



MANAGING CONTINUOUS DEPLOYMENT OF A CLOUD HOSTED APPLICATION USING SYNCHRONIZATION TECHNIQUE

Jumbo Promise¹ & Umejuru Daniel²

¹ Department of Information Technology, University of Port Harcourt, Choba, Nigeria

² Department of Computer Science, University of Port Harcourt, Choba, Nigeria

E-mail: jumbo.promise@uniport.edu.ng¹, <https://orcid.org/0009-0007-6921-0419>

daniel_umejuru@uniport.edu.ng², <https://orcid.org/0009-0007-9843-2248>

D.O.I.: 10.5281/zenodo.19630308

ABSTRACT

In cloud computing research, management and continuous deployment of cloud applications has been a major challenge. Due to its computational complexity, many tools and techniques have been developed. In this work we present an efficient methodology called Rapid development methodology. This method is an enhancement of the existing method where feature flag approach was lacking. The new system identifies this approach as a way or technique to improve the existing system as it regards updating students' courses online and their results including carryover courses and missing scripts of which the existing system was unable to achieve. Programming Languages like Java Script (JS) and PhP were used as front end design and MySQL database as the back end design later in the research work in order to reduce time for new software features, advance the process, and eliminate potentially deployment errors as it regards online student's course registration and result checker.

KEYWORDS: Managing, Continuous Deployment, Cloud Hosted, Application, Synchronization, Technique.

INTRODUCTION

To achieve success in current global condition in which technologies, ideas and tools are continually changing and updated, information and other following factors must be correct, readily accessible, easily located and constantly delivered timely to every person that is intended. To automate software advancement process, practitioners uses current concept known as De-Ops which means "Development and Operations" (Humble, 2020) and this concept promotes continual connections between operation workers and developers through several rules, tools and practices that optimize delivering time of involved software. Particularly, foundation of De-Ops is its infrastructure usually in form of Code and it is technique to automation infrastructure usually on basis of practices from software creation. There are several cloud-based De-Ops processes that use capacities obtainable from Cloud Computing using Infrastructure Code to substructure computerization. Moreover, cloud-based De-Ops procedures help for continual delivering of infrastructure along with software apps which means cloud resources. Configurations handling



equipment like Ansible1, Puppet and Chef accomplished computerization of infrastructure that provisions in Cloud and each of these configuration tool has particular script language used in manual way to design infrastructure provisions in Cloud. Thus, the manual handling of these script semantics of these configuration tools is time-wasting and activity prone to several errors. (Phillips et al., 2023) Though these tools for configuration possess high automation level in infrastructure provisions, it still remains serious concern to computerize development procedure based on concept of continual delivering cloud resources.

The novel concept, De-Ops arise for handling of massive cloud-hosted apps. The combination word De-Ops means integration or combination of software design with operations widely speaking, its intent is to minimize time to market for novel software features and to remove potential costly placement errors. With increased competition in software market, firms pay notable attention and allot resources to design and deliver good quality software at increased pace. Continual Integration (CI), Continual Delivering (CDE) and Continual Deployment (CD) known as continual practices in this work are practices that are targeted at helping firms to increase their advancement and delivering software features with no compromising or given-up quality (Humble and Farley, 2024). Whilst CI means combining work-in-progress several times daily, CDE and CD possess capacity to quickly and dependably release values for buyers by creating computerization support as needed (Fitzgerald, 2023).

These practices are supposed to generate several benefits like creating more and faster feedback from customers processing platform, having recurrent and dependable releases which trigger enhanced buyers satisfaction with quality, through CD, connection between designing and operating teams is supported and manual operations could be removed (Chen, 2025). Several industrial situations revealed that continual practices are creating in-road in software design for industrial practices across several areas and firms not minding their sizes (Leppanen et al., 2024).

Also designing continual practices is not simple task because organization processes and tool might not be available to support complex and problematic nature of these practices, thus several researchers present several definitions of De-Ops (Andrej, et al, 2025) but it is seen generally as collection and integration of mutual reinforcing activities that responds to changing motivations of developing and operating teams. Developers look for faster change and chance to integrate novel software as quickly as obtained and possible while operating workers need more stable condition to handle risk “mechanical” like in continual placement

A De-Ops continual placement channel is computerized tool chain. When developers check in code, several steps occur with no manual intervention. The novel software is designed based on one form of tools and placed in test environment and provisioned by some other tools. Other tools activate these tests and when they are successful, the novel codes are released to stage and productions environ. De-Ops is aided by cloud computing and would be unthinkable with no infrastructure-as-a-service clouds (Len, et al, 2024).

2.0 Development of Cloud Computing

According to Danielson, cloud calculating is location non-dependent computing in which shared servers avails resources like software and data to PC and other connected devices based on demand (Danielson 2020). This is natural development of common adoption of virtualized service-based architecture and using computing. Details are hindered to consumers that don't need expertise over technology infrastructure "in the cloud" which aids them (Danielson, 2020).

The concept of 'cloud' is no longer current work in computing society but for several years; 'cloud' is employed as network chart to express internet or components of network that is handled outside environment of IT firms. Even though this concept started making more sense in recent years, some persons like John-McCarthy, an evolutionary computer expert suggested or supposed in his report in 1961 that "computation may someday be organized as a public utility". The concept didn't start at that time because hardware, software and other infrastructure were never available.

Cloud Computing is defined as type, pattern or form of computing which avails suitable network access to shared set of configurable resources that could be placed rapidly with high effectiveness and lower management handling (Mell and Grance 2023)

According to (Danielson 2020) Cloud Computing represent massive distributed computing concept or idea which is propelled by increasing economies of scale where sets of scalable, abstracted, virtualized, managed, storage, and service platforms are delivered based on needs and demands from of external buyer via internet.

(Danielson 2020) stated that "cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers" NIST utilized sketch to define cloud system sees figure 2.1 as a model which comprises of four placements; private, public, generic or hybrid and community (Dillon et al., 2022) and these four placement models explains levels of services that are given to cloud buyers. NIST also stated three service which includes "Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)" (Mathisen, 2021) and these are Service-based platform usually employed to determine services that are availed by cloud. Furthermore, NIST also noted using diagram several essence of Cloud Computing service (Vadym et al 2023) and its features are included as follows;

- i) Demand services
- ii) Wider network access
- iii) Resource pooling
- iv) Fact elasticity
- v) Calculated services (see figure 2.1)

Visual Model Of NIST Working Definition Of Cloud Computing
<http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html>

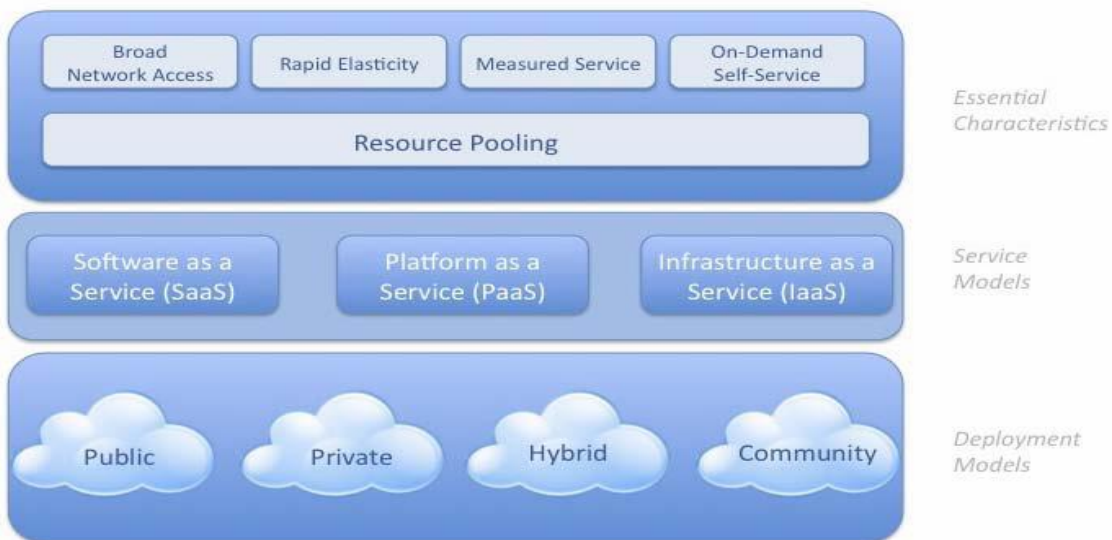


Figure 2.1: General Concept on cloud computing (Source: Mell and Grance, 2023)

Cloud platform provides several benefits. Selecting public or private platform implementation depend on usage, security, compliance requirements and other needs, but with suitable placement cloud platform could avail appreciable savings, good IT services, and high reliability level.

2.1 Cloud deployment model

These are four types of Cloud placement models

- Private
- Community
- Public
- Hybrid

2.1.1 Private Cloud

These are cloud platform owned and controlled by particular firm and could be handled by either Cloud Consumer firms or third party firms. They could be launched or hosted on premises of owner firm (private clouds site) or outsourced to some hosting firm outsourced private platform.

2.1.2 Community Cloud

Cloud platform which is shared among several firms and supports particular community concerns example mission, security needs policy and compliance needs. Government agencies, universities, banks etc. usually find these kinds of platform beneficial. Community platform equally have two different scenarios:

- On site platform scenario
 - Used by community platform enforced on premises of buyers composing community platform
- Out-sourced community platform
 - Used by community platform where server is given-out to hosting firms.

Examples of platform cloud:

- Google Application for Nation or States (Big data)
- Microsoft Community Cloud for governments (MGCC).

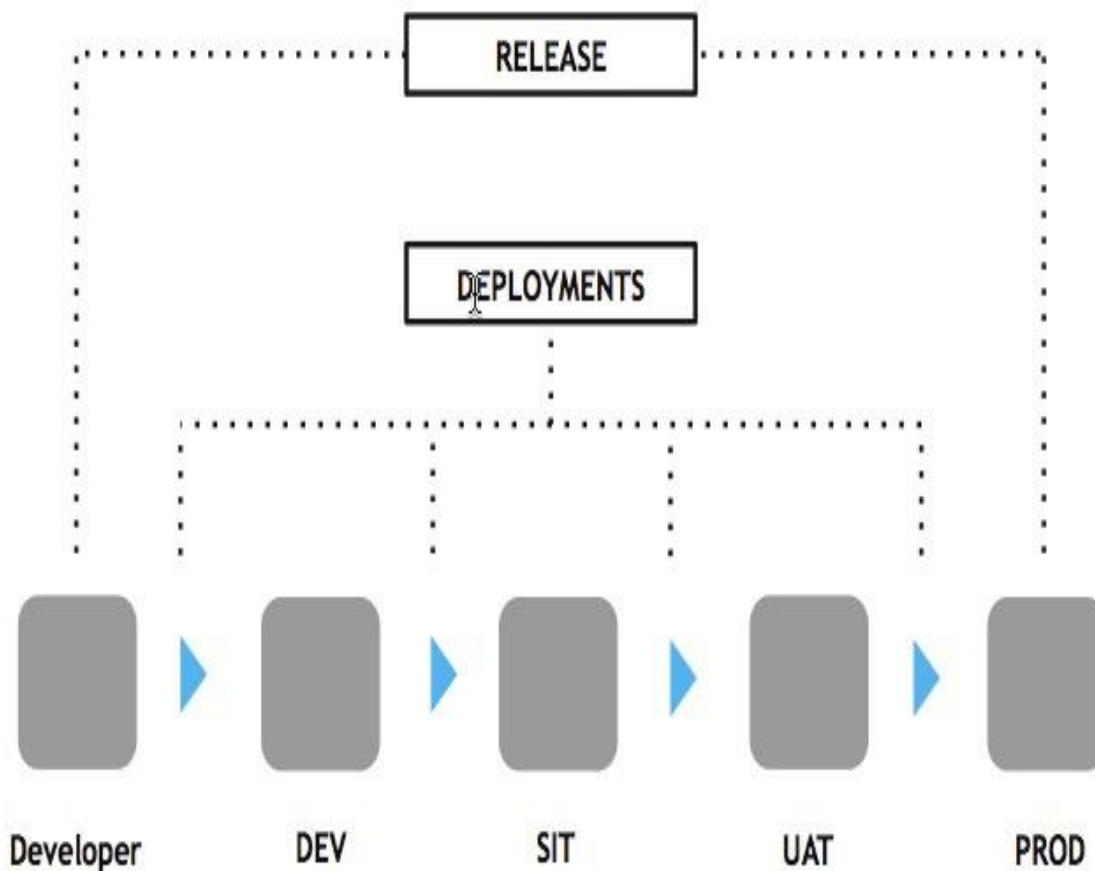


Figure 2.2: Illustration of Release and Deployment Mechanization (Source: Dillion, 2023)

3.0 Methodology

The method used in this work is Rapid application. This software development method uses recursive and object-based development technique for which entire system is grouped into subsystems and modules. RAD is used within short time periods like 60 to 90 days to deliver “functional system”, it is used in business, data and process modeling, application generation, testing and turnover. Since RAD process highlights on reuse, many program elements and interfaces are tested already which minimize time for test and development. The proposed system focuses on the use of feature flag in the developing a continuous software which the flag can either be turned OFF or ON. We will implement the feature flag concept using the Result Checking System (which comprises of students, course, departments, course registration e.t.c) such that some un-used feature that require an update will be turn OFF pending when the features is updated. The students (as a registered uses) login into the web application to check results after all courses attended by the students has been registered, and also such student partake in the examinations While the features manager can actually flag Off/On any component of the dataset from the Admin portal, and can also Edit/Detail/Delete any of the Students results database until they are ready with all the results online before the students can actually check their results. Feature flags are technique employed for software development which tries to present or generate alternative to maintain multiple-source-code branches in a way that every involved feature is tested before completion and release.

Feature toggles could allow short software incorporation cycles. The improved System architecture is shown in figure 3.1

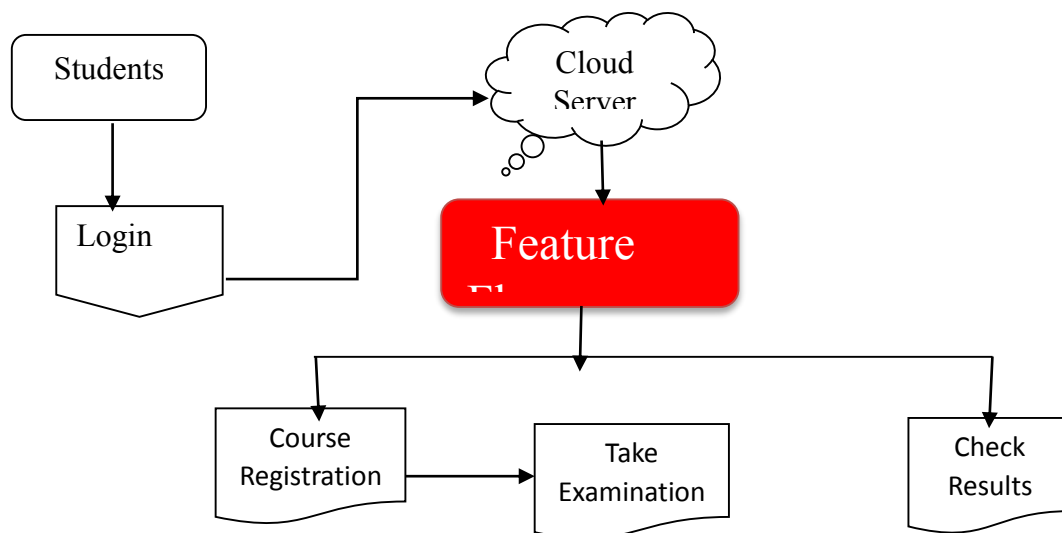


Figure 3.1: Proposed System Architecture



3.1 Explanation of the Proposed System Architecture

The proposed system focuses on the use of feature flag in the developing a continuous software which the flag can either be turned OFF or ON. We will implement the feature flag concept using the Result Checking System (which comprises of students, course, departments, course registration e.t.c) such that some un-used feature that require an update will be turn OFF pending when the features is updated. The students (as a registered uses) login into the web application to check results after all courses attended by the students has been registered, and also such student partake in the examinations While the features manager can actually flag Off/On any component of the dataset from the Admin portal, and can also Edit/Detail/Delete any of the Students results database until they are ready with all the results online before the students can actually check their results. Feature flags are technique employed for software development which tries to present or generate alternative to maintain multiple-source-code branches in a way that every involved feature is tested before completion and release.

3.1.1 Algorithm of the Proposed System

Algorithm: Modified Cloud Computing

Input: Students, Courses, results

Output: Grade (i.e A, B, C)

Method:

Step 1: Register Students

Step 2: Register Courses

Step 3: Add Lecturers

Step 4: Calculate Students' scores

Step5: Upload Results

Step 6: Check Results

Step 7: Manage Students

Step 8: Manage Courses

Step 9: Manage Results

Step10: If score ≥ 70 , then give A

Step 11: else

Step 12: Give B or C or D or F based on the output score

4.0 Result and Discussions

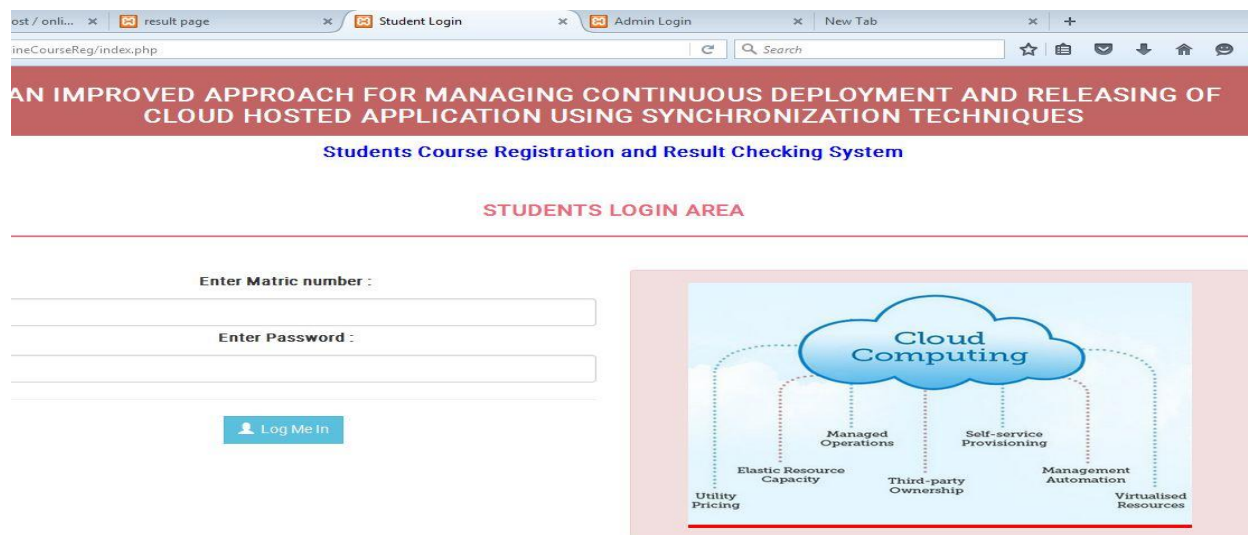


Fig. 4.1: Student's Login Page

Figure 4.1 shows the students login window, which gives information about the students, allows students register their courses, view their registered courses and check their results.

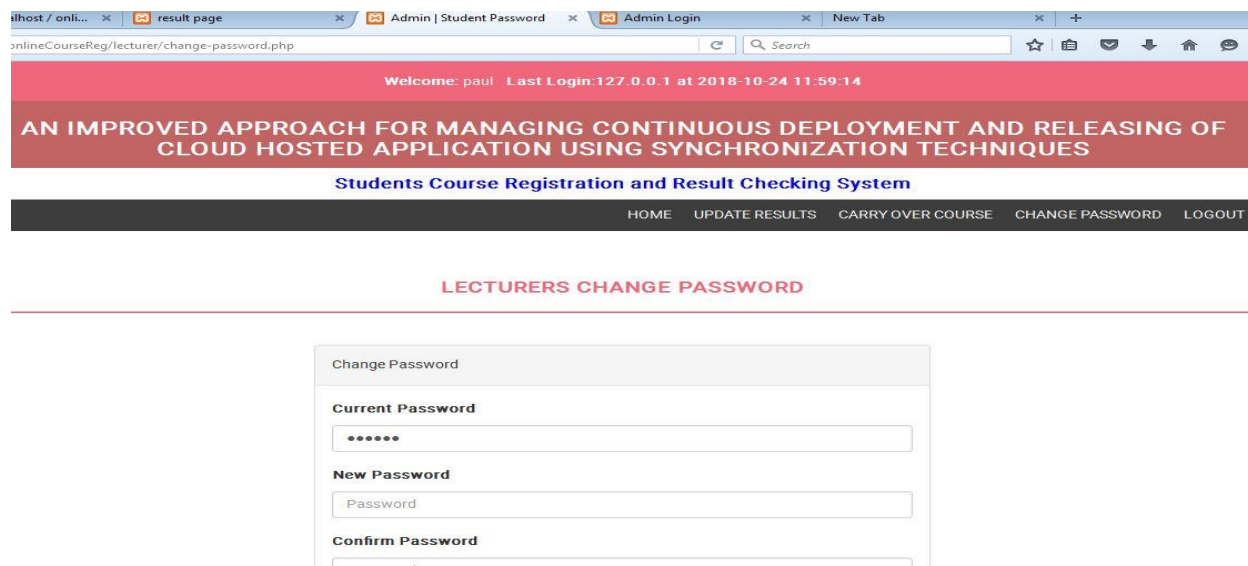


Fig. 4.2: Lecturer's Page

Figure 4.2 shows the lecturers page, where the lecturer can check courses offered by students, update results, Manage results and change password if need be.

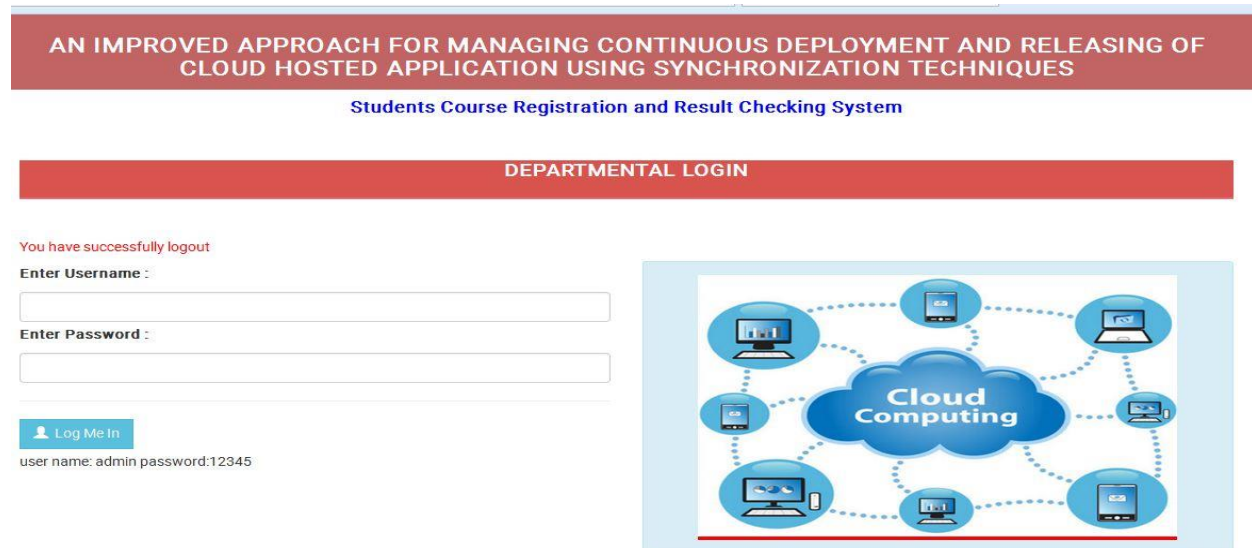


Fig. 4.3: Admin Login Page

Figure 4.3 shows the admin login window, where the department manages student's sessions and semesters, register students, manage registered courses, add lecturers and show student logs.

5.1 Conclusion

Cloud deployment management forms a major part of current firms and is intended to support handling activities, for a better decision making policy especially in student information and Management system. Cloud hosted application deployment and management attempts to improve the effectiveness of students online course registration and result checking especially accuracy, timeliness, quality information retrieval, synchronization and management. In this study we designed online course registration and students result checker by using feature flag approach which allows the department to manage students information, courses and results; update carryover courses when rewritten or update found missing scripts thereafter the system automatically calculates all the scores and give the required output as results. An application program was designed using the improved cloud computing algorithm for speedy implementation on a computer system.



Reference

- Andrej, Dyck, Ralf, Penners, and Horst Lichter, (2025) "Towards definitions for release engineering and DevOps," in Proceedings of the Third International Workshop on Release Engineering (RELENG), IEEE Press, Piscataway, NJ. 56-102
- Chen, (2025) —Continuous Delivery: Huge Benefits, but Challenges Tool, IEEE Software, vol. 32(2), 50-54.
- Danielson, M. H. (2020): Performance Evaluations of Symmetric Encryptions Algorithms. IJCSNS International Journals of Computer Science and Network Security, 8(12), 234-249.
- Dillon, S. A., Ayed, A. S.; and Norby, K. J. (2022): New Proposed Robust, Scalable and Secure Network Cloud Computing Storage Architecture. Journal of Software Engineering and Applications 2(1), 19-36.
- Fitzgerald, and Stol, J. (2023) —Continuous Software Engineering: A Roadmap and Agenda, Journal of Systems and Software, 123(1), 451-466.
- Humble, J. and Farley, D. (2024) Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, Addison-Wesley Professional. 1(1), 546-559.
- Len Bass, Ingo Weber, and Liming Zhu (2024) DevOps: A Software Architect's Perspective (1sted.), Addison-Wesley Professional, ISBN: 0134049845.
- Leppanen, M. Makinen, S., Pagels, M., Mantyla, V and T. Mannisto, (2024)—The Highways and Country Roads to Continuous Deployment, IEEE Software, 32(2), 64-72.
- Mathisen S. S (2021). Security Attacks and Cryptography Solutions for Data Stored in Public Cloud Storage. International Journal of Computer Science and Information Technology and Security. 6(1), 498-503.
- Mell P. and Grance T (2023). "Draft NIST working definition of cloud computing," Online at <http://src.nist.gov/groups/SNS/cloud-computing/index.html>. models of Cloud Computing. J Network Computation Application 34(1):1–11.
- Phillips A., Sens, M., De Jonge, A., and van Holsteijn, M. (2023) The IT Manager's Guide to Continuous Delivery: Delivering business value in hours, not months: XebiaLabs. 2(3), 231-225
- Vadym M. & Artem V. (2023). Security Risk Analysis for Cloud Computing Systems. The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 3(3), 340-347.