design to gauge the extent to which the system can be improved by addressing the issues mentioned in this section through the integration of Devops principles for Blockchain application development.

3.1.2 Information gathering

The study was conducted based on the information collected from multiple sources such as existing literature, industry reports, case-studies and so on. During the initial phases of information gathering, a comprehensive study was conducted to understand the different types of Blockchain applications and the technical stack that is required for their development. This helped in choosing the right resources to implement and test the solution that was proposed.

There are several types of Blockchain applications such as decentralized applications (DApps), Hyperledger Fabric applications, non-fungible Tokens (NFT), crypto currencies and so on. The choice of application on which the experiment will be performed was strategically made to produce the best possible results that will validate the research question.

The key factors based on which the decision was made to choose DApps over other types of Blockchain applications such as Hyperledger Fabric are (Apprecode, 2023):

- *Complexity involved due to decentralization*: DApps are deployed into typical public Blockchain networks such as Ethereum, where the participants are distributed across multiple nodes and they are not controlled by a central authority. Due to the decentralized nature, it introduces several challenges which makes it ideal to assess the effectiveness of Devops practices for DApp development through continuous integration and deployment processes. Hyperledger applications, on the other hand, are private permissioned Blockchain networks where the participants in the network are trusted. Therefore, as the network is not decentralized, integrating a Devops pipeline is less complex as compared to the DApps.
- *Market value and innovation*: Decentralized applications are growing rapidly in the marker featuring high level of innovation and there is scope for new opportunities in the field. However, the other types of Blockchain applications are already well established making it less favourable to test Devops methodologies.
- *User base and security*: DApps cater to a broad spectrum of users as these applications are part of various industry domains ranging from decentralized finance (DeFi) to supply chain management and social media. High scalability, availability and faster updates are required due to its wide spread usage. As a result, Devops will facilitate the continuous deployment process with faster iterations. Whereas, applications like Hyperledger Fabric, are enterprise-based settings that cater to private transactions and scalability requirements are more often predictable and different.
- *CI/CD integration challenges:* DApps involves the creation of smart contracts, which once deployed, cannot be altered due to the immutable nature of Blockchain technology. As a result, the implementation of CI/CD pipelines becomes challenging that requires careful considerations to ensure error-free deployments. Devops principles can provide insights into how these issues can be overcome. Hyperledger Fabric applications are more flexible in nature when it comes to update and governance mechanisms, which makes the process of CI/CD integration less complex.

Having said that, the name of the Decentralized application being used for the purpose of the research is “The Yoga Studio” whose technical specifications will be discussed in the following sections.

3.1.3 Architecting, implementation and evaluation

A theoretical model was developed for integrating Devops into the DApp development process. The model was developed using various Devops tools such as Github, Github actions for CI/CD, ESLint for code quality analysis, SonarCloud for security vulnerability detection, containerization tool such as Docker and Kubernetes for orchestration. The integration of these tools to evaluate deployment speed, security, scalability and availability of the DApp was tested efficiently in a controlled environment. The findings were validated against the existing studies and industry benchmarks.

3.1.4 Documentation

The practical implications of the Devops integration with Blockchain DApp has been highlighted and has been documented in detail. The format in which the results have been presented would serve beneficial for both scholars in academia as well those in the industry.

3.2 Ethical considerations

Ethics and data governance are crucial components that needs to be taken into account while conducting research. They help in guiding the research design and practices eventually leading to responsible development procedures. Additionally, they help in validating the research, uphold academic and scientific integrity and protects the rights of the individuals who are part of the research. While the integration of Devops principles for Blockchain application development offers numerous benefits, it also poses ethical challenges that needs to be addressed during the development phase. As part of the research and implementation, following ethical issues have been considered and addressed:

- *Data privacy*: It is an ethical principle sometimes referred to as "privacy of information", where it states that sensitive user information needs to be protected. In the context of the research, there are several Devops tools such as Docker, Kubernetes, Github actions, SonarCloud and so on that have been integrated to the pipeline. These tools have been configured to adhere to the standards like GDPR, where sensitive user information has been anonymized and managed securely.
- *Security*: All the credentials such as passwords, access tokens, AWS login details, secret keys and so on have been configured separately in properties files rather than being hardcoded into the pipeline to avoid data breaches and attacks.
- *Bias*: The security tools such as SonarCloud that has been integrated to detect security vulnerabilities has been configured to detect all types of security related errors and issues eliminating bias involved in the tool. Manul review of the code is also enabled.
- *Informed consent:* A thorough analysis has been conducted to check the engagement of human participants to guarantee the autonomy of decision-making process for them.
- *Conflict of Interest*: The research has been conducted impartially without bias or under the influence of outside parties.
- *Transparency*: Research strategies, assumptions, methodologies and limitations have been well-documented for the clear understanding of all readers, thus maintaining transparency.

In conclusion, this research methodology provides an exacting and thorough way to explore the integration of Devops practices for Blockchain application development. It also has the potential to significantly influence the organisations approach towards Blockchain application development in the future.

4 Design specifications

This section details the architectural framework, techniques, technical stack and resources that have been utilized to integrate Devops procedures for DApp development cycle.

4.1 Materials and equipment used

The research involved the usage of various resources to setup the experiment for analysis against Key Performance Indicators (KPIs). Below is a detailed inventory of resources that were utilized to achieve the solution of the proposed research question:

4.1.1 Software tools

- Source Control Management tool (SM): Github
- CI/CD tool: Github actions
- Containerization technology: Docker
- Orchestration: AWS Elastic Kubernetes Service (EKS)
- Code quality analysis: ESLint for Static Code Analysis
- Security vulnerability detection tool: SonarCloud
- Docker images repository: AWS Elastic Container Registry (ECR)
- Integrated Development Environment (IDE): Visual Code Studio (VCS) and AWS Cloud 9

4.1.2 Blockchain platform and technical stack

- Blockchain application: Decentralized application (DApp), The Yoga Studio
- Application software stack: Nodejs/Express framework
- Blockchain platform: Ethereum
- Smart Contracts deployment tool: Remix
- Smart Contracts code: Solidity programming
- Test network: Sepolia
- DApp Wallet: Metamask
- Computation Engine: Ethereum Virtual Machine

4.1.3 Infrastructure specification

- Hardware (AWS EKS Cluster) node: “t3.medium”, Kernel Version: 5.10.220-209.869.amzn2.x86_64, OS image: Amazon Linux 2
- Load Balancer Type: Classic

4.2 Architectural framework

The implemented architectural framework is designed smoothly to include Devops methods into the DApp development process. The framework is broadly divided into 3 main layers, each of which is explained in detail from sections 4.2.1 to 4.2.3.

4.2.1 Blockchain Layer: Architecture specifications of the "The Yoga Studio" Decentralized application

A Decentralized Application (DApp) is an application that operates on a peer-to-peer (P2P) Blockchain network which is open-source and distributed in nature. The application runs on multiple nodes on the network rather than on a single computer machine. DApps are considered to be part of the current evolution in the World Wide Web (WWW) called the Web3. The application is developed using "Nodejs" programming language and "Express" framework. Nodejs is an opensource, cross platform, runtime environment for application development in JavaScript. "Express framework", on the other hand, is a popular Node web framework that is capable of handling various HTTP requests such as GET, POST, DELETE and so on. "Bootstrap" has been used as a frontend framework for designing responsive web pages in a quick manner. DApps run on the popular Blockchain platform called the "Ethereum" network where the transactions are stored publicly in a distributed ledger.

The Ethereum network facilitates the secure execution of "Smart contracts". Smart contracts are small pieces of software programs that are written in object-oriented "Solidity" programming language and gets executed only when pre-defined conditions within the source code are met. For the purpose of implementation and development purpose, the smart contracts developed as part of "The Yoga Studio" called the "payment.sol" and "appointment.sol" are deployed into the Ethereum test network called the "Sepolia". Ethereum Sepolia test faucet is a developer tool that provides ethers (ETH) to test the DApp before being deployed into the "Ethereum mainnet". An Integrated Development Environment (IDE) called the Remix IDE was used to deploy the "Smart Contracts" into Sepolia test network. The solidity code is compiled by the Remix IDE to convert to bytecode which is eventually deployed into the "Ethereum Virtual Machine (EVM) that also generates a code called the "Application Binary Interface (AIB)" post deployment. The AIB code will be used along with the contract address to interact with the smart contracts from outside the Blockchain network. The "Smart Contracts" functionality is basically enabled by the EVM which is nothing but a network of multiple nodes. It also maintains the current state of a Blockchain network.

The test ethers are basically requested via the Alchemy account (https://www.alchemy.com/) using a wallet address such as Metamask. Alchemy offers a robust set of SDKs and APIs for building and scaling of Decentralized applications. Metamask wallet, originally developed by ConsenSys Software Inc., a blockchain-based organisation, is a cryptocurrency wallet that enables interaction between users and the Blockchain network. The wallet is accessed by users via a mobile application or installed as a browser extension. Figure 1 represents the overall architecture of "The Yoga Studio" DApp.

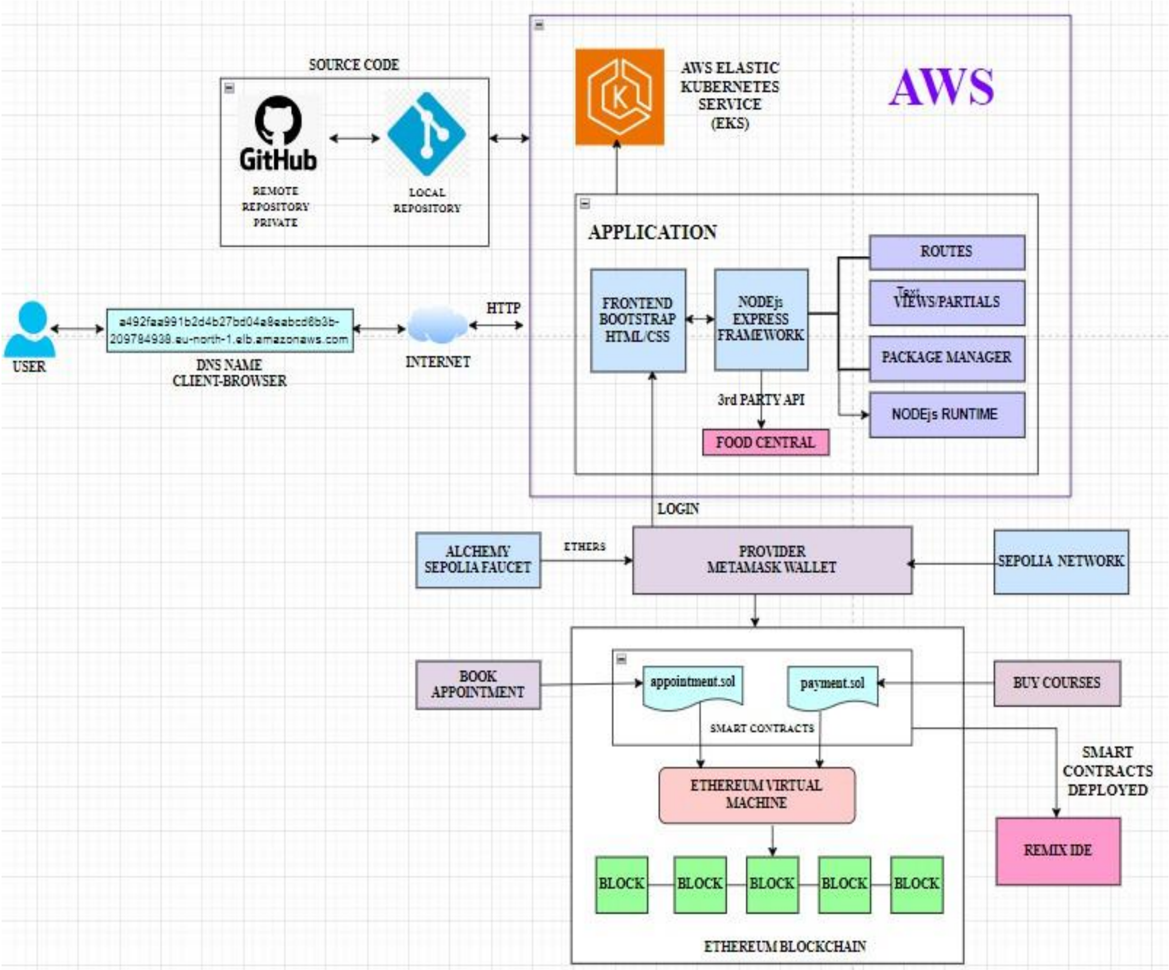


Figure 1: Architecture diagram of "The Yoga Studio" DApp

4.2.2 *Devops pipeline: Continuous Integration and Continuous Deployment: Containerization and Orchestration*

The primary goal of the research was to implement an end-to-end automated Devops framework in order to evaluate the performance of the development process for a Blockchain based application called "The Yoga Studio" DApp. A CI/CD pipeline enables automated deployments by allowing developers to integrate the code changes frequently, reliably and quickly. Hence CI/CD process is considered to be the foundation of the Devops methodology. In this regard, to achieve a full-fledge automated pipeline, several Devops tools were integrated into the CI/CD pipeline at various stages. This section provides insights into the different tool sets used and details regarding the new configuration scripts that has been developed to facilitate the integration of Devops principles into the DApp development process.

To begin with, the source code repository that was used to maintain the frequent code commits was "Github". Github is a web-based Source Code Management (SCM) tool, that allows for efficient source code versioning, control and collaboration, offering features such as pull requests, commit histories, issue tracking and continuous integration via Github actions. "Github actions" is a continuous integration and "Continuous Delivery" platform that enables developers to create custom workflows for their source code repositories to build, test and deploy the applications automatically. Github provides several virtual machines such as Windows, Linux and macOS to execute the workflows. A YAML-based script called the “main.yml” has been created to define the workflows at various stages of the pipeline. The pipeline is integrated with an open-source JavaScript utility called as “ESLint” that identifies and reports linting errors in “JavaScript” code (ESLint, 2024). This enhances the code quality as it detects bugs and code convention errors during the development phase. Another tool integrated as part of the Devops pipeline is the security vulnerabilities tool called the “SonarCloud”. It’s a Software-as-a-Service (SaaS) code analysis tool that detects bugs, security issues, security hotspots, code smells, vulnerabilities automatically enabling developers to maintain a clean and secure code base (Sonarsource, 2024).

The application deployment was done using containerization and orchestration techniques using tools such as “Docker” and “AWS Elastic Kubernetes Service”. Docker is an open-platform containerization tool that enables packaging of the applications and its dependencies into separate “containers” to maintain isolation, portability and consistency (Docker, 2024). These containers are managed and coordinated for automated deployments, scaling, and networking using orchestration techniques. Therefore, the tool that was used for the orchestration purpose was “AWS Elastic Kubernetes Service” which is a managed service provided by AWS. It eliminates the need for manual human intervention to maintain the Kubernetes Control plane components. Figure 2 represents the overall design of the research that was conducted and its implementation as part of Devops framework for the Decentralized Blockchain application development.

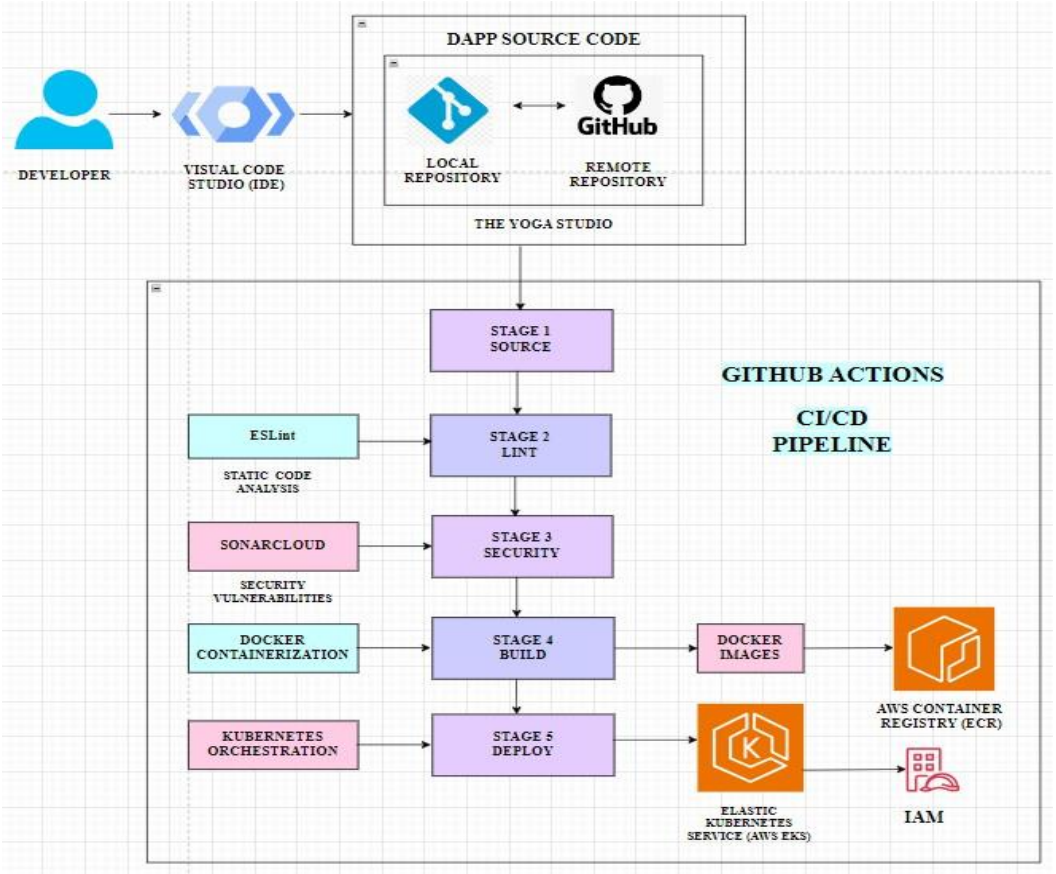


Figure 2: Architecture diagram of Devops integration (CI/CD) with Blockchain-based DApp

5 Implementation

The implementation phase of the research focused on developing a robust, secure and scalable system to deploy and manage a Blockchain-based Decentralized application (DApp) by integrating Devops principles into its development lifecycle on Amazon Web Services (AWS). The primary goal was to create an end-to-end automated CI/CD pipeline, that incorporates advanced code quality analysis and security mechanisms leveraging the benefits of containerization and orchestration techniques using Docker and Kubernetes. This implementation aimed to improve the software engineering process of the DApp by evaluating the output obtained against the level of resilience, security, availability and integrity achieved. The various software tools and infrastructure used to implement the solution has already been discussed in "Section 4: Design Specifications". This section provides insightful information of how the implementation was done to obtain the desired output.

5.1 Phase 1: Development of the Decentralized application, "The Yoga Studio"

The DApp, developed in Nodejs/Express framework is a Decentralized application that interacts with Ethereum network. The application was initially tested locally using the Visual Code Studio (VCS) IDE. Key development activities included the following:

- *Development of Smart contracts*: The "Smart contracts", that are required for user interaction with the Blockchain network, was created using solidity programming language and deployed using Remix IDE. These programs were thoroughly tested for security and functionality.
- *User Interface (UI):* Bootstrap libraries have been used as part of the UI design and development. These libraries hold a collection of HTML, CSS and Javascript related class names copied into the "public/js" and "public/css" folders.
- *Backend services:* These services were developed using Nodejs to manage the interaction between UI and the Blockchain network.

The architecture diagram of the application showing the integration between the Smart Contracts, UI and the backend services is represented in Figure 1 under "Section 4: Design specifications".

5.2 Phase 2: Setup of CI/CD workflow via "Github actions"

The CI/CD pipeline was constructed to automate the various stages of the development process such as building, testing and deployment process of the DApp. The different stages were defined and YAML files (".github/workflows/main.yml") were configured for each stage with its respective trigger conditions. This section outlines the various stages that were implemented to obtain the desired results that eventually validated the research question.

5.2.1 Stage 1 - Source

This phase was implemented by integrating the source code repository with Github Actions (CI server), which triggered the workflow every time there were new changes and commits being pushed into the repository on Github.

5.2.2 Stage 2 - Lint - Static Code Analysis via ESLint

Linting stage was introduced into the pipeline in order to ensure code quality and to maintain consistent coding standards, at the early stages of development process. Linting enables the detection of bugs, errors and anti-patterns by identifying these issues before moving the code into the staging/production environments. As a result, cost is reduced as it prevents fixes later on that are time-consuming. ESLint is a popular linting tool for JavaScript applications which was configured to obtain immediate feedback for developers. The configuration of ESLint included the installation of ESLint global dependencies from package.json file and by configuring a job in “main.yml” file in “Github actions” to run on specific folders in the source code to detect code related issues.

ESLint offers several advantages over other tools in the market. Firstly, it offers substantial level of customisation where it allows developers to configure their own linting rules, extend configuration settings, and integration with wide variety of plugins. ESLint is also updated to support modern ECMAScript features and hence it can be easily configured for modern frameworks in Javascript. In addition to this, ESLint has a broad community support making it compatible to work with frameworks like react and angular.

5.2.3 Stage 3 - SonarCloud scan for detecting Security Vulnerabilities in Blockchain DApp

“SonarCloud scan” is the third stage in the Devops pipeline, configured through the “main.yml” script and it runs after the “Lint” stage. The primary goal of implementing this phase was to introduce the concept of “DevOpsSec” where security related issues are detected during the early stages of the development lifecycle. At this stage, the Github workflow performs a security scan on the code, where the script checks for “High” and “Medium” severity security issues from SonarCloud, and if any “high-severity” issues are encountered, the application deployment was blocked and the pipeline exited with error. The pipeline was re-run after resolving the issues.

SonarCloud is a cloud-based service that integrates with version control systems enabling developers to resolve issues related to security vulnerabilities, bugs, technical debt, code smells and so on. The tool offers wide variety of features as compared to other security tools in the market. Some of the features available in SonarCloud include authentication method through Single Sign-on (SSO) using Security Assertion Markup Language (SAML), project configuration capabilities for across the organisation, comprehensive metrics for code analysis in terms of coverage, adherence, duplication and so on, and support for more than 25 programming languages.

SonarCloud integration between the CI/CD pipeline and the SonarCloud Web Interface was enabled by configuring “SONAR_TOKEN” to authenticate the pipeline. “GITHUB TOKEN” environment variable was used by SonarCloud to interact with Github repository.

The scan was triggered by the action "SonarSource/sonarcloud-github-action@master" in the "main.yml" script. Github transfers the results to the SonarCloud web portal via the SonarCloud API.

5.2.4 Stage 4 – Build using Docker for containerization

The application is prepared to be deployed at this stage after the successful completion of "Lint" and "SonarCloud scan" stages. Containerization techniques are suitable for Blockchain-based application development as it provides consistency across all the environments by behaving the same way across development, testing, staging and production environments. It offers isolation of containers by separating application components, thereby reducing the risk of vulnerabilities. Containers contribute to a rapid deployment cycle and enables scaling quickly.

This stage is triggered for deployment into Amazon Elastic Container Registry (AWS ECR) and AWS Elastic Kubernetes Service (AWS EKS) through the Github Actions workflow. The AWS credentials are configured to integrate the pipeline with the AWS ECR. At the beginning of this stage, required dependencies were installed including the Web3 libraries and compiler for the solidity code. The workflow checked for changes in the "Smart contracts". If there were any changes that were being found, the code was compiled and deployed once again using the Remix IDE. The code changes were then pushed to the repository to re-execute the pipeline. "Dockerfile" is configured to use the official "node.js" base image to setup the Docker container into which the application was deployed. Docker builds the Docker images form the Dockerfile and pushes the docker image to the AWS Container Registry that maintains all the versions of the docker images. This ensured that the latest version of the application code was packaged into the Docker container and pushed to ECR for deployment.

5.2.5 Stage 5 – Deploy into AWS Elastic Kubernetes Service (AWS EKS)

The packaged application was then deployed into AWS EKS by initially configuring the pipeline to connect with AWS EKS by installing certain plugins like "kubectl" which enables the execution of "kubectl" commands against the correct Kubernetes Cluster. Kubernetes ensures orchestration of the containerized Blockchain DApp by automating the deployments, ensuring efficient scaling as and when required, executing administrative tasks and to ensure all the pods and nodes are up and running as per the configuration of the cluster. This way the application availability is maintained throughout.

AWS EKS is a managed orchestration service that aligns with the general configuration of Kubernetes with control plane components and worker machines called nodes that forms the underlying architecture. EKS handles the configuration of the clusters, thus lowering the operational overhead on users. Scalability is another feature that is handled automatically by EKS based on the demand. EKS also maintains high availability as it replicates the Kubernetes components such as the control-plane and the worker nodes across availability zones. The managed service also allows integration with "Identity and Access Management"

(IAM) framework for Role-Based Access Controls (RBAC), facilitating secure connection to nodes and services.

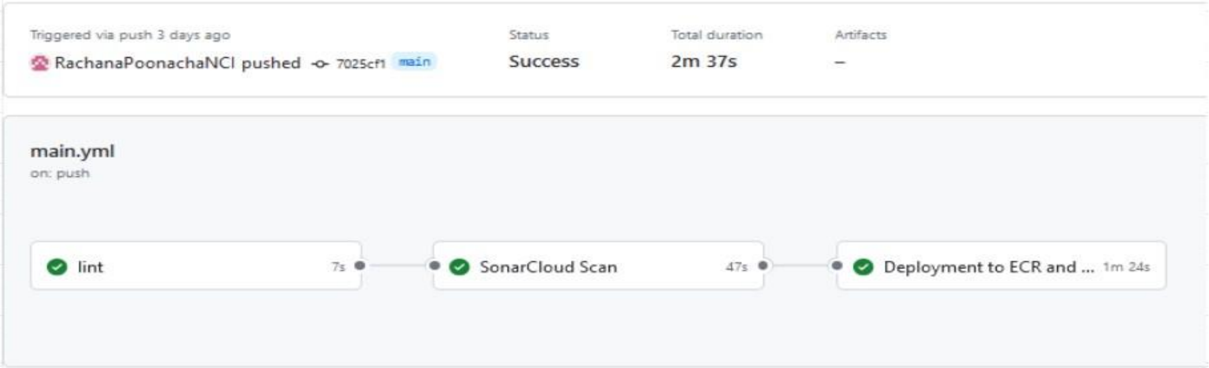
The AWS EKS cluster was created before the deployment of the application. The EKS Cluster is configured through YAML scripts such as “yoga-app-deployment.yaml” and “yoga-app-service.yaml”. These configurations will allow the pods to run inside the cluster nodes which hosts tightly-coupled containers into which the DApp is deployed.

AWS EKS automatically provisions a Classic Load Balancer and the application is accessed via the DNS name of the Load Balancer. Therefore, the application URL is: “http://a492faa991b2d4b27bd04a8eabcd6b3b-209784938.eu-north-1.elb.amazonaws.com”. Figure 3 represents a successful CI/CD pipeline stages as obtained from Github Actions for “The Yoga Studio” DApp. Figure 4 represents the nodes and pods running on AWS EKS Cluster. Figure 5 represents the UI of “The Yoga Studio” DApp via the DNS.

The results and evaluation of the output obtained from implementation is discussed in the following section.

6 Evaluation

This section will present the key findings of the study and highlights the significant outcomes laying the groundwork for discussions and interpretation.



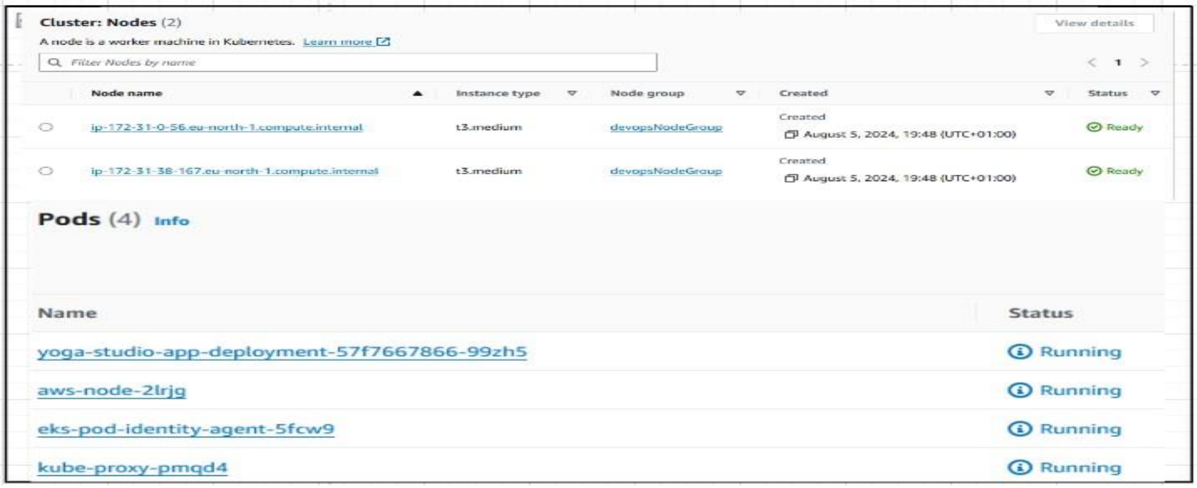
Triggered via push 3 days ago
RachanaPoonachaNCI pushed 7025cf1 main

Status	Total duration	Artifacts
Success	2m 37s	–

main.yml
on: push

✓ lint 7s → ✓ SonarCloud Scan 47s → ✓ Deployment to ECR and ... 1m 24s

Figure 3: CI/CD Pipeline for the DApp from Github Actions



Cluster: Nodes (2)

A node is a worker machine in Kubernetes. Learn more

Filter Nodes by name

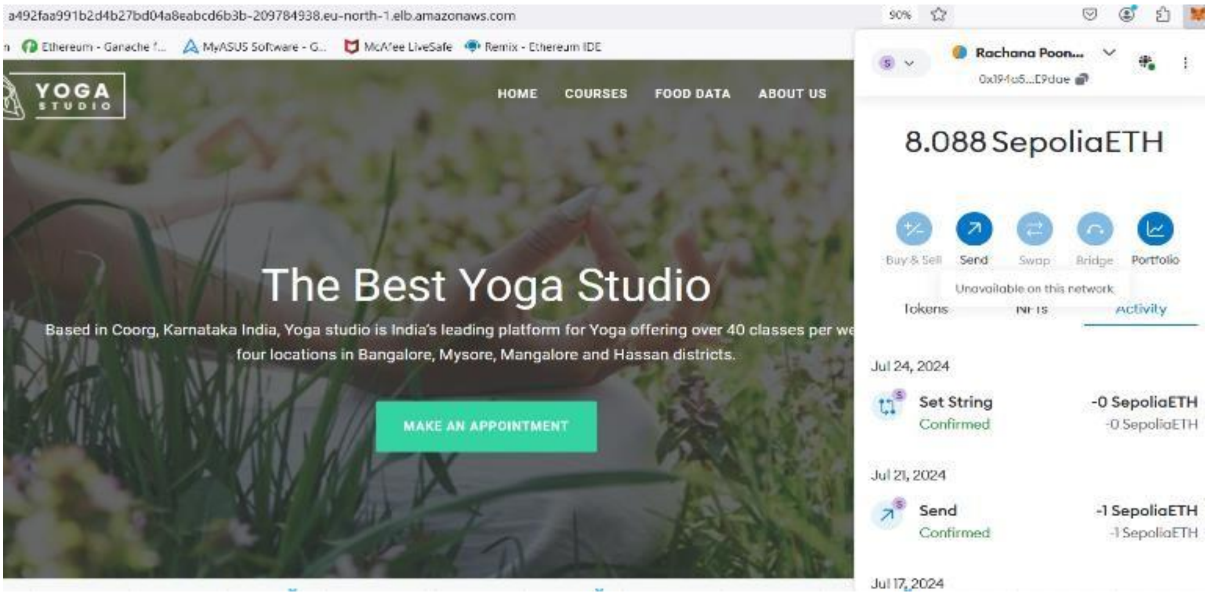
View details

	Node name	Instance type	Node group	Created	Status
○	ip-172-31-0-56.eu-north-1.compute.internal	t3.medium	devopsNodeGroup	Created August 5, 2024, 19:48 (UTC+01:00)	Ready
○	ip-172-31-38-167.eu-north-1.compute.internal	t3.medium	devopsNodeGroup	Created August 5, 2024, 19:48 (UTC+01:00)	Ready

Pods (4) Info

Name	Status
yoga-studio-app-deployment-57f7667866-99zh5	Running
aws-node-2lrjg	Running
eks-pod-identity-agent-5fcw9	Running
kube-proxy-pmqd4	Running

Figure 4: Application up and running in AWS EKS

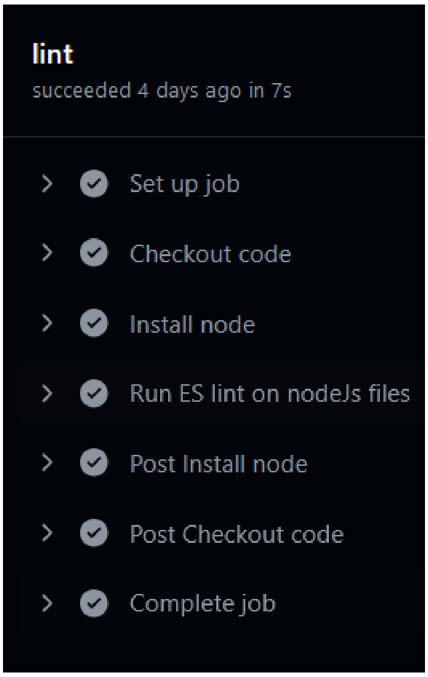


Based in Coorg, Karnataka India, Yoga studio is India's leading platform for Yoga offering over 40 classes per we four locations in Bangalore, Mysore, Mangalore and Hassan districts.

Figure 5: "The Yoga Studio" DApp

6.1 Experiment 1: At the "Lint" stage of the CI/CD Pipeline

The integration of ESLint tool in the Devops CI/CD pipeline played a crucial role in achieving JavaScript coding standards by identifying potential code issues and contributed towards strengthening the security of the application. Figure 6 represents the different steps that was executed as part of the "Lint" job configured in the Devops pipeline on "Github Actions".



lint

succeeded 4 days ago in 7s

- Set up job
- Checkout code
- Install node
- Run ES lint on nodeJs files
- Post Install node
- Post Checkout code
- Complete job

Figure 6: "Lint" stage in CI/CD pipeline

The CI/CD pipeline failed to execute the subsequent stages such as "SonarCloud scan" and "Deploy" after it encountered Linting errors at the "Lint" stage in the code as shown in

Figure 7. The errors were identified in the "/routes/" folder at the following location: "/home/runner/work/Blockchain_latest_v2/Blockchain_latest_v2/routes/index.js" and the errors were displayed in the CI/CD pipeline as shown in Figure 8.

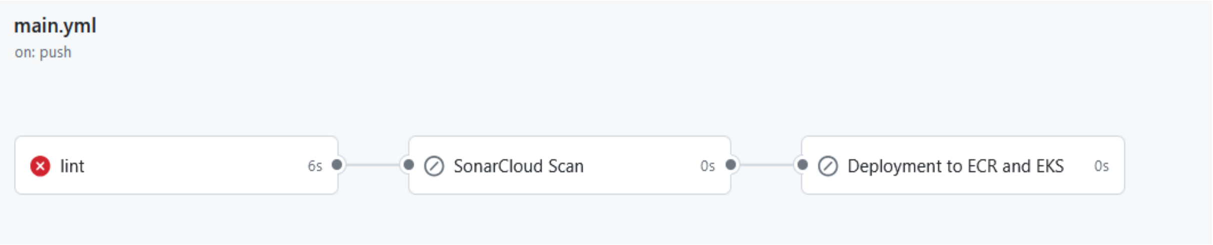
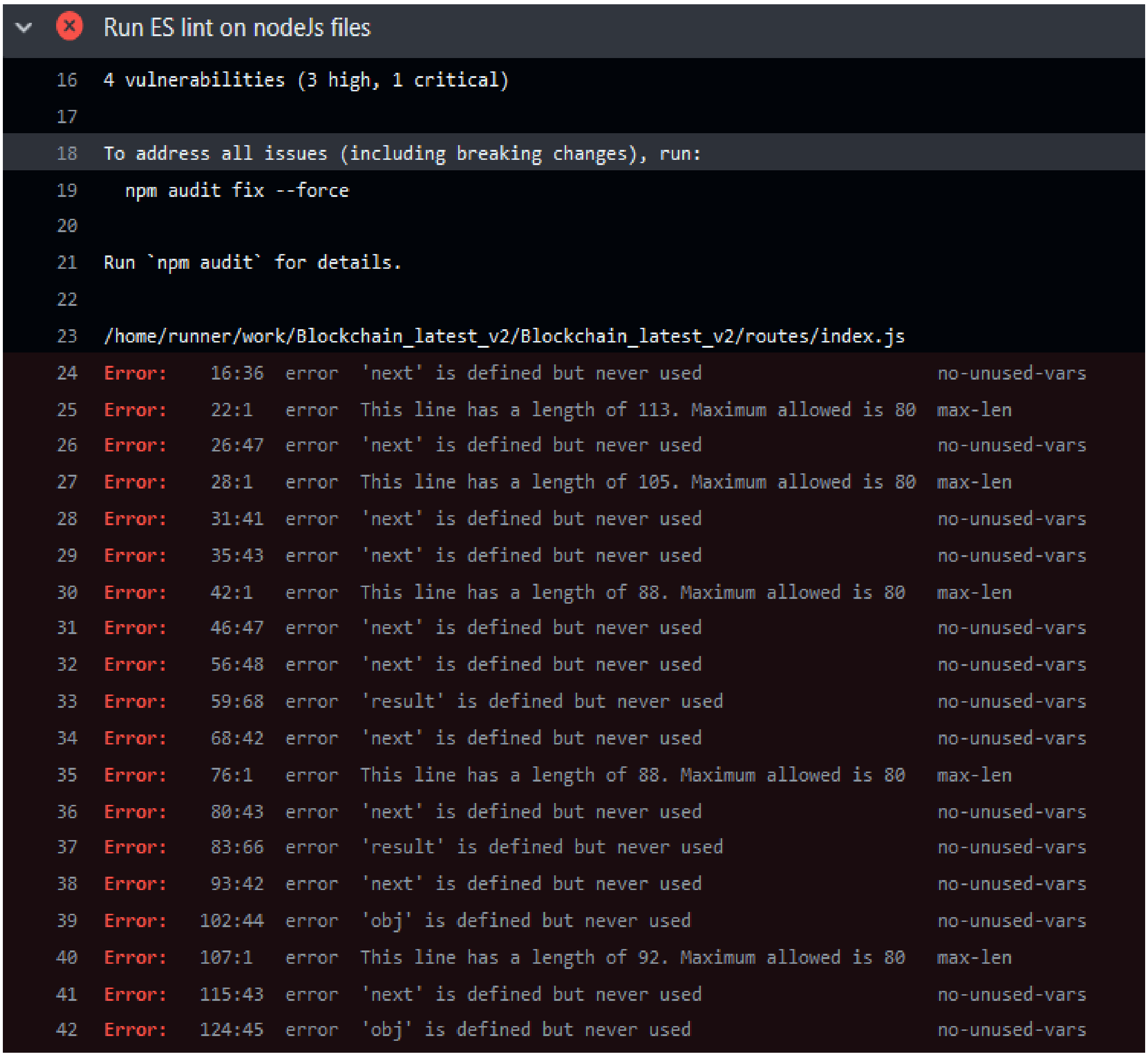


Figure 7: CI/CD pipeline failure due to Linting errors

A notification will be sent to the administrator of the source code repository informing about the unsuccessful jobs in the Github workflow.

The identified errors related to "no-unused-vars" and "max-len" were corrected locally and the job was re-triggered, resulting in a successful CI/CD pipeline as shown in Figure 9.



Run ES lint on nodeJs files

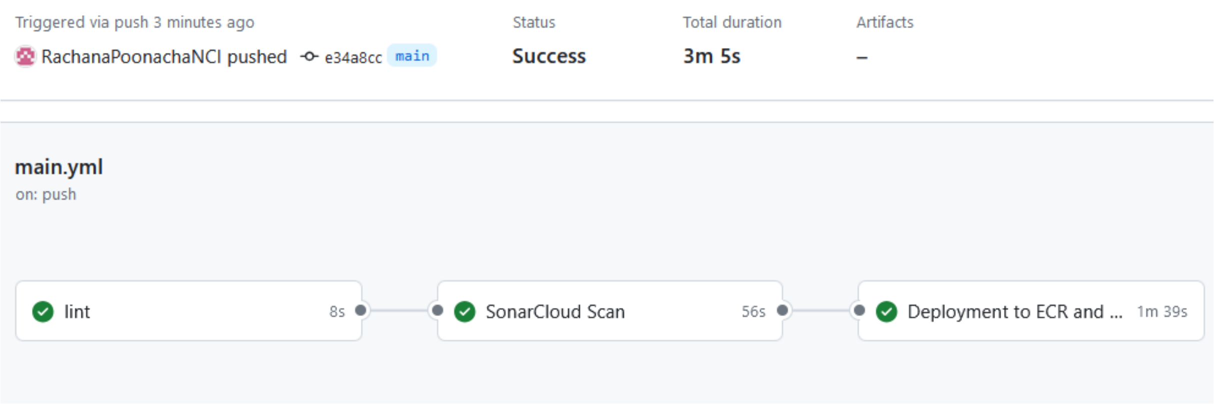
```
4 vulnerabilities (3 high, 1 critical)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

/home/runner/work/Blockchain_latest_v2/Blockchain_latest_v2/routes/index.js
Error:    16:36  error  'next' is defined but never used                  no-unused-vars
Error:    22:1   error  This line has a length of 113. Maximum allowed is 80  max-len
Error:    26:47  error  'next' is defined but never used                  no-unused-vars
Error:    28:1   error  This line has a length of 105. Maximum allowed is 80  max-len
Error:    31:41  error  'next' is defined but never used                  no-unused-vars
Error:    35:43  error  'next' is defined but never used                  no-unused-vars
Error:    42:1   error  This line has a length of 88. Maximum allowed is 80   max-len
Error:    46:47  error  'next' is defined but never used                  no-unused-vars
Error:    56:48  error  'next' is defined but never used                  no-unused-vars
Error:    59:68  error  'result' is defined but never used                no-unused-vars
Error:    68:42  error  'next' is defined but never used                  no-unused-vars
Error:    76:1   error  This line has a length of 88. Maximum allowed is 80   max-len
Error:    80:43  error  'next' is defined but never used                  no-unused-vars
Error:    83:66  error  'result' is defined but never used                no-unused-vars
Error:    93:42  error  'next' is defined but never used                  no-unused-vars
Error:   102:44  error  'obj' is defined but never used                   no-unused-vars
Error:   107:1   error  This line has a length of 92. Maximum allowed is 80   max-len
Error:   115:43  error  'next' is defined but never used                  no-unused-vars
Error:   124:45  error  'obj' is defined but never used                   no-unused-vars
```

Figure 8: Linting errors identified in "/routes/" folder



Triggered via push 3 minutes ago
RachanaPoonachaNCI pushed e34a8cc main

Status
Success

Total duration
3m 5s

Artifacts
–

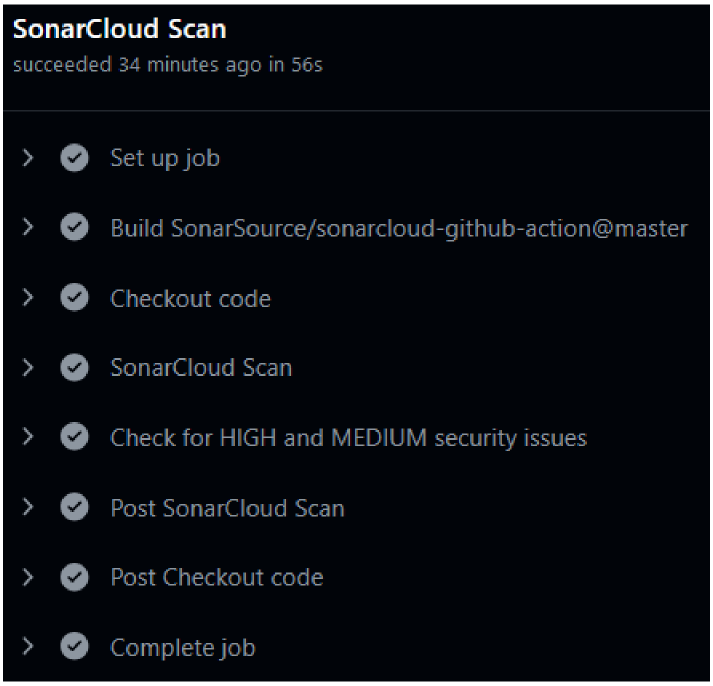
main.yml
on: push

✓ lint 8s
✓ SonarCloud Scan 56s
✓ Deployment to ECR and ... 1m 39s

Figure 9: Linting errors fixed and the job triggered to obtain a successful CI/CD pipeline

6.2 Experiment 2: At the “SonarCloud scan” stage of the CI/CD Pipeline for Security evaluation

The “SonarCloud scan” stage plays a crucial role in ensuring the security of the DApp by identifying security issues such as Denial of Service (DoS), permissions issues, consistency and maintainability issues, insecure configurations, Cross-site Request Forgery (CSRF) and so on. By incorporating SonarCloud into the Devops pipeline, the security related issues are detected and resolved at early stages of the development lifecycle. Figure 10 represents the different steps that was executed as part of the “SonarCloud scan” job configured in the Devops pipeline on “Github Actions”.

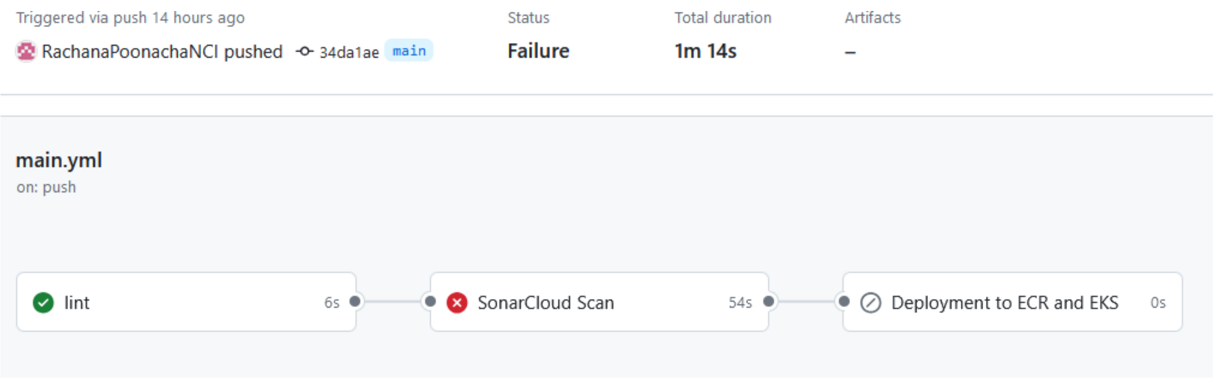


SonarCloud Scan
succeeded 34 minutes ago in 56s

- ✓ Set up job
- ✓ Build SonarSource/sonarcloud-github-action@master
- ✓ Checkout code
- ✓ SonarCloud Scan
- ✓ Check for HIGH and MEDIUM security issues
- ✓ Post SonarCloud Scan
- ✓ Post Checkout code
- ✓ Complete job

Figure 10: “SonarCloud scan” stage in CI/CD pipeline

The CI/CD pipeline was setup with specific conditions where the pipeline will fail to execute in case a “HIGH” or “MEDIUM” severity “SECURITY” issue was encountered as shown in Figure 11. This ensured that the identified vulnerabilities were addressed way before the application was being deployed into Kubernetes.


Triggered via push 14 hours ago
RachanaPoonachaNCI pushed 34da1ae main

Status
Failure

Total duration
1m 14s

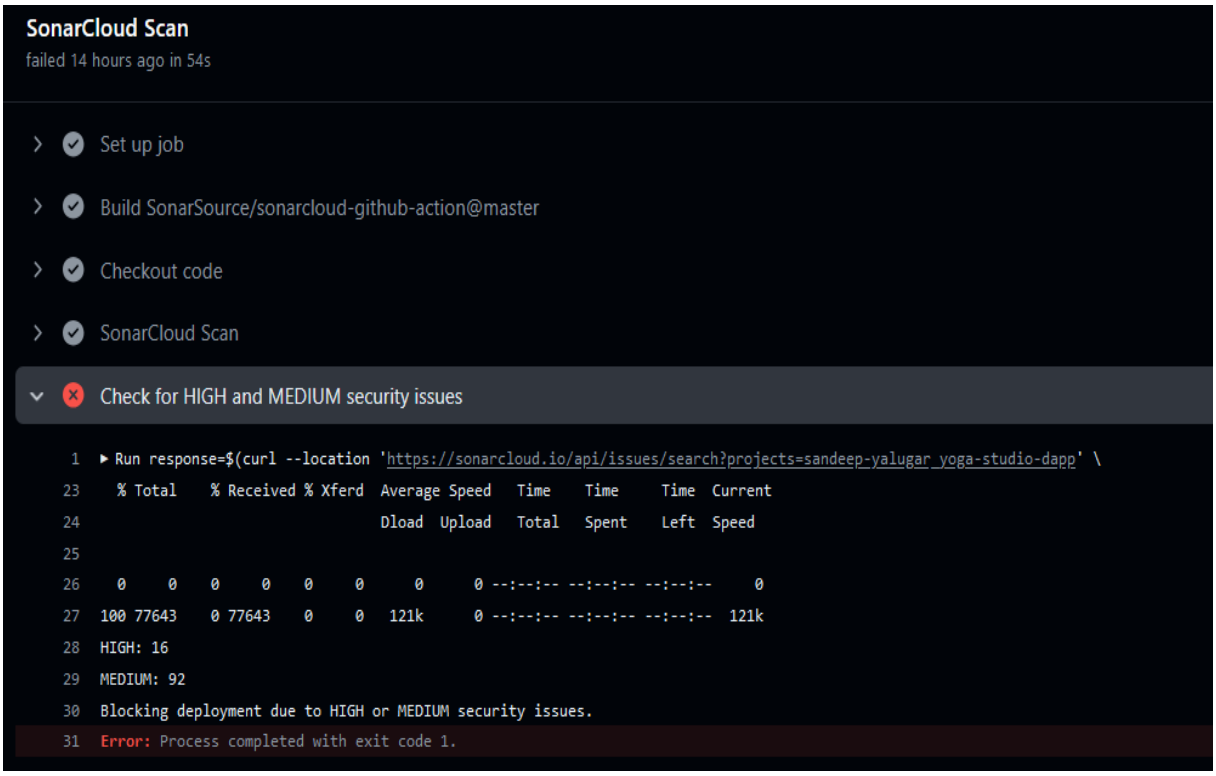
Artifacts
–

main.yml
on: push

lint 6s
SonarCloud Scan 54s
Deployment to ECR and EKS 0s

Figure 11: “SonarCloud scan” failed due to security issues in CI/CD Pipeline

Figure 12 represents the error details of the failed CI/CD pipeline due to the detection of security issues. The issues detected can then be analysed through the “SonarCloud” Web Interface which is integrated with Github. The tool provides a “Summary” of the “Overall code” that was analysed including details of where the issue is coming from, reason for the issue, how the issue can be fixed, and so on as shown in Figure 12. The issues can then be reviewed and resolved as per the fixes provided in the tool as shown in Figure 13 where the pipeline ran successfully with “zero” security issues. This will enable a strong security mechanism for the Blockchain-based DApp. Figure 14 represents a successfully run pipeline where the application was deployed into Kubernetes after fixing the encountered security issues.

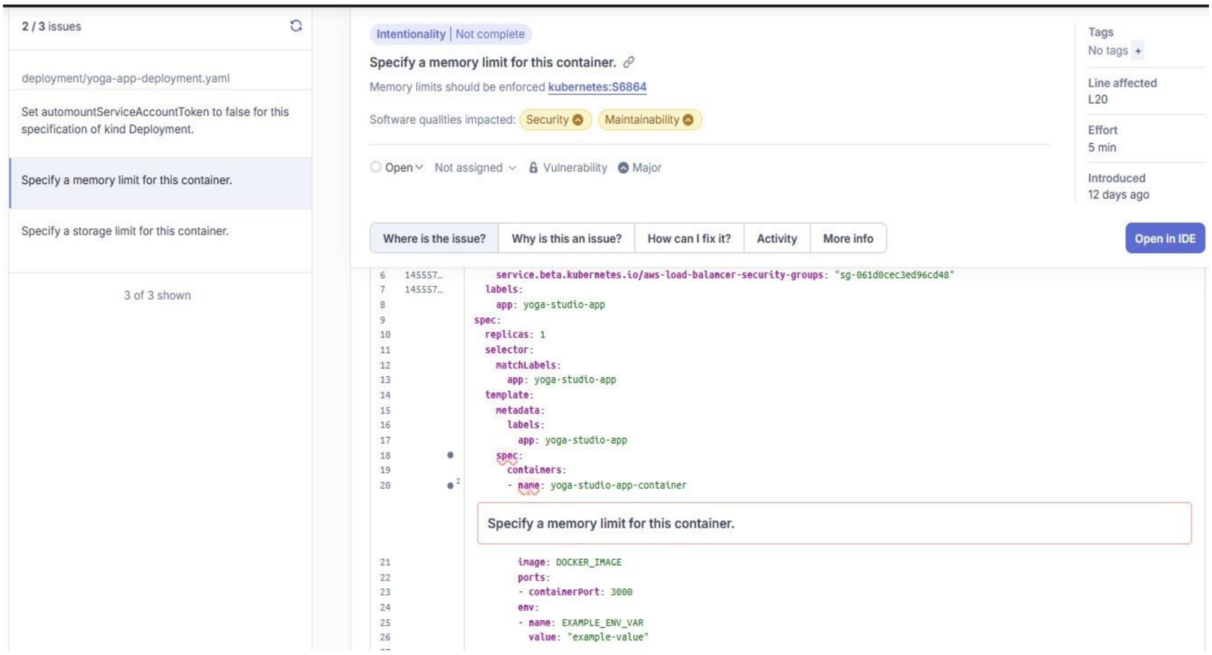

SonarCloud Scan
failed 14 hours ago in 54s

Set up job
Build SonarSource/sonarcloud-github-action@master
Checkout code
SonarCloud Scan
Check for HIGH and MEDIUM security issues

```
▸ Run response=$(curl --location 'https://sonarcloud.io/api/issues/search?projects=sandeep-yalugar_yoga-studio-dapp' \
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 77643    0 77643    0     0   121k      0 --:--:-- --:--:-- --:--:--  121k
HIGH: 16
MEDIUM: 92
Blocking deployment due to HIGH or MEDIUM security issues.
Error: Process completed with exit code 1.
```

Figure 11: “SonarCloud scan” report with errors from Github Actions



2 / 3 issues

deployment/yoga-app-deployment.yaml

Set automountServiceAccountToken to false for this specification of kind Deployment.

Specify a memory limit for this container.

Specify a storage limit for this container.

3 of 3 shown

Intentionality | Not complete

Specify a memory limit for this container.

Memory limits should be enforced kubernetes:S6864

Software qualities impacted: Security Maintainability

○ Open Not assigned Vulnerability Major

Tags
No tags +

Line affected
L20

Effort
5 min

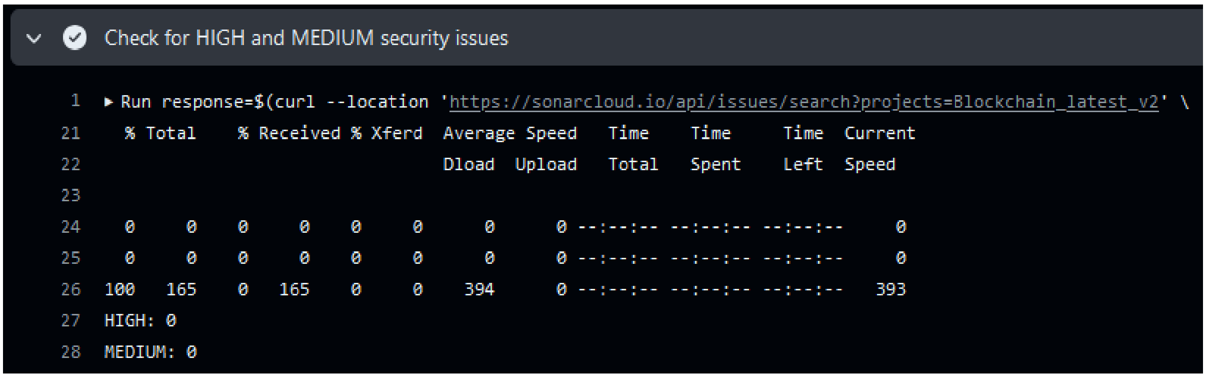
Introduced
12 days ago

```
    service.beta.kubernetes.io/aws-load-balancer-security-groups: "sg-061d0cec3ed96cd48"
  labels:
    app: yoga-studio-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: yoga-studio-app
  template:
    metadata:
      labels:
        app: yoga-studio-app
    spec:
      containers:
      - name: yoga-studio-app-container
```

Specify a memory limit for this container.

```
        image: DOCKER_IMAGE
        ports:
        - containerPort: 3000
        env:
        - name: EXAMPLE_ENV_VAR
          value: "example-value"
```

Figure 12: “SonarCloud scan” report from Web UI with security issues

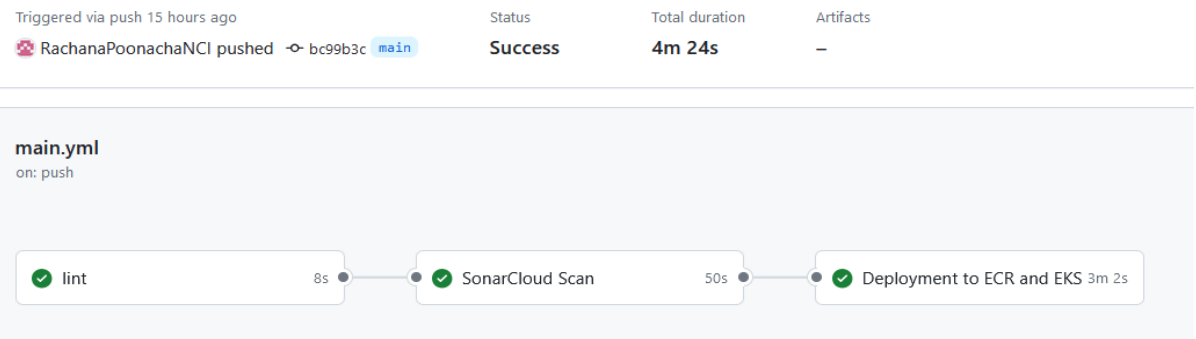


Check for HIGH and MEDIUM security issues

```
▸ Run response=$(curl --location 'https://sonarcloud.io/api/issues/search?projects=Blockchain_latest_v2' \
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100   165    0   165    0     0    394      0 --:--:-- --:--:-- --:--:--   393
HIGH: 0
MEDIUM: 0
```

Figure 13: “SonarCloud scan” report from Web UI after fixing the security issues



Triggered via push 15 hours ago
RachanaPoonachaNCI pushed bc99b3c main

Status
Success

Total duration
4m 24s

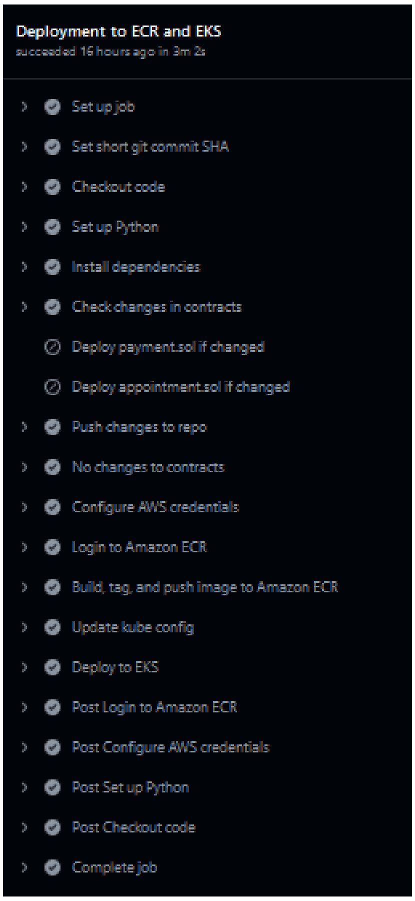
Artifacts
–

main.yml
on: push

Figure 14: “SonarCloud scan” executed successfully after resolving the security issues in Github actions

6.3 Experiment 3: At the “Deploy” stage of the CI/CD Pipeline, application deployment into AWS EKS Cluster

The “Deploy” stage of the pipeline is where the application is deployed into AWS Elastic Kubernetes Service (AWS EKS). This is an important step in the CI/CD process to automate the deployments so that a reliable and consistent code is delivered to the production environment. The various steps executed as part of the pipeline is as shown in Figure 15.



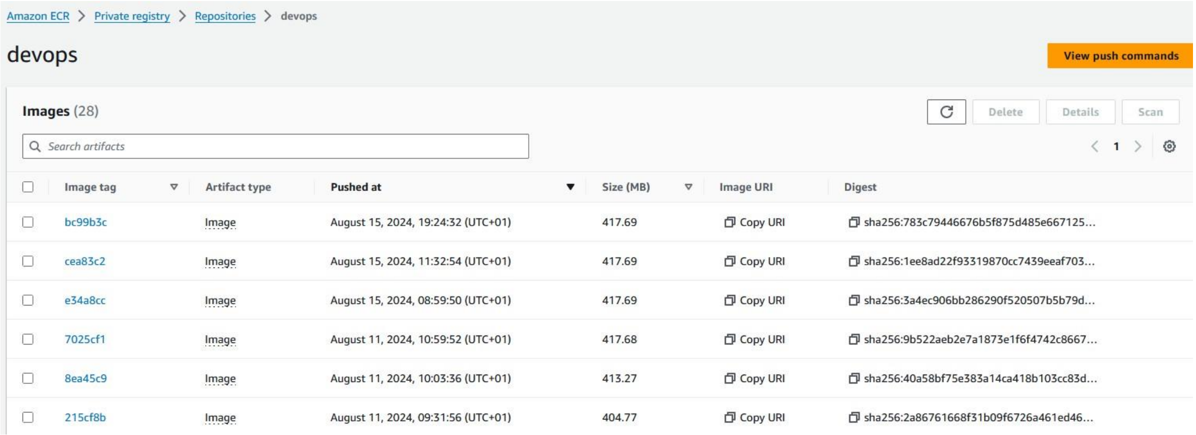
Deployment to ECR and EKS
succeeded 16 hours ago in 3m 2s

```
> ✓ Set up job
> ✓ Set short git commit SHA
> ✓ Checkout code
> ✓ Set up Python
> ✓ Install dependencies
> ✓ Check changes in contracts
  ⊘ Deploy payment.sol if changed
  ⊘ Deploy appointment.sol if changed
> ✓ Push changes to repo
> ✓ No changes to contracts
> ✓ Configure AWS credentials
> ✓ Login to Amazon ECR
> ✓ Build, tag, and push image to Amazon ECR
> ✓ Update kube config
> ✓ Deploy to EKS
> ✓ Post Login to Amazon ECR
> ✓ Post Configure AWS credentials
> ✓ Post Set up Python
> ✓ Post Checkout code
> ✓ Complete job
```

Figure 15: “Deploy” stage in CI/CD pipeline

The 5 major steps that were executed in this stage following the “SonarCloud scan” stage included the following:

- *Environment setup and code checkout*: The pipeline started with checking out the code from the Github repository and the necessary dependencies were installed.
- *Detecting change in Smart Contracts*: The pipeline checked for any modifications to the “Smart Contracts”. This was to make sure that the latest modified “Smart Contracts” code was compiled for deployment.
- *Deployment of Smart Contracts*: In case of any changes detected in the “Smart Contracts” code, the respective “Smart Contracts” were compiled to retrieve the AIB code generated and other binary files. These compiled programs were then deployed using a python deployment script.
- *Docker build and push*: Docker build the image of the application and pushes the same to AWS Elastic Container Registry (AWS ECR) for storing by tagging each image to the Git commit SHA as shown in Figure 16.



Images (28)

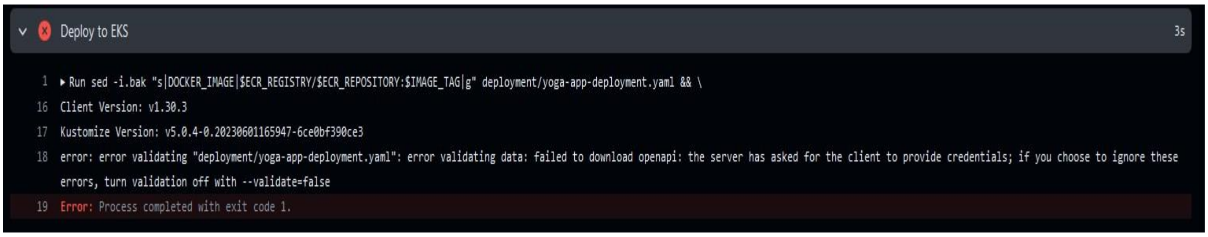
□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
□	bc99b3c	Image	August 15, 2024, 19:24:32 (UTC+01)	417.69	Copy URI	sha256:783c79446676b5f875d485e667125...
□	cea83c2	Image	August 15, 2024, 11:32:54 (UTC+01)	417.69	Copy URI	sha256:1ee8ad22f93319870cc7439eeaf703...
□	e34a8cc	Image	August 15, 2024, 08:59:50 (UTC+01)	417.69	Copy URI	sha256:3a4ec906bb286290f520507b5b79d...
□	7025cf1	Image	August 11, 2024, 10:59:52 (UTC+01)	417.68	Copy URI	sha256:9b522aeb2e7a1873e1f6f4742c8667...
□	8ea45c9	Image	August 11, 2024, 10:03:36 (UTC+01)	413.27	Copy URI	sha256:40a58bf75e383a14ca418b103cc83d...
□	215cf8b	Image	August 11, 2024, 09:31:56 (UTC+01)	404.77	Copy URI	sha256:2a86761668f31b09f6726a461ed46...

Figure 16: Docker images of the DApp being stored in AWS ECR

- *Deployment of the application into AWS EKS*: Finally, the pipeline configures the Kubernetes environment to deploy the latest Docker image to AWS Elastic Kubernetes Service (AWS EKS), making sure that the application is running smoothly in the cloud environment.

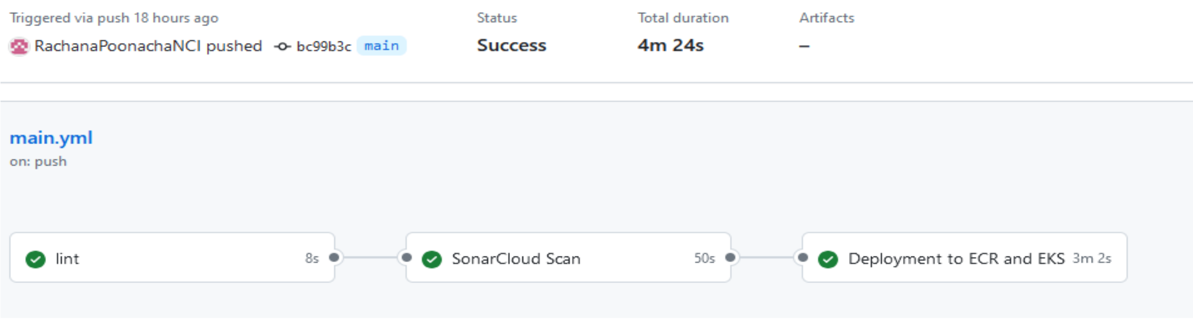
The entire process was automated due to the integration of Devops methodology for the Blockchain application development, which is essential for maintaining the reliability of the Decentralized application ("The Yoga Studio" DApp).

The CI/CD pipeline at the "Deploy" stage can fail due to several reasons as shown in Figure 17. The errors were resolved and the pipeline was re-run to obtain a successful deployment of the DApp into AWS EKS Cluster as shown in Figure 18.



```
▸ Run sed -i.bak "s|DOCKER_IMAGE|$ECR_REGISTRY/$ECR_REPOSITORY:$IMAGE_TAG|g" deployment/yoga-app-deployment.yaml && \
Client Version: v1.30.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
error: error validating "deployment/yoga-app-deployment.yaml": error validating data: failed to download openapi: the server has asked for the client to provide credentials; if you choose to ignore these
errors, turn validation off with --validate=false
Error: Process completed with exit code 1.
```

Figure 17: Error in deployment from Github Actions



Triggered via push 18 hours ago
RachanaPoonachaNCI pushed bc99b3c main

Status: Success
Total duration: 4m 24s
Artifacts: –

main.yml
on: push

Figure 18: Successful deployment of the application to AWS EKS through the Devops pipeline

6.4 Discussion

The experiments conducted in this research – starting from the integration of "SOURCE" stage in the CI/CD pipeline, spanning the "Lint" stage, "SonarCloud scan" stage, "Docker build" and "Deploy" stages, has offered meaningful insights into the integration of Devops practices into Blockchain application development.

The "Source" stage in the CI/CD pipeline highlighted the importance of integrating the code changes continuously with the Continuous Integration (CI) server like Github actions, by providing the foundation for deployment. The "Source" stage is essential for improving the code quality, facilitating collaboration among the members of the team where every developer would be working on the most recently updated code base, provides continuous feedbacks to the developers thereby reducing the debugging time, rapid iterations and faster release cycles and so on.

The results from "Experiment 1: At the "Lint" stage of the CI/CD Pipeline", reveals the importance of maintaining coding standards early in the development process. The tool was successful in identifying code issues like convention errors, bugs, unused variables, improper syntax, and so on, thereby preventing likelihood of errors being induced further along the pipeline, to the later stages of development. While the integration of "ESLint" tool was efficient enough to detect code issues during the development phase of the Blockchain-based DApp, improvement to the rulesets to make it more aligned with specific project configurations, could be more effective. This enhancement can be added in future iterations.

In "Experiment 2: At the "SonarCloud scan" stage of the CI/CD Pipeline for Security evaluation", it was found that the integration of vulnerability detection tool like the "SonarCloud" has effectively demonstrated the capacity of the pipeline to stop the deployment of the application when security issues of "HIGH" and "MEDIUM" severity levels are present. Some of the security issues that were resolved include: Auto mount of service tokens in Kubernetes deployment yaml that could lead to unauthorized access/data breach/privilege escalation, memory limit that was not set for a particular container that could lead to insufficient resource allocation and Denial-of-Service, storage limit not set for containers that could lead to unexpected application behaviour or resource exhaustion in nodes and so on. Although the findings have proved satisfactory to begin with, incorporating a second line of defence like dynamic analysis of the code with threat modelling and penetration testing can improve the state of the security of the Decentralized Application.

A critical analysis of "Experiment 3: At the "Deploy" stage of the CI/CD Pipeline, application deployment into AWS EKS Cluster", reveals the significance of smooth automated deployments in a cloud-native setup. Kubernetes deployment enables automatic scaling of containers in scenarios of fluctuating loads in DApps due to the varying volumes of transactions. It ensures high availability of the Blockchain DApp by maintaining the replicas of the containers across multiple nodes thereby minimizing the downtime in case of a node failure. As the Blockchain transactions are resource intensive, use of Kubernetes for orchestration will optimize the use of resources by scheduling the containers efficiently based on the demand and availability. As the Blockchain applications are decentralized, Kubernetes deployment supports this by decentralized management of services that are containerized across all the nodes in the cluster. Additionally, Kubernetes distributes traffic equally through load-balancing ensuring that the reliability and responsiveness of the DApp is maintained. Having said that, the experiment could have explored more on the different deployment strategies and induced load testing methods to validate the responsiveness of the application in case of fluctuating loads.

In summary, the conducted experiments provided valuable insights into some more key factors such as: “The speed of deployment”, which was accelerated due to the automated deployments and quick fixes in the code as the pipeline was integrated with Linting and security tools that helped in identifying the code related errors and security issues in the early stages of development, “Versioning and release management” that was effectively handled by tagging the docker images with the Git commit SHA resulting in changes being tracked systematically in Github as well as in AWS ECR, “Security” was dramatically improved due to the addition of “SonarCloud” security tool and “Availability, Resilience, Scalability, Integrity” of the application was supported by deploying the application into AWS EKS guaranteeing that the DApps could manage the performance and dependability in distributed settings.

Overall, the experiments conducted has laid the groundwork for the integration of Devops into the development of the Blockchain DApp, a significant enhancement over the approaches documented in the existing Literature.

7 Conclusion and Future Work

The goal of this study was to provide a solution to the research question, “Can the integration of advanced security mechanisms for vulnerability scanning triggered at the release build time of a CI/CD pipeline and orchestration technology using Kubernetes enhance the resilience, integrity, scalability and availability of Blockchain based applications such as DApps where the results of the scan can be recorded and evaluated for security threats and issues at the early stages of the Blockchain Software Development Life Cycle?”. The primary objective was to assess the software development process of the Blockchain-based Decentralized application when integrated with the Devops principles. A critical analysis of the implemented solution has led to the conclusion that there was a tremendous improvement in the overall system security, availability, integrity and resilience. In this regard, the work included the designing and implementation of a framework such as a CI/CD pipeline based on the Devops methods, incorporating security mechanisms and linting tools into the pipeline to detect security vulnerabilities and enhance the code quality during the early stages of the development. Additionally, orchestration through Kubernetes, contributed to improved availability and scalability, ensuring that the DApps could maintain the performance through varying conditions.

As part of the future work, the developed CI/CD pipeline can be optimized to explore the integration of advanced security scanning tools for “Dynamic Application Security Testing (DAST)” of DApps. In addition to this, the pipeline can be integrated with “Testing” stage to perform functional tests, solidity code tests, load tests, performance tests, compliance tests, audit tests and so on. Finally, the developed pipeline can also be adapted to test similar factors as done in this research on Blockchain-based framework called the Hyperledger Fabric.

References

Apprecode (2023), *“DevOps for Blockchain: Implementing Distributed Ledgers with Continuous Integration”,* Available at: https://apprecode.com/blog/devops-for-blockchain-implementing-distributed-ledgers-with-continuous-integration (Accessed: 13 August 2024).

Docker (2024), *“Docker Overview”,* Available at: https://docs.docker.com/guides/docker-overview/ (Accessed: 13 August 2024).

Eranga Bandara, Xueping Liang, Peter Foytik, Sachin Shetty, Nalin Ranasinghe, Kasun De Zoysa (2021), "Rahasak—Scalable blockchain architecture for enterprise applications", *Journal of Systems Architecture,* Volume 116, 2021, 102061, ISSN 1383-7621, https://doi.org/10.1016/j.sysarc.2021.102061.

ESLint (2024), *“Getting started with ESLint”,* Available at: https://eslint.org/docs/latest/use/getting-started (Accessed: 13 August 2024).

Hongsong Chen, Xietian Luo, Lei Shi, Yongrui Cao, Yongpeng Zhang. (2023), ‘Security challenges and defense approaches for blockchain-based services from a full-stack architecture perspective’ *Blockchain: Research and Applications,* Volume 4, Issue 3, 2023, 100135, ISSN 2096-7209, https://doi.org/10.1016/j.bcra.2023.100135.

K. Duan, J. M. Caballero and X. Jing. (2023), ‘Scrum-based development model: Improve the engineering quality and testing method of blockchain projects,’ *in 2023 8th International Conference on Business and Industrial Research (ICBIR).* Bangkok, Thailand, 2023, pp. 807-811, doi: 10.1109/ICBIR57571.2023.10147439.

Khan. D, Jung. L.T, Hashmani. M.A (2021), "Systematic Literature Review of Challenges in Blockchain Scalability", *Appl. Sci. 2021*, 11, 9372. https://doi.org/10.3390/app11209372.

M. J. H. Faruk, S. Subramanian, H. Shahriar, M. Valero, X. Li and M. Tasnim. (2022) “Software Engineering Process and Methodology in Blockchain-Oriented Software Development: A Systematic Study” *in 2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA).* Las Vegas, NV, USA, 2022, pp. 120-127, doi: 10.1109/SERA54885.2022.9806817.

M. R. Islam, M. M. Rahman, M. Mahmud, M. A. Rahman, M. H. S. Mohamad and A. H. Embong (2021), ‘A Review on Blockchain Security Issues and Challenges’ *in 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC)*, Shah Alam, Malaysia, 2021, pp. 227-232, doi: 10.1109/ICSGRC53186.2021.9515276.

M. Shoaib Farooq & U. Ali (2023), *"Harnessing the Potential of Blockchain in DevOps: A Framework for Distributed Integration and Development",* Available at: https://www.researchgate.net/publication/371222810_Harnessing_the_Potential_of_Blockcha in_in_DevOps_A_Framework_for_Distributed_Integration_and_Development (Accessed: 12 August 2024) .

M. Wöhrer and U. Zdun, "DevOps for Ethereum Blockchain Smart Contracts," *2021 IEEE International Conference on Blockchain (Blockchain),* Melbourne, Australia, 2021, pp. 244-251, doi: 10.1109/Blockchain53845.2021.00040.

N. Sánchez-Gómez, J. Torres-Valderrama, J. A. García-García, J. J. Gutiérrez and M. J. Escalona (2020), "Model-Based Software Design and Testing in Blockchain Smart Contracts: A Systematic Literature Review," *in IEEE Access*, vol. 8, pp. 164556-164569, 2020, doi: 10.1109/ACCESS.2020.3021502.

Noama Fatima Samreen, Manar H. Alalfi. (2023), “An empirical study on the complexity, security and maintainability of Ethereum-based decentralized applications (DApps)”, *Blockchain: Research and Applications*, Volume 4, Issue 2, 100120, ISSN 2096-7209, https://doi.org/10.1016/j.bcra.2022.100120.

P. Zheng, Z. Jiang, J. Wu and Z. Zheng (2023), "Blockchain-Based Decentralized Application: A Survey," *in IEEE Open Journal of the Computer Society*, vol. 4, pp. 121-133, 2023, doi: 10.1109/OJCS.2023.3251854.

Q. Zhou, H. Huang, Z. Zheng and J. Bian (2020), "Solutions to Scalability of Blockchain: A Survey," *in IEEE Access*, vol. 8, pp. 16440-16455, 2020, doi: 10.1109/ACCESS.2020.2967218.

Reyes, M. Jimeno, R. Villanueva-Polanco (2023) *"*Continuous and Secure Integration Framework for Smart Contracts*", Sensors 2023*, 23, 54, https://doi.org/10.3390/s23010541.

R. Hegadi, S. S. K. Akella, K. Reddy and P. Kumar C (2023), "Analysing and Mitigating Common Vulnerabilities in Smart Contracts in Web3 Ecosystem: A Comprehensive Study," *2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS),* Mangaluru, India, 2023, pp. 1-5, doi: 10.1109/ICDDS59137.2023.10434537.

Rupsingh Mathwale (2023) "AHFD: A Framework for Deployment and Management of Hyperledger Fabric Enterprise Blockchain", *2023 International Conference on Data Science and Network Security (ICDSNS)* 979-8-3503-0159-5.

S. Bankar and D. Shah, "Blockchain based framework for Software Development using DevOps", *in 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE),* Navi Mumbai, India, 2021, pp. 1-6, doi: 10.1109/ICNTE51185.2021.9487760.

S. M. Idrees, M. Nowostawski, R. Jameel, A.K. Mourya (2021). 'Security Aspects of Blockchain Technology Intended for Industrial Applications' *Electronics 2021,* 10, 951. https:// kdoi.org/10.3390/electronics10080951.

Sonarsource (2024), *"SonarCloud Documentation",* Available at: https://docs.sonarsource.com/sonarcloud/ (Accessed at: 13 August 2024).

Sultan. N.A. and Qasha. R.P (2023), "Container-based Virtualization for Blockchain Technology: A Survey", *Jordanian Journal of Computers & Information Technology*, 9(3).

Tomasz Gorski (2021), "Towards Continuous Deployment for Blockchain", *Appl. Sci. 2021*, 11, 11745. https://doi.org/10.3390/app112411745.

W. Fan, H. J. Hong, X. Zhou and S. Y. Chang (2021), "A Generic Blockchain Framework to Secure Decentralized Applications," *ICC 2021 - IEEE International Conference on Communications,* Montreal, QC, Canada, 2021, pp. 1-7, doi: 10.1109/ICC42927.2021.9500924.

Zhenwu Shi, Chenming Jiang, Landu Jiang, Xue Liu (2021), "HPKS: High Performance Kubernetes Scheduling for Dynamic Blockchain Workloads in Cloud Computing," *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, Chicago, IL, USA, 2021, pp. 456-466, doi: 10.1109/CLOUD53861.2021.00060.