A Complete Guide to DevOps Best Practices

<https://doi.org/10.5281/zenodo.6376787>

International Journal of Computer Science and Information Security (IJCSIS),Vol. 20, No. 2, February 2022

Justin Onyarin OgalaDepartment of Computer Science, Faculty of Computing,University of Delta, Agbor, Delta State, Nigeria justinoo2001@gmail.com

Abstract

— DevOps is a systematic methodology, tools, andphilosophical framework for automating and integratingsoftware development and IT operations teams. Teamempowerment, cross-team communication and cooperation,and technological automation are all emphasized. DevOpsrefers to a collection of integrated activities or procedures usedto automate and interconnect software development processeswith IT developers with the goal of swiftly and reliablyproducing, testing, and delivering deliverables. DevOps hasresulted in developers or practitioners using endless loops tohighlight the link between development lifecycle phasesregularly. Even though the numerous activities or processes ina DevOps build a loop and flow sequentially, iteration requiresthat the flow be collaborative and repeated at all times in orderto improve the entire lifecycle. Various software automationtrends might be equipped to manage the industry's currentsoftware and technology if DevOps investigates themthoroughly.

Index Terms

— DevOps, CI/CD, Software developmentpractices,

I.

I

NTRODUCTION

evOps refers to a collection of integrated activities or procedures used to automate and interconnect softwaredevelopment processes with IT, developers, with the goal ofswiftly and reliably producing, testing, and deliveringdeliverables. DevOps is a word that relates to bothdevelopment and operations, and it reflects a culturalrelationship between developers and operators whosefunctions were previously separated (Whittle, 2014).DevOps has resulted in developers or practitioners usingendless loops to highlight the link between developmentlifecycle phases regularly. Although the many activities or processes in a DevOps make a loop and flow sequentially,the iteration suggests that the flow must be continuallycollaborative and repeated to enhance the overall lifecycle(Gruver, 2016). Communication was a key factor in thedevelopment of DevOps. Other developers and designerssaw the process as a collaborative way for developers andengineers to use iterative automation to complete jobs andother activities as quickly as possible.The integration of the development and operations teamswas aided by several reasons. The notion was born out ofthe organizations' long-standing challenges and concerns,since their tasks are greatly scattered, making theiroperations more complex (Armstrong, 2016). The capacityto tackle difficulties from an individual position got less asthe complexity chain grew more complicated. Providingvirtual machines in a large network region, difficult setup ofextended network devices and servers, and installingnumerous applications are only a few of the essentialinterventions that led to the merger (Armstrong, 2016).Furthermore, log collecting and aggregation, monitoringservices, network performance monitoring, and application performance monitoring had all become complicated issues.Furthermore, the developers, operators, and engineers wereunable to comprehend the events that led to thesecomplexities, resulting in false alarms and the need forremediation.DevOps Best Practices

i.

Combination of Strength

When a diverse collection of operators and developersworked, they came up with a reasonable solution to the real-world operating complexity (Gruver, 2016). Thesecomplicated processes have been broken down intomanageable chunks for simple comprehension. Monitoring, performance management, deployment automation,infrastructure automation, log management, andconfiguration management are among them.

ii.

Infrastructure automation

Unlike in the past, when infrastructural components hadto be built from the ground up, now there is no need to buildservers from the ground up, acquire electricity, or manuallylink data centers. In addition, manually connecting amachine to a network is no longer necessary. Most of these procedural and time-consuming processes have becomeineffective and inessential when the developers andoperations teams were combined. Previously, these processes were laborious and tedious, and they onlyhappened after an operation had run into problems (Gruver,2016). All operations may be carried out, controlled, andmaintained in a single location utilizing a simple code,according to DevOps. The emergence of cloud servicesallows users to access real-time information via a webapplication. This eliminates the requirement for data centersor silos to perform normal operations. Most processes, suchas IT infrastructure, became automated once DevOps wasintroduced, eliminating the need to physically visit datacenters and silos to receive hardware services or makenetwork changes. Cloud services are extremelyadvantageous since they lower linear demand costs andallow for automatic provisioning without having to pay forhardware services (Meena, 2014). These services are

by a variety of suppliers. Azure, Heroku, Ubuntucloud, Amazon web services, HP cloud, RackSpace cloud,and EngineYard are some of the options.

iii.

Configuration management

Previously, when the hardware was installed, developers ornetwork users had to manually implement, execute, andconfigure numerous packages. DevOps brings the power ofintelligent or automated configuration management to bearon the problem. DevOps automates the installation andsetup of hardware packages through the use of simplescripted scripts. Because servers are constantly and in real-time deployed in every event, computerized setup solutionsare advantageous. Assume a user has to make changes oradjustments to many machines. In that situation, all usershave to do is execute them on a single system, and they willautomatically duplicate over all other devices (Meena,2014). Puppet, Ansible, Pallet, Chef, Salt Stack, and Bcfg2are some of the suppliers providing DevOps tools. Theautomated tools are also provided by Amazon's OpsWorksand Rightscale providers.Previously, most operational settings had rigorousstandards and limitations on who had access to the production field, who could make changes, and when thesechanges may be implemented. Physical engagement withhardware, particularly in data centers, was required to bringabout these improvements (Meena, 2014). Theseexpectations and unrelated procedures have hampered thecapacity of development and operations to work together.However, the application and maintenance of variousDevOps methodologies and stand-alone operations are stillunclear. Before deciding on the optimal strategy to deploymodels for infrastructure development, one must firstconsider the operating viewpoint, which is critical.Furthermore, when deploying an application in a pre- production region, the Ops teams of developers mustcomprehend the development perspective. (2014, Google)

iv.

Continuous Monitoring

Initially, the developers and operations team had to physically manage or control all of the systems in a networkto guarantee that there were no defects or possible threats.These operations were inefficient and ineffective. Thedesigns are automated using DevOps to guarantee that thesystems' essential notifications are never missed by theoperators, organizations, or commercial organizations.Monitoring, messaging, and ticketing tools are all firmlyinterwoven into these platforms. Filtered noise is used inthese techniques to boost signals (Google, 2014). Unlike previous models, which issued alerts to a single machine orunit, DevOps systems convey signals to numerous channelsand provide the required steps to remedy issues, whether potential or ongoing.Aside from issuing alarms, DevOps-enabled solutions havesimplified and streamlined on-call administration. Within asingle user interface, the designer may create and modifyschedules as well as specify escalation policies. Theoperation team will know who is on the line and who isresponsible in the event of an emergency. The criticalalarms are always acknowledged when they emerge in everyincident. Furthermore, the monitoring systems provide forcomprehensive reporting and analytics (Veritis, 2016). Theautomated algorithms can identify areas of success as wellas places where improvements can be made. All warningsand foreign incident-related occurrences are tracked by thesystems. The plans also include strong reporting andanalytical patterns for revealing the source of the alert, aswell as the team's performance in terms of risk identificationand resolution. It also manages how workload is distributedacross multiple channels in a network.

v.

Log Management

Primary features such as infrastructure monitoring play akey part in the running of production applications,regardless of how quickly businesses adopt new technologyadvancements. The observability idea is vital fordeployment success with minimal disruptions in every stageof operation and a steady delivery pipeline as enterprisesand other organizations gain traction (Kuchler, 2016).Developers and operations teams may monitor andunderstand the application's behavior before releasing it to production using log and event management, which is partof the delivery pipeline's entire process. Although some mayclaim that log management adds to the development process, work, and time, it aids in the growth of theorganization's important software development activitieswhile avoiding difficulties that may be avoided throughout production. It may also aid in the creation of a fluid,seamless user interaction stage, reducing the need to re-architect production solutions (Kuchler, 2016).

vi.

Continuous Testing

The significant requirement for log management insidehighly dispersed systems has risen as the technologicalenvironment in numerous industries has evolved. Therehave been significant advancements in the creation of appsand services, as well as the ability to deploy applicationsacross numerous logs as a service vendor and the pervasiveness of containers. In terms of service capacity building with diverse languages, the demand for datagathering, monitoring, and tracking across a linked networkhas risen dramatically (Bass et al., 2015). The logmanagement procedure used to be a time-consuming andarduous task that entailed executing search commands on alllocal servers. In instances when the server clusters wererestricted to 20 servers and individual operators wererequired, the system had limited scalability. With DevOps,there has been an increase in the number of virtualizedsystems with minimal operational expenses inside a normal business firm. With the rising virtualization of systemsunder DevOps, independent developers now can generatedesigns in parallel while using the best-suited technologyfor production. These events take place in a sharedenvironment forconstruction, production, and staging. Logaggregation eliminates customization difficulties whendevelopers use several technologies with different log formats. (Bass et al., 2015).

vii.

Observability

DevOps has developed some rules for data monitoring frommany perspectives to forecast a system's correctness andstability. Observability is based on three interconnectedideas that help developers and operations teams operatemore efficiently. The first component is externalmonitoring, which looks at external run-ups to verify thatweb applications provide consumers with a contemporaryexperience (Bass et al., 2015).The metrics and distributed tracings, on the other hand,identify linkages between distributed applicationsthroughout a system. It also aids in the detection of errorsand exceptions from an application, as well as the necessaryresolution procedures. Finally, the events and logs aid in the provision of context information from invested data in thegiven situation. When combined with other concepts, it alsomakes it possible to track out bugs in a program.II.

L

ITERATURE

R

EVIEW

As technology evolves and assumes new perspectives,DevOps has been characterized and marketed as the nexttransformation drive in the IT profession. DevOps is amethodology that is based on the facts of a particularcompany and industrial application (Contributor, 2014).DevOps is in demand across all industries because of itscapacity to foster collaboration. DevOps has gained a lot oftraction in the manufacturing industry, which is a large fieldof applicability. DevOps fits into space since the industry issurrounded by interconnected processes and activities.Furthermore, the manufacturing market has been steadilygrowing due to rapid delivery and innovation, which hasfueled integrated development and operations.DevOps requires enhanced cooperation and simplifiedcommunication, which are the cornerstones and foundationsof manufacturing (Veritis, 2016). DevOps is booming in avariety of sectors, and the manufacturing industry is benefiting from three of them: innovation, automation, andcollaboration. The combination of the three elementsresulted in improved communication and output in thesector. Furthermore, the teams work better together towardthe same goal, resulting in faster product delivery and lessmarketing time (Humble and Farley, 2011).

A.

Collaboration

According to studies, a lack of teamwork has led to thedownfall of numerous businesses, particularly in theautomobile sector. Collaboration is an organization'scultural intuition, and every entity looks forward toachieving such levels with ease. However, the process is notas simple as some may believe, which is where DevOpscomes into play (Veritis, 2016). DevOps is a term that refersto the integration of a company's development andoperations teams. The majority of the automobilemanufacturing businesses rely on silos to keep track of the jobs they need to complete. DevOps, on the other hand,helps to bridge the gap between the Ops and Dev teams bymaking them aware of their roles (Bird, 2016). As they aretied up with testing and preparing the items formanufacturing, this results in a quality improvement atevery stage of development. SIEMEN is the greatestexample of a manufacturing company that has successfullytapped into good teamwork.

B.

Automation

Automation is the use of technology to complete tasks withless human intervention. Automation aids in the speeding upof processes and scaling of environments, as well as the building of CI/CD workflows (continuous integration,continuous delivery, and continuous deployment). Themajority of machinery is automated to guarantee that thingsare produced and delivered quickly to fulfill customer needs(Bird, 2016). The automation process, on the other hand, isnot addressed. Automation has aided the industrial industryis gaining momentum on previously provided solutions totime-consuming tasks. Manufacturers, on the other hand,should embrace DevOps to achieve greater robustness inthis area. To see success with DevOps, industries mustfollow the four installation steps, which include reducinginefficiency, testing, deploying, and operating. Enhancedautomation is a business that has taken advantage of theseautomation tools. Inefficiency steering and testing have been used by the industry to shift their operations from a procedural to a product-based setting. As a result of theoptimized development, the production has altered (Humbleand Farley, 2011). To speed up the development process,they've used automated deployment and operation as adriving factor. Finally, this has facilitated openness, ease ofdeployment, and rapid market release of deliverables.

C.

Innovation

To lessen the pressure that comes with developmentcycles, industry have assured that quality and output levelsare optimal. As a result, an environment receptive tocreativity has emerged. Innovativeness has given theautomotive industry a competitive advantage in the market.DevOps has aided industries in achieving the maximumdegree of innovation (Humble and Farley, 2011). DevOpshelps businesses to minimize the amount of time spent on production or activity execution, allowing them to makemore informed decisions by transferring processes to teams.The approach of traditional developers differs in severalways, one of which is based on coding. Initially, industrialdevelopers used a variety of application features andcombined them at the end of the coding process. When bugsand flaws appear, the developers must complete the entirework to correct them. As a result, there have beenunnecessary delays in manually detecting bugs throughintegration. Through CI/CD phases, DevOps has brought asignificant shift in the sector, where developers releaseupdated code at a comparatively faster pace and with a highfrequency. Developers can sample and cross-check code intiny chunks and do a continuous compilation with such activities (Bob, 2016). For frequent code checks, the testing,security, and UI components are typically automated, andit's time to go to the repository. Naturally, this leads toearlier problem identification and faster bug resolution,increasing developer productivity. The supply ofdeliverables becomes continuous after the deployments have been handled through pipeline release.III.

T

HE

D

EV

O

PS

L

IFE

C

YCLE

The DevOps Lifecycle is a cycle of continuousimprovement and self-evaluation that ensures tasks arecompleted as fast and effectively as possible, both in termsof developing a quality product and maintaining andupgrading it. DevOps brings together the development andoperations teams to achieve optimal efficiency. The idea ofintegrating development and operations personnel helpsDevOps achieve its aims through enhancing communicationamong all stakeholders from planning to delivery, as well asautomating the delivery process in a variety of methods.This will benefit organizations:



Increase the frequency of deployments



Reduce the time it takes to get a product to the market.



New releases have a lower failure rate.



Reduce the time between fixes.



Increase the average time to recovery.This is performed by following and implementing a simplelifecycle of activities:



Continuous Development



Continuous Integration



Continuous Testing



Continuous Monitoring



Virtualization and ContainerizationDevOps promotes a culture of continuous everything. As aresult, there is a considerably shorter iterative cycle ofcontinuous improvement with much shorter update cycles,which reduces time to market while making it much easierto find and rectify flaws. While you repeat the cycle, you'llhave a good method to follow when implementing yourDevOps endeavor. Each team should be well-equipped withopen tools and guidelines before beginning work at anylevel. The tools can be tailored to the developer's needs andobjectives (Bob, 2016). As a result of the technique, high-quality, dependable software can be produced quickly. Thecycle is as follows as shown in figure 1:Figure 1: The DevOps Life CyclePlanning, the building, integrating and deploying,monitoring, operating, and responding through feedback passage are the lifecycles.a.

PlanningA problem statement and a scope description are used tospecify the needed resources that will be used during the project. Second, the developers give an overview of themost recent systems and their goals. Finally, they analyzethe study's practicability and feasibility to calculate howlong it will take to complete. Fourth, the team thinks aboutthe system's possible dangers, hazards, restrictions, security,and integrations. Finally, they create a project-widefeasibility study (Tutorialspoint, 2012). b.

BuildingThe developer creates system designs and converts the SRSdocument into a logical component with full requirementsfor coding execution once the initial stage of project planning is completed and authorized. The construction ofcontingencies, team training, and maintenance via anoperating plan comes next. The papers are sent to theimplementation level after the design is complete. Codingwith a certain source code and programming language isrequired for implementation. For error and defect detection,the system elements are coupled in a stressful environment(Tutorialspoint, 2012). After the system has been integrated,a test report is created by allocating resources.c.

MonitoringDevOps chips in during this phase, primarily throughout themaintenance and support lifecycle, to detect the many problems and bugs in the system. Traditionally,maintenance tasks were carried out under the ongoingobservation of users or developers. They madeimprovements that allowed software to go through a definedtime of testing or put in place reasonable criteria once asystem was up and running. After the testing step, themethod tackles lingering errors and resolves them manually.Today, DevOps steps in to solve the problem by developingsimple scripts that can virtually monitor the system, detectexternal components, and raise alarms. The alerts aretriggered and addressed automatically, or they recommendthat the developers and operations team take the appropriateactivities to resolve them. As a result, the maintenance andsupport procedure is automated and does not require physical interaction. This allows the engineers to spend theirefforts on more productive tasks (Crispin and Gregory,2015).d.

DevOps Process FlowThe teams have had moments since the introduction ofDevOps into the development cycle. Because DevOps is thefuture of IT operations, it's vital to understand the many procedures that create flow and the best way to apply it.DevOps' various principles serve as a practical and cultural basis for businesses. While developers focus their efforts onthe fundamental concepts of automation, collaboration, anditeration, they also strive to enhance the system regularly(Duffy, 2015). The developers and operations team test thesystem regularly and learn from their previous code issues

https://doi.org/10.5281/zenodo.6376787 International Journal of Computer Science and Information Security (IJCSIS),Vol. 20, No. 2, February 2022https://doi.org/10.5281/zenodo.63767874https://sites.google.com/site/ijcsis/ISSN 1947-5500

and failures. They act quickly on system feedback andrecommendations to improve their level of performance.This reduces the amount of time, money, and effort requiredfor deployment.These concepts have been extended to include actions thatfocus on automation and tool supply for rapid deploymentin lean-agile frameworks. By adopting the agile approach,which focuses on software integration, iteration, delivery,and deployment, the automation process has enabledoperators and other IT professionals to put their attentioninto smooth procedures. These are only a few of the actionsthat add to the cooperation process during the pipeline'sdevelopment.e.

DevOps BenefitsThe industrial platform has benefited greatly from DevOps.Most sectors and corporations, notably in the United States,have discovered the benefits of DevOps via the fastadoption of methodology and technology in theirdevelopment cycles For example, Enhanced Automation,located in the United States, has eliminated the usualdifficulties associated with sluggish IT operations(Opsgenie, 2015). Through agile advantages, the companyhas solely focused on automation, collaboration, andflexibility. These companies have gained the followingadvantages: first, they spend less time selling their productsor services since they are up-to-date and answering real-time problems. As a result, they can provide theircomponents to the market at a faster rate.Secondly, this quality of the return on investment has vastlyimproved with time. Finally, customers have expressed highlevels of pleasure as a result of receiving services and products in real-time as a result of speedier delivery(Opsgenie, 2015). Fourth, excellent operational efficiencyhas been achieved through reduced operating time and littleexpense and effort. This is made possible via automation.Fifth, improved cooperation has opened a space fordevelopers and IT operators to use simple codes to tacklecurrent challenges. Finally, these codes aid in the rapiddiscovery and correction of faults and issues. Thesedevelopment teams anticipate changes in processes andculture in real-time, reducing the risk of misinformation and process misalignment. Efficiency and product qualityimprove with clear and consistent communication. Regularintegration, deployment, and testing also aid in the creationof faster processes and the discovery of errors.f.

DevOps ImplementationThe process of implementing DevOps in a business appearsto be cumbersome and intimidating at first. This is becausethe operation necessitates both procedural and culturalchanges. When a company decides to embrace DevOps, itmust examine the steps that will be implemented gradually(Bass et al., 2015). One by one, the phases should beimplemented. Depending on an organization's existing position, management may consider using the agiletechnique. The following are the phases in the sequentialimplementation process: initially, the company must adoptan agile development model. Second, think about movingtheir infrastructure, software, and platform to the cloud.This entails moving away from the conventional reliance oncloud-based service delivery (Bass et al., 2015). It alsoeliminates the need for data centers and silos because theorganization's processes are available in real-time. Finally,the organization's activities must be adapted to the CI/CD pipeline approach. Fourth, they should automate theirsoftware deployment process to eliminate human contactsand improve market product delivery speed. Finally, theyshould automate software testing to detect faults and problems in code. Finally, after installation, they mustassure continuous deployment.Organizations must be aware that automated DevOpsnecessitates changes in tooling and infrastructure. If theorganization lacks the necessary tools and infrastructure toenable the DevOps process flow, it risks generating gaps.As a result, a business must use an agile approach tointegrate and automate all stages of the DevOps process tocreate a legitimate DevOps process (Bass et al., 2015).Additionally, a company may consider using graphics toestablish DevOps procedures, staff, training, and processdeadlines. This aids in the implementation process byensuring that everyone is on the same page.IV.

D

EV

O

PS

T

OOLS

Various tools must be implemented into the IT field'sdevelopment and operating procedures in terms of thinking.As a result, while implementing DevOps, the company mustconsider different devices. Git is a basic solution based ondistributed source code that allows developers and operatorsto follow the progress of their projects. Other componentslike workflow choices, staging regions, and branchingenvironments can be used to conveniently travel back andforth between distinct laws. Lucidchart is a visual tool thathelps developers and IT professionals design quick-to-implement yet easy-to-understand procedures and data.Furthermore, during the beginning stages of DevOpsdevelopment, the team must be taught and givendocumentation to track the development progress and keepall members informed (Lucidchart, 2015).Containerization has often switched to the grouping ofobjects into different units and clusters for automation anddelivery (Kar, 2015). Kubernetes is the most widely usedDevOps tool and open-source solution for assistingdevelopers in moving their projects to the next level. Theseare only a few of the tools used in the DevOps development process, which includes many others. Nonetheless, toolmanagement remains a challenge for IT engineers.Lucidchart resolves any challenges that arise throughout thedevelopment process, regardless of the device used in theapplication.V.

FUTURE

Various software automation trends might be equipped tomanage the industry's current software and technology ifadequately investigated by DevOps. Industry machines will become autonomous as a result of new technologies andautomated software. The key challenge with autonomousequipment is DevOps' capacity to persuade executives, operators, and drivers that these self-driving robots areaccurate and safe to ride or run (Abouzaid, 2016). Nonetheless, the whole DevOps process will be modernizedin the future through extensive automation, cooperation, andintegration to accelerate rapid delivery.VI.

C

ONCLUSION

Before understanding the meaning and purpose of DevOps,one must first understand their industry and position. Somesystems need a high level of automation and support, whileothers do not necessitate complicated processes. Thesemethods, on the other hand, necessitate application andrequirement alignment, rapid development, and novelinnovation testing. DevOps can assist release small batchesfor market effectiveness and efficiency after the industry hasa consistent resource to the market.

Author Details:

Ogala Justin Onyarin

Address all correspondence to:

justinoo2001@gmail.comOgala Justin OnyarinDepartment of Computer Science, Faculty of Computing,University of Delta, Agbor, Delta State, NigeriaR

EFERENCES

[1]

AbouZaid, A. (2016). Continuous delivery and maturitymodel: DevOps. Retrieved September 18, 2019, fromhttp://tech.aabouzaid.com/2016/01/continuous-delivery-and-maturity-model.html[2]

Armstrong, S. (2016).

DevOps for networking: Boost your organization's growth by incorporatingnetworking in the DevOps culture

. Birmingham, UK:Packt Publishing.[3]

Atlassian. (2015) DevOps: Breaking the DevelopmentOperation Barriers. Retrieved February 26, 2021, fromhttps://www.atlassian.com/devops\[4]

Bass, L., Weber, I., & Zhu, L. (2015).

DevOps: A software architect's perspective

. Old Tappan, NJ:Pearson Education.[5]

Bird, J. (2016).

DevOps sec: Delivering secure software through continuous delivery

. Sebastopol, CA:O'Reilly Media.[6]

Bob, R. (2014) The DevOps Life Cycle. RetrievedFebruary 26, 2021, from https://devops.com/the[7]

Contributor. (2014) DevOps with purpose: It's aboutyour application--retrieved February 26, 2021, fromhttps://devops.com/devops-with-a-purpose[8]

Crispin, L., & Gregory, J. (2015).

More agile testing: Learning journeys for the whole team

. Boston, MA:Addison-Wesley.[9]

Duffy, M. (2015).

DevOps automation cookbook: Over120 recipes covering essential automation techniques

.Birmingham, UK: Packt Publishing.[10]

Google. (2014) 2014 State of DevOps Report.Retrieved February 26, 2021, fromhttps://services.google.com/fh/files/misc/state-of-devops-2014.pdf[11]

Gruver, G. (2016).

Start and scaling DevOps in theenterprise

. Pennsauken, NJ: BookBaby.[12]

Humble, J., & Farley, D. (2011).

Continuous delivery

.Boston, MA: Pearson Education.[13]

Kar, S. (2015) Best 2014 DevOps tools and trends thatdefine DevOps' future in 2015. Retrieved February 26,2021, from https://siliconangle.com/2015/01/13/best-2014-devops-tools-and-trends-that

[14]

Lucidchart. (2015) Understanding the DevOps processflow. Retrieved February 26, 2021, fromhttps://www.lucidchart.com/blog/devops-process

J.[15]

Meena, S. (2014) Configuration Management.Retrieved February 26, 2021, fromhttps://www.slideshare.net/sahilsk/configuration-management-stackexpress-20140610

J.

[16]

Microsoft. (2013) 32 principles and practices ofsuccessful continuous Integration, ContinuousDelivery, and DevOps. Retrieved February 26, 2021,from http://davidfrico.com/devops-principles.pdf[17]

Opsgenie. (2015) Modern incident management foroperating always-on services. Retrieved February 26,2021, fromhttps://www.atlassian.com/software/opsgenie/what-is-opsgenie?[18]

Tutorialspoint. (2012) System Development Life Cycle.Retrieved February 26, 2021, fromhttps://www.tutorialspoint.com/system\_analysis\_and\_design/system\_analysis\_and\_design\_development\_life\_cycle.htm[19]

Veritis. (2016) DevOps Implementation inManufacturing Sector: Meet the Culture-DrivenApproach. Retrieved February 26, 2021, fromhttps://www.veritis.com/blog/devops-implementation-in

[20]

Whittle, D. (2014) An Introduction to DevOps.Retrieved February 26, 2021, fromhttps://devops.com/introductiontodevops

https://doi.org/10.5281/zenodo.6376787 International Journal of Computer Science and Information Security (IJCSIS),Vol. 20, No. 2, February 2022https://doi.org/10.5281/zenodo.63767876

<https://sites.google.com/site/ijcsis> ISSN 1947-5500