Graphical user interface, text, application

Description automatically generated

**SWE7103-2 DevOps: Project Report**

**DEVOPS IN LEAVE MANAGEMENT SYSTEM**

**Submitted To:**

**Dr. Naveed Islam**

**Submitted By:**

**Austin Orumwense 2214630**

**Blessing Peter 2214077**

**Davison Isesele 2212656**

**Toyin Samad 2215831**

**A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT FOR THE AWARD OF AN MSc. DEGREE IN CLOUD AND NETWORK SECURITY**

**January 2023**

***The University of Bolton***

*Deane Road, Bolton, BL3 5AB*

<http://www.bolton.ac.uk>

# TABLE OF CONTENTS

[TABLE OF CONTENTS 1](#_Toc124264890)

[LIST OF TABLE 2](#_Toc124264891)

[LIST OF FIGURES 3](#_Toc124264892)

[LIST OF ABBREVIATIONS 5](#_Toc124264893)

[ABSTRACT 6](#_Toc124264894)

[CHAPTER 1 DEVOPS IN LEAVE MANAGEMENT SYSTEM 7](#_Toc124264895)

[1.1 INTRODUCTION 7](#_Toc124264896)

[1.2 IMPORTANCE OF DEVOPS 8](#_Toc124264897)

[1.3 DEVOPS TOOLS 9](#_Toc124264898)

[1.4 CI/CD PROCESS 10](#_Toc124264899)

[1.5 OBJECTIVES 11](#_Toc124264900)

[CHAPTER 2 RESEARCH 12](#_Toc124264901)

[2.1 LITERATURE REVIEW 12](#_Toc124264902)

[2.2 EXISTING SYSTEMS 13](#_Toc124264903)

[2.3 OVERVIEW OF THE SYSTEM 14](#_Toc124264904)

[2.4 GANTT CHART 15](#_Toc124264905)

[CHAPTER 3 METHODOLOGY 17](#_Toc124264906)

[3.1 ARCHITECTURE OF DEVOPS AND TOOLS USED 17](#_Toc124264907)

[3.2 PLANNING 17](#_Toc124264908)

[3.3 CODING 17](#_Toc124264909)

[3.4 BUILDING & TESTING 19](#_Toc124264910)

[3.5 JENKINS 29](#_Toc124264911)

[3.6 CONTINUOUS DEPLOYMENT 37](#_Toc124264912)

[3.7 OPERATION 39](#_Toc124264913)

[3.8 MONITORING 43](#_Toc124264914)

[CHAPTER 4 RESULT AND DISCUSSION 45](#_Toc124264915)

[4.1 TASK DIVISION 48](#_Toc124264916)

[4.2 PERSONAL REFLECTIONS 49](#_Toc124264917)

[4.3 LEARNING REFLECTIONS 50](#_Toc124264918)

[CHAPTER 5 CONCLUSION AND FUTURE SCOPE 52](#_Toc124264919)

[5.1 FUTURE SCOPE 52](#_Toc124264920)

[5.2 CONCLUSION 54](#_Toc124264921)

[REFERENCES 55](#_Toc124264922)

[APPENDIX 57](#_Toc124264922)

# LIST OF TABLE

[Table 4‑1 Task Division 49](#_Toc124263941)

# LIST OF FIGURES

[Figure 1.1. DevOps tools 10](#_Toc124264595)

[Figure 1.2 CI/CD Pipeline 10](#_Toc124264596)

[Figure 2.1 Flow chart of the Leave management system 15](#_Toc124264597)

[Figure 2.2 Project Gantt chart 16](#_Toc124264598)

[Figure 3.1 Home page code on visual studio 18](#_Toc124264599)

[Figure 3.2 Figure About Us page code on visual studio 18](#_Toc124264600)

[Figure 3.3 Local Version control system (Scott & Ben, 2014) 21](#_Toc124264601)

[Figure 3.4 Centralised Version Control System 22](#_Toc124264602)

[Figure 3.5 Distributed version control system (Scott & Ben, 2014) 22](#_Toc124264603)

[Figure 3.6 local repository 24](#_Toc124264604)

[Figure 3.7 Updating Kali Linux apt repository 24](#_Toc124264605)

[Figure 3.8 switching to root user and installing software properties 25](#_Toc124264606)

[Figure 3.9 Figure adding git repository 25](#_Toc124264607)

[Figure 3.10 Installing git on Kali Linux 25](#_Toc124264608)

[Figure 3.11 Version of Git installed 26](#_Toc124264609)

[Figure 3.12 Git configuration 26](#_Toc124264610)

[Figure 3.13 GitHub Architecture. (Reshma, 2022) 27](#_Toc124264611)

[Figure 3.14 Git account for Group-1-A-F 28](#_Toc124264612)

[Figure 3.15 Group-1-A-F GitHub repository 28](#_Toc124264613)

[Figure 3.16 Group -1-A-F uploaded files in GitHub 29](#_Toc124264614)

[Figure 3.17 updating yum software package 30](#_Toc124264615)

[Figure 3.18 Adding Jenkins repository to yum software package 31](#_Toc124264616)

[Figure 3.19 Importing Jenkins GPG key 31](#_Toc124264617)

[Figure 3.20 Installing Java 31](#_Toc124264618)

[Figure 3.21 version of Java installed 32](#_Toc124264619)

[Figure 3.22 Installing Jenkins 32](#_Toc124264620)

[Figure 3.23 Enabling Jenkins to start at boot 33](#_Toc124264621)

[Figure 3.24 Starting Jenkins service 33](#_Toc124264622)

[Figure 3.25 Unlocking Jenkins 33](#_Toc124264623)

[Figure 3.26 Jenkins password 33](#_Toc124264624)

[Figure 3.27 Creating Jenkins admin user 34](#_Toc124264625)

[Figure 3.28 Jenkins homepage 34](#_Toc124264626)

[Figure 3.29 Installing GitHub plugin on Jenkins 35](#_Toc124264627)

[Figure 3.30 Configuring pipeline script from SCM and repository URL 36](#_Toc124264628)

[Figure 3.31 Configuring the branches to build 37](#_Toc124264629)

[Figure 3.32 Jenkinsfile script on GitHub repository 37](#_Toc124264630)

[Figure 3.33 Installing pip.py 38](#_Toc124264631)

[Figure 3.34 Installing Ansible 38](#_Toc124264632)

[Figure 3.35 Confirming the version of Ansible installed 38](#_Toc124264633)

[Figure 3.36 AWS EC2 Instance 40](#_Toc124264634)

[Figure 3.37 Apache server installation on EC2 instance 41](#_Toc124264635)

[Figure 3.38 PHP server installation on EC2 instance 41](#_Toc124264636)

[Figure 3.39 Installing AWS plugin 42](#_Toc124264637)

[Figure 3.40 Restarting Jenkins after AWS installation 42](#_Toc124264638)

[Figure 3.41 Monitoring the EC2 instance using the Nagios web interface 44](#_Toc124264639)

[Figure 4.1 Login homepage 46](#_Toc124264640)

[Figure 4.2 Administrator module 46](#_Toc124264641)

[Figure 4.3 Staff homepage 47](#_Toc124264642)

[Figure 4.4 Leave successfully approved 47](#_Toc124264643)

[Figure 4.5 Jenkins build a dashboard 48](#_Toc124264644)

# LIST OF ABBREVIATIONS

AWS AMAZON WEB SERVICE

EC2 ELASTIC CLOUD COMPUTE

CI CONTINUOUS INTEGRATION

CD CONTINUOUS DELIVERY

DEVOPS DEVELOPMENT AND OPERATIONS

HTML HYPERTEXT MARKUP LANGUAGE

PHP PHP HYPERTEXT PREPROCESSOR

VCS VERSION CONTROL SYSTEM

HR HUMAN RESOURCE

IT INFORMATION TECHNOLOGY

AI ARTIFICIAL INTELLIGENCE

# ABSTRACT

This project aims to build and execute a leave management system using DevOps principles and practices. The team's efficiency and production are guaranteed to increase if a culture of collaboration, continuous integration, and continuous deployment (CI/CD) is adopted. Team leaders will be able to approve or reject leave requests and keep track of team availability on the system's central platform, which will be available for team members to request, track, and manage their time off.

The project will begin by conducting a needs assessment to understand the current leave management process and identify pain points and opportunities for improvement. This will involve interviews with team members and managers, as well as a review of existing leave management systems and processes. Based on the assessment, a tool will be designed and developed to address the identified challenges and meet the requirements of the DevOps team. The tool will be implemented and tested in a pilot phase to gather feedback and make any necessary adjustments. Finally, the project will conclude with a comprehensive evaluation of the tool, including a comparison of the pre-and post-implementation leave management processes and an assessment of the tool's impact on team productivity and satisfaction. Overall, this project will demonstrate the benefits of applying DevOps principles to leave management and how it can lead to increased organizational efficiency and effectiveness.

Keywords: Leave management, CI/CD, Git, DevOps tools, DevOps principles.

# CHAPTER 1 DEVOPS IN LEAVE MANAGEMENT SYSTEM

## 1.1 INTRODUCTION

Leave management systems have become increasingly popular in recent years as a way for organizations to streamline the process of tracking and approving employee time off. A leave management system is a software application that helps organizations manage the leave taken by their employees, including vacation, sick leave, and other types of time off. The purpose of this project is to design and implement a leave management system for a large organization using a DevOps solution. The system will allow employees to request and track their leave, and it will also provide managers with the tools they need to approve or deny leave requests. The system will also provide reporting capabilities.

Leave management systems are software applications that help organizations manage the leave taken by their employees, including vacation, sick leave, and other types of time off. The goal of a leave management system is to streamline the process of requesting and tracking leave, the benefits of a leave management system include increased efficiency, improved accuracy, and better decision-making. Web-based leave management systems are particularly useful because they can be accessed from any device with an internet connection. This is especially beneficial for employees who are working remotely (Serdar, 2022). Novo Solutions emphasizes the importance of securing a leave management system, suggesting that organizations implement measures such as secure login protocols and data encryption to protect sensitive employee data and prevent unauthorized access (NovoSolutions, 2020).

Leave management systems should be user-friendly to ensure that they are easy for employees and managers to use. Integrating a leave management system with an organization's existing HR systems can be a challenge, but careful planning and coordination can ensure a smooth and successful integration (Ceridian, 2019).

## 1.2 IMPORTANCE OF DEVOPS

DevOps is a culture and practise in software engineering that strives to integrate software development (Dev) and software operation (Ops) (Kim, Behr, & Spafford, 2013). “The main goal of DevOps is to shorten the systems development life cycle and provide continuous delivery with high software quality” (Kim et al., 2016). Agile planning, continuous integration, continuous delivery, and application monitoring are all examples of DevOps approaches (Sharma, 2017).

One key aspect of DevOps is the collaboration between developers and operations teams (Kim et al., 2013). Teams can provide applications and updates more rapidly and reliably by cooperating and automating the process of creating, testing, and distributing software (Kim et al., 2016). “This is especially important in today's fast-paced business environment, where the demand for new and improved software is constant” (DevOps Institute, 2018). There are many tools and technologies that can be used to support DevOps practices, including version control systems, containerization platforms, and monitoring tools (Sharma, 2017). Some common DevOps tools include Git, Jenkins, Ansible, AWS EC2 and Nagios (Kim et al., 2016).

Overall, DevOps aims to enable organizations to deliver value to their customers faster and more reliably (Kim et al., 2013). By adopting DevOps practices, teams can improve their efficiency and agility, leading to increased competitiveness and success (DevOps Institute, 2018). In this project, we will be designing and implementing a leave management system for any organization. The system will be web-based, allowing employees to request and track their leave from any device with an internet connection. It will also be customizable, so that it can be tailored to the specific leave policies and rules of each organization.

One of the key goals of the project will be to incorporate DevOps into leave management system. Another important aspect of the project will be the integration of the leave management system with the organization's existing HR systems. This will require careful planning and coordination to ensure that the data is transferred accurately and efficiently between the systems. Ensuring the security of the leave management system will also be a key focus (Novo Solutions, 2020). The system will need to be designed with robust security measures in place, such as secure login protocols and data encryption, to protect sensitive employee data and prevent unauthorized access.

Finally, the leave management system will require the system to be highly configurable, so that it can be easily modified to meet the organization's evolving needs.

## 1.3 DEVOPS TOOLS

The goal of DevOps is to integrate software development and operations (Dev+Ops). It is a broad field that uses different tools at different times. Separate tools can be used to implement DevOps. Any tool you use will fully depend on the infrastructure and model you want to employ to develop a particular piece of software. There is no clear indication that one DevOps tool is better than another. It is simply because those tools are necessary for the business infrastructure or because the user prefers them. The tools we used in our project are listed below:

* Git Hub
* Jenkins
* Ansible
* AWS
* Nagios

Text

Description automatically generated with medium confidence

Figure 1.1. DevOps tools

In chapter 3, these tools will be thoroughly explained.

## 1.4 CI/CD PROCESS

Continuous Integration and Continuous Deployment are abbreviated as CI/CD. It is a strategy for software engineering that aims to accelerate the time between producing code and using it. After merging all code changes into a single repository, the Continuous Integration (CI) process automatically builds and tests the code. This makes it easier to identify integration problems as soon as feasible (Shahin et al., 2017). CD which is Continuous Deployment is the process of automatically delivering code updates to production after they have passed all tests. This lowers the possibility of human mistake throughout the deployment process and enables enterprises to deliver new features and upgrades more often (Zhang et al., 2018). Together, CI/CD aids businesses in delivering software upgrades more swiftly and trouble-free, enhancing operations.

A screenshot of a device

Description automatically generated with low confidence

Figure 1.2 CI/CD Pipeline

In the above figure, we illustrated how we implemented the CI/CD process starting from Planning – we gathered as a team to discuss the project overall. Coding – We decided to code with Microsoft Visual Studio. Building – here we used GitHub. Testing and Release – we used Jenkins (Continuous Integration tool). Deployment – we used Ansible (Continuous Delivery tool). Operation – we used AWS EC2 While Monitoring we used Nagios.

## 1.5 OBJECTIVES

This project aims to further improve the current leave management system by adding additional features and introducing DevOps tools and principles in such a way that it would be flexible for the administrative department and user-friendly to employees. Our proposed solution will:

* Improve communication between employees and manager
* Ensure collaboration in development and delivery
* Ensure continuous update and integration of new features
* Reduce overhead cost and absenteeism due to irregular leave patterns
* Ensure that error in calculating leave and attendance drops significantly
* Improve employee productivity and efficacy
* Ensure legal compliances
* Improve employee satisfaction and retention rate.

# CHAPTER 2 RESEARCH

Traditional project management views development as a continuous set of precisely specified actions. However, because the software development life cycle is so fluid and unpredictable, the traditional methodology did not work. “The term software engineering was coined at a NATO conference in Garmisch-Partenkirchen in 1968. Software development was seen to be driven based on the principles and practices seen in engineering” (Kurkela, 2020).

## 2.1 LITERATURE REVIEW

According to Lwakatare et al., DevOps is "A mindset substantiated by a set of practices to encourage cross-functional collaboration between teams, especially development and IT operations within a software development organization, in order to operate resilient systems and accelerate delivery of change" (Lwakatare et al., 2016).

Rao suggested a fingerprint-based attendance monitoring system for educational institutions, with one fingerprint sensor and one LCD screen strategically distributed around the building. This proposal's key benefit is that students can use a fingerprint sensor to record their attendance. When a student imprints his or her fingerprint on the sensor, the system periodically updates the system server with the student's attendance information. This is done in order to maintain attendance. The primary function of this system is to reduce the issue of manual labour, and the automatic attendance system saves time (Rao, 2013).

The leave management system that Adisa proposed in 2016 for Al-Hikmah University gives its customers the option to seek leaves by completing an easy online form. The leave request can be cancelled by the employee or leave manager even after acceptance, and there is an embedded feedback option. The system's shortcomings include the fact that the database size keeps growing as more data is added every day, the fact that on-premise servers can fail, and the fact that the system is often web-based and requires extensive testing to find any security flaws. By shifting the programme to the cloud and introducing large data storage and automation, these restrictions can be minimised (Adisa, 2016).

In 2021, Battina examined the methods for overcoming the difficulties that DevOps poses for software development. Her research included a detailed overview of the issues that can occur when utilising DevOps, as well as suggestions for how to solve them. It was stated that these difficulties may be reduced by managing risk efficiently and budgeting for the inevitabl.e difficulties associated with using DevOps. As a result, she concluded that “early DevOps security implementation can help with the quick detection of vulnerabilities and the correction of operational problems before they become an issue” (Battina, 2021).

Typically, software initiatives are difficult and uncertain. Agile project management aims to manage complexity and ambiguity by shortening the time between planning and execution, recognising that not all information is available at the time of planning, and emphasising innovation and learning. Change from traditional project management to agile project management to deal with complexity and unpredictability by relying on people and their creativity rather than on processes, moving from command and control to shared decision-making and self-management, and dealing with people's limitations rather than on processes. The Agile Manifesto's guiding principles serve as the foundation for agile software development (Dyba, 2014).

## 2.2 EXISTING SYSTEMS

All employees are impacted by the HR duty of managing employee leave requests. By means of employee handbooks or HR policy manuals, most organisations effectively communicate their clear leave policies to managers and staff. HR is responsible for overseeing the administration of managerial leave, keeping track of leave balances, and responding to questions from staff members and their managers regarding the status of and eligibility for their leave. Employees also struggle with filling out paper paperwork and chasing after approvers to get their leave approved. A relatively basic HR process costs employees, managers, and HR considerable sums of money in wasted time. The manual method of managing leaves of absence causes numerous issues in daily operations.

## 2.3 OVERVIEW OF THE SYSTEM

Our project seeks to eliminate all drawbacks of the existing leave management system with their tools, technology, and delivery. This is achieved by significantly improving the current leave management system by implementing DevOps tools and principles as well as by including new features and functionality in a way that is adaptive for the administrative department and user-friendly for the staff. The system will become better as a result of these added features by:

* Ensuring continuous collaboration
* Improved speed to market
* Continuous integration and delivery
* System stability
* Breaking down Silos
* Resource and cost reduction
* Increased performance
* Innovation and creativity
* Job satisfaction
* Fewer failures

The flow chart and decision tree in implementing the leave management system is described below showing a flow from employee login module to leave approval.

Diagram

Description automatically generated

Figure 2.1 Flow chart of the Leave management system

## 2.4 GANTT CHART

As part of the planning and execution of the project, the Gantt Chart is generated below. Each step is broken down into several tasks that team members complete either individually or collectively.

Timeline

Description automatically generated

Figure 2.2 Project Gantt chart

Using the Gantt chart we have successfully created a visual representation of our project schedule providing a clear and simple way to understand the relationship between tasks, their duration, and the dependencies between them.

# CHAPTER 3 METHODOLOGY

## 3.1 ARCHITECTURE OF DEVOPS AND TOOLS USED

The overall objective of this project is to use DevOps tools and culture in such a way that Continuous Integration and Continuous Delivery (CI/CD) would be possible. CI/CD a set of automated phases that developers can confidently concentrate on or rely on to quicken the development lifecycle of any application.

## 3.2 PLANNING

A plan is a future-focused strategy or set of actions used to accomplish a specific objective. In any development process, there must be a plan. This project is a group work that includes Austin Orumwense, Blessing Peters, Davison Isesele, and Toyin Samad (GROUP 1-A-F). We had a meeting on the 11th of November 2022 to discuss on the project overall. Our discussion was centred on following DevOps culture and its processes to develop an application. In our subsequent meetings, we decided the tools we would use at every phase of our development. Coding – Microsoft Visual Studio Code. Building and Testing – Git/GitHub (version control and code commit) and Jenkins (Continuous Integration). Release and Deployment – Ansible (Continuous Delivery). Operation – AWS EC2. Monitoring – Nagios.

## 3.3 CODING

Visual Studio Code is a common code editor in use used today. It is an open-source code editor designed by Microsoft and it runs on multi-platforms i.e., it is compactable with different operating systems such as Windows, Linux and macOSText

Description automatically generated

Figure 3.1 Home page code on visual studio

The tools needed by developers for their fundamental edit-build-debug cycle are combined with a code editor's simplicity of use in Visual Studio Code. It provides robust extensible architecture, lightweight debugging, broad code navigation, and understanding. It also offers lightweight integration with existing tools (Microsoft, 2022)

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3.2 Figure About Us page code on visual studio

In this project, the application's web page was structured using HTML 5, styled using CSS 3, and designed using PHP, a user-friendly server-side and programming language. We were able to create the logic and behaviour we required on the application's web page using PHP, JavaSript, which allows interaction between the server-side and user - making navigation easier, and MySQL, a database that was utilized in conjunction with PHP for data retrieval and updating.

## 3.4 BUILDING & TESTING

#### **3.4.1 GIT**

Git is a well-known and effective open-source version management programme that controls changes made to files as they are altered over time by various users. Git allows users to keep track of file changes and backup earlier iterations of data. Git maintains the differences between branches when managing many variations of a product or when changes are made by various people (Orviz Fernández, 2022)

#### **3.4.2 VERSION CONTROL SYSTEM**

Version control is a system that allows developers to record or save changes on a particular file or collection of files. The system keeps track of any changes made to the file(s), and any version can be recalled at any time.

A version control system is a type of software that assists the development team in effectively managing and tracking any changes made to the source code, as well as the details of who made what changes and when. “A separate branch is created for every contributor who made the changes, and the changes aren’t merged into the original source code unless all are analysed as soon as the changes are green signalled, they merged to the main source code. It not only keeps source code organized but also improves productivity by making the development process smooth” (Geegsforgeeks, 2022).

##### **3.4.3 BENEFIT OF VCS (VERSION CONTROL SYSTEM)**

In application development process, the benefit of Version Control System includes the following:

* + Collaboration - increases project pace of development by facilitating effective collaboration.
  + Product delivery - utilizes the collaborators abilities, productivity, and speed of product delivery through improved communication and support.
  + Access to files - through VCS, the project contributors can access, contribute, or make change regardless of their location in the world.

It allows developers to contributors to project simultaneously and there is a working copy available for each contributor (Geegsforgeeks, 2022).

##### **3.4.5 TYPES OF VERSION CONTROL SYSTEM** VCS are of three types:

* Local Version Control System
* Centralised Version Control System
* Distributed Version Control System

##### **3.4.6 LOCAL VCS**

This is the most basic type of version control system. It makes use of a locally stored database on a specific computer. Changes to files are preserved in the form of a patch. The most recent modifications to a file since the most current version are available in each patch set. If the local computer or its database is damaged, the modifications made to each patch set are lost because they are saved locally (Serengeti, 2020)

Diagram

Description automatically generated

Figure 3.3 Local Version control system (Scott & Ben, 2014)

##### **3.4.7 CENTRALISED VCS**

In centralised version control system, all file versions are maintained on a single server, and it allows team collaboration. Centralised version control system allows Collaborator can retrieve files from the serve, push files back to the server simultaneously and each collaborator's activities are share amongst the entire collaborators The problem or demerit of this system is downtime of the central server. If the server is down, it would be difficult for any team member to work on the project. In the case of accidental damage of the storage media, if no backup is done, all the project files will be lost (Serengeti, 2020)

Diagram

Description automatically generated

Figure 3.4 Centralised Version Control System

##### **3.4.8 DISTRIBUTED VCS**

In this system, each clone replicates the original data while the repository and its history are precisely replicated. If a server is damaged, any user's repository on computers working together through it can be moved and utilized to repair the server. It is quick, provides accurate modification tracking, and has higher dependability. One of the setbacks with distributed version control systems is that non-sequential file modification numbers make referencing difficult (Serengeti, 2020)

A picture containing text, screenshot, sign, outdoor

Description automatically generated

Figure 3.5 Distributed version control system (Scott & Ben, 2014)

#### **3.4.9 GIT ARCHITECTURE**

The GIT architecture has three layers in the local system. The layers include:

* Workspace
* Staging area
* Local repository

##### **3.4.10 WORKSPACE**

This is the first layer of the GIT architecture. The layer enables the alteration or modification of source code. It is usually created in the local system when a GIT project is installed and initialized.

##### **3.4.11 STAGING AREA**

In this second layer, after the source code has been modified to suit the intended project, the code is staged in this layer – staging area using the git command – ‘git add’. With git add, the code will move from the workspace to the staging area with a snapshot created. If there is any relevant modification in the workspace, git add command is used to update or sync the code in the staging area (Scott & Ben, 2014)

##### **3.4.12 LOCAL REPOSITORY**

This is the third tier of the GIT architecture. The local repository is resident in the local machine. Here, after all necessary modifications are made on the source code, the code is committed from the staging area by applying the command – ‘commit’ (Intellipaat, 2022)

Diagram

Description automatically generated

Figure 3.6 local repository

From the figure above, in the workspace, the command ‘git add’ is used - to add files to the git staging area. This is done after modification of the source code.

In the staging area, the command ‘commit’ is used. The command keeps track modifications or changes in the repository with a unique tag or identification.

From the local repository, git checkout is used - to switch branches.

##### **3.4.13 GIT INSTALLATION – LINUX**

In Linux, git installation can be done via the terminal by using the following commands:

* *sudo apt update* (This command will update the Kali apt repository)

Text

Description automatically generated

Figure 3.7 Updating Kali Linux apt repository

* *sudo su* (This command will switch user to root account – admin privileges)
* *sudo apt-get install software-properties-common*

A picture containing text

Description automatically generated

Figure 3.8 Switching to root user and installing software properties

sudo add-apt-repository ppa:git-core/ppa (This will add git repository)

A picture containing graphical user interface

Description automatically generated

Figure 3.9 Adding git repository

* *sudo apt-get install git* (This command will install git)

Text

Description automatically generated

Figure 3.10 Installing git on Kali Linux

* *git –-version* (To verify the git installation)

Graphical user interface, text

Description automatically generated

Figure 3.11 Version of Git installed

##### **3.4.14 GIT CONFIGURATION**

Once the installation is complete, the next step is to configure the username – this will be used to track changes.

* *git config --global user.name “group1af”*
* *git config --global user.email “group1af@gmail.com”*

Text

Description automatically generated with medium confidence

Figure 3.12 Git configuration

#### **3.4.15 GITHUB**

GitHub is a website and cloud-based service that allows users to store and manage their Git repositories, as well as collaborate with other users on projects. Git is a version control system that is used to keep track of file changes and coordinate work on those files by numerous persons. GitHub provides a web-based interface for users to manage their Git repositories, as well as tools for project collaboration and version control. It also offers features such as bug tracking, project management, and team communication. Many developers use GitHub to share and collaborate on open-source software projects (Danielle, 2022).

##### **3.4.16 GITHUB ARCHITECTURE**

The flow code between Git on a local system and Github on the remote server is shown in figure blow. In the flow, few commands are used at every stage of the execution process

.Diagram, timeline

Description automatically generated

Figure 3.13 GitHub Architecture. (Reshma, 2022)

From the figure above, git add - to add files to the git staging area.

* **Commit** - each commit in a local repository is tracked using the commit command, and each commit has an ID.
* **Git push** – uploading a repository from a local computer to a remote Github server.
* **Git** pull – retrieving a copy repository from the remote Github server to the local system
* **Git checkout** – utilized to switch branches
* **Git merge** – is used to combine two branches

##### **3.4.17 PUSHING REPOSITORY TO GITHUB**

Below is the necessary step to push the repository to GitHub:

**Step I**: Account creation - to make use of GitHub, the first thing to do is to create an account. In this project, we created an account and named it GROUP-1-A-F as seen in the figure below

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3.14 Git account for Group-1-A-F

**Step II:** Create repository – from the account page, click on repository and the click on new repository. For the purposes of this project, we named our repository **leavesystem.**

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3.15 Group-1-A-F GitHub repository

**Step III:** Add files, Commit and Push repository to GitHub – On the repository created – leavesystem, our source codes and other necessary files were added, commit, and pushed to GitHub.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 3.16 Group -1-A-F uploaded files in GitHub

## 3.5 JENKINS

Several processes in the software development process can be automated with the help of Jenkins, an open-source automation server. Developers may automatically build, test, and release their code changes with this continuous integration solution. Jenkins can be used to streamline continuous integration and delivery (CI/CD) workflows to automate processes including creating, testing, and delivering software. It is written in Java and can be run on most operating systems – Windows, macOS and other Unix-like operating systems (Mahir Mohmmad, 2021)

#### **3.5.1 JENKINS WORKFLOW**

In Jenkins, a workflow is a series of tasks or steps that are automated as part of a continuous integration and delivery process. These tasks can include building and compiling code, running tests, packaging applications, and deploying code to production environments.

A Jenkins workflow can be triggered in a number of ways, such as when code is pushed to a version control repository, when a new commit is made to a branch, or when a new pull request is opened. The workflow can be configured to run a series of tasks in a specific order, or it can be set up to run tasks in parallel.

One of the key benefits of using Jenkins for workflow automation is that it provides a visual representation of the workflow, including the status of each task and any errors that may have occurred. This makes it easy to track the progress of the workflow and identify any issues that need to be addressed.

#### **3.5.2 JENKINS INSTALLATION ON EC2**

In this project, we installed Jenkins as continuous integration tool. It was installed on our EC2 instance which we have already created above. Here are the steps to install Jenkins on an Amazon Web Services (AWS) EC2 instance:

**Step I:** On the EC2 instance, update the software packages with the command

* *Sudo yum update -y*

Text

Description automatically generated

Figure 3.17 updating yum software package

**Step II:** Add the Jenkins repository to the system package manager's list of repositories by running the following command:

* *sudo wget -O /etc/yum.repos.d/jenkins.repo* [*https://pkg.jenkins.io/redhat-stable/jenkins.repo*](https://pkg.jenkins.io/redhat-stable/jenkins.repo)

Text

Description automatically generated

Figure 3.18 Adding Jenkins repository to yum software package

**Step III:** Import the Jenkins GPG (Jenkins-CI) key to enable installation using the following command:

* *sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>*

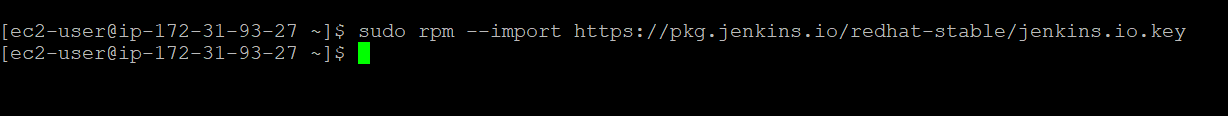


Figure 3.19 Importing Jenkins GPG key

**Step IV**: Jenkins was exclusively created in Java, therefore in order for Jenkins to function properly, Java must be installed on the base operating system using the following command:

* *sudo amazon-linux-extras install java-openjdk11 -y*

A computer screen capture

Description automatically generated with medium confidence

Figure 3.20 Installing Java

Graphical user interface, text

Description automatically generated

Figure 3.21 version of Java installed

**Step V:** Install Jenkins using the system package manager by running the following command:

* *sudo yum install Jenkins*

Text

Description automatically generated

Figure 3.22 Installing Jenkins

**Step VI:** Jenkins service can be enabled to start at boot with the command:

* *sudo systemctl enable Jenkins*

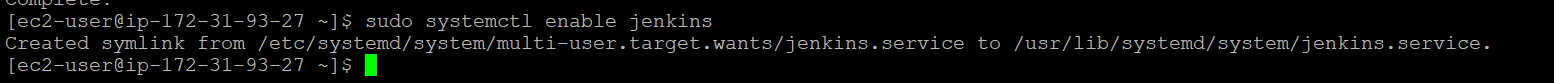


Figure 3.23 Enabling Jenkins to start at boot

**Step VII:** Start the Jenkins service using the following command:

* *sudo systemctl start Jenkins*



Figure 3.24 Starting Jenkins service

**Step VII:** To access Jenkins, open the Jenkins URL in a web browser. The URL is typically in the form <http://[ec2-instance-public-ip]:8080>. In our EC2, the URL is <http://ec2-44-211-124-172.compute-1.amazonaws.com:8080/>

Graphical user interface, text, application

Description automatically generated

Figure 3.25 Unlocking Jenkins

To unlock Jenkins with the administrator password, the following command is used:

* *sudo cat /var/lib/jenkins/secrets/initialAdminPassword*



Figure 3.26 Jenkins password

Graphical user interface, text, application

Description automatically generated

Figure 3.27 Creating Jenkins admin user

In the figure above, we created username and full name with GROUP1AF and GROUP1AF-LEAVESYSTEM respectively. Next, we saved and proceeded to Jenkins homepage

Graphical user interface, application, Teams

Description automatically generated

Figure 3.28 Jenkins homepage

The figure above is the first homepage of Jenkin. On the left-hand side, there are several options such as to create a new item, build history, manage Jenkins etc.

#### **3.5.3 JENKINS PLUGINS**

Once the installation is complete Jenkins plugins can be installed. Jenkins plugins are add-ons that extend the functionality of Jenkins. There are over 1000 plugins available for Jenkins, which provide integrations with a wide range of tools and services. Some common Jenkins plugins we used in this project include:

* GitHub plugin: Allows Jenkins to pull code from Git repositories.
* Amazon Web Services plugin: Allows Jenkins to run builds and tests on AWS
* Nagios plugin: Allows Jenkins to send notifications to Nagios, a popular open-source monitoring tool, when the status of a Jenkins builds changes. This can be useful for keeping track of the health of Jenkins builds and for integrating Jenkins with existing Nagios monitoring systems.
* Ansible plugin: Allows Jenkins to run Ansible playbooks as part of a build.

#### **3.5.4 INTEGRATING GITHUB WITH JENKINS**

To integrate Github with Jenkins, you will need to install the GitHub plugin in Jenkins. Here are the steps to do this:

Click the "Manage Jenkins" link from the Jenkins dashboard.

Click on the "Manage Plugins" link.

On the "Available" tab, search for "Git plugin" in the filter search box.

Select the Github plugin and click the "Install without restart" button.

Wait for the plugin to be installed, then click the "Restart Jenkins when installation is complete, and no jobs are running" checkbox.

Graphical user interface, application

Description automatically generated

Figure 3.29 Installing GitHub plugin on Jenkins

Click the "Restart Jenkins" button to apply the changes.

#### **3.5.5 CONFIGURING GITHUB ON JENKINS**

After the GitHub plugin is installed, you can set up a Jenkins job to pull code from a GitHub repository. Here are the steps to implement this:

Go to the Jenkins dashboard and click the "New Item" link.

Enter a name for the job and select "Pipeline" as the type.

Scroll down to the "Source Code Management" section and select "Git" as the repository type.

In the "Repository URL" column, enter the Git repository's URL.

Graphical user interface, text, application, email

Description automatically generated

Figure 3.30 Configuring pipeline script from SCM and repository URL

If the repository requires authentication, click the "Add" button next to the "Credentials" field and enter the username and password.

Specify the branches to build “\*/master”

Table

Description automatically generated with medium confidence

Figure 3.31 Configuring the branches to build

Under the "Build" section, add a build step to run a shell command or a script that builds the code.

Save the job and build it to test the integration.

Indicate the script path “Jenkinsfile” – this script will be available automatically on GitHub

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure 3.32 Jenkinsfile script on GitHub repository

## 3.6 CONTINUOUS DEPLOYMENT

Ansible is an open-source software tool that provides configuration management, application deployment, and task automation. It allows you to manage and control a large number of servers from a single machine, using a simple, human-readable language. With Ansible, you can automate tasks such as installing software, updating servers, and configuring applications and services. You can use Ansible to manage systems that run a variety of operating systems, including Linux, macOS, and Windows

#### **3.6.1 ANSIBLE INSTALLATION ON EC2 SERVER**

To install ansible the following commands are used:

* *python3 -m pip -V* (This command will verify the Python version installed)
* *python3 get-pip.py --user* (To install the latest python pip)

Graphical user interface, text

Description automatically generated

Figure 3.33 Installing pip.py

* *python3 -m pip install --user ansible* (This command will install ansible)

Graphical user interface, text

Description automatically generated

Figure 3.34 Installing Ansible

* *ansible --version* (To confirm the version of ansible installed)

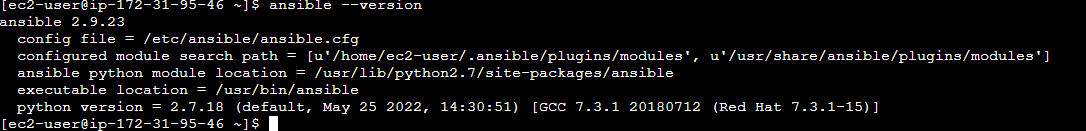


Figure 3.35 Confirming the version of Ansible installed

#### **3.6.2 INTEGRATING ANSIBLE WITH JENKINS**

Integrating Ansible with Jenkins can be useful if you want to use Jenkins to run your Ansible playbooks. Here's a general outline of the steps you can follow to set this up:

Install the Ansible plugin on Jenkins server.

In Jenkins, create a new project and choose the "Build a free-style software project" option.

In the "Source Code Management" section, specify the location of your Ansible playbooks.

In the "Build" section, add a build step and choose the "Execute Ansible playbook" option.

In the "Inventory" field, specify the location of your inventory file.

In the "Playbook path" field, specify the path to the playbook you want to run.

Save your changes and build the project.

This is just a basic example, and you can customize the Jenkins project to suit your needs. For example, you may want to add additional build steps, or set up Jenkins to run the playbook automatically when certain conditions are met.

## 3.7 OPERATION

Amazon EC2 (Amazon Elastic Compute Cloud) is a web service that offers resizable cloud computing power or compute capacity. Its goal is to simplify web-scale cloud computing for developers. With Amazon EC2, users can launch and manage virtual servers, called "instances," in the cloud. These instances can be configured to meet specific needs, such as running a particular operating system or using a certain amount of memory and CPU. Amazon EC2 allows users to scale their resources up or down as needed, giving them the flexibility to meet changing demands.

Amazon EC2 is part of the Amazon Web Services (AWS) cloud computing platform and infrastructure. It is used by a wide range of businesses and organizations around the world, including start-ups, enterprises, and government agencies.

As required by the project, we created Amazon EC2 instance and named it **GROUP1AF-LEAVESYSTEM** as seen in the figure below.

Graphical user interface, text, application, email

Description automatically generated

Figure 3.36 AWS EC2 Instance

On the EC2 instance server, Amazon Linux server was installed. We created key pair with the name group1afkey and security group with the name group1af-leave system. These two securities can be found on the navigation bar by the left under network and security.

#### **3.7.1 INSTALLING DEPENDENCIES ON EC2 SERVER**

On the EC2 instance we installed dependencies such as Apache web server, PHP, Jenkins and Nagios.

Apache web server installation

* *sudo yum install httpd -y* (This command will install Apache)
* *sudo service httpd start* (This command will start Apache service)
* *sudo service httpd status* (This command will check the status of Apache)

Text

Description automatically generated

Figure 3.37 Apache server installation on EC2 instance

PHP Installation

* *“sudo yum install -y amazon-linux-extras”*
* *“sudo amazon-linux-extras | grep php”*
* *“sudo amazon-linux-extras enable php7.4”*
* *“sudo yum clean metadata”*
* *“sudo yum install php-cli php-pdo php-fpm php-json php-mysqlnd”*

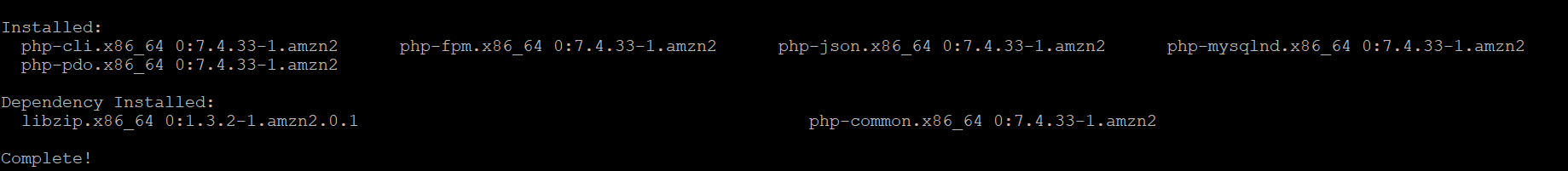


Figure 3.38 PHP server installation on EC2 instance

#### **3.7.2 INTEGRATING AWS WITH JENKINS**

With this plugin, you can use Jenkins to manage AWS resources such as EC2 instances, S3 buckets, and RDS databases, as well as deploy applications to AWS environments.

To install the AWS Plugin for Jenkins, follow these steps:

Go to the Jenkins dashboard and click on the "Manage Jenkins" link.

Click on the "Manage Plugins" link.

On the "Available" tab, search for "AWS plugin" in the filter search box.

Select the AWS plugin and click the "Install without restart" button.

Wait for the plugin to be installed, then click the "Restart Jenkins when installation is complete, and no jobs are running" checkbox.

Graphical user interface, application

Description automatically generated

Figure 3.39 Installing AWS plugin

Click the "Restart Jenkins" button to apply the changes.

Graphical user interface, text, application, Teams

Description automatically generated

Figure 3.40 Restarting Jenkins after AWS installation

#### **3.7.3 INTEGRATING AWS WITH JENKINS**

After the AWS Plugin is installed, you can set it up to access your AWS resources. To do this, you will need to configure the AWS credentials in Jenkins.

Here are the steps to configure the AWS Plugin in Jenkins:

Go to the Jenkins dashboard and click on the "Manage Jenkins" link.

Click on the "Manage Nodes and Cloud" link.

Click on the " Configure Clouds" link, add new cloud, select EC2

Scroll down to the "AWS" section and click the "Add AWS Credentials" button.

Enter the AWS access key and secret key and give the credentials a descriptive name. Save the changes.

## 3.8 MONITORING

Using Nagios, a free and open-source monitoring tool, administrators may make sure that all systems, programmes, services, and business operations are running smoothly. It is designed to be extensible and can monitor network devices, servers, and applications. Nagios can send notifications when it detects problems and can also alert administrators when it detects that a problem has been resolved. It is a powerful tool that can help organizations to ensure that their IT infrastructure is running smoothly and that any potential issues are detected and addressed in a timely manner.

#### **3.8.1 NAGIOS CONFIGURATION**

To integrate Nagios with an Amazon Elastic Compute Cloud (EC2) instance, you can follow these steps:

Launch an EC2 instance and install Nagios on it. You can use an Amazon Machine Image (AMI) that includes Nagios, or you can install Nagios on a fresh EC2 instance using a script or manual installation.

Install the necessary plugins and dependencies on the EC2 instance. Nagios uses plugins to check the status of different resources, such as servers, applications, and network devices. You will need to install the appropriate plugins for the resources you want to monitor.

Configure Nagios to monitor the EC2 instance. This involves creating configuration files that specify the resources to be monitored and the parameters for each check. You will also need to configure Nagios to send notifications when it detects problems or when problems are resolved.

Start the Nagios service on the EC2 instance. Once Nagios is configured and the necessary plugins are installed, you can start the Nagios service and begin monitoring your EC2 instance.

Graphical user interface

Description automatically generated with medium confidence

Figure 45 3.41 Monitoring the EC2 instance using the Nagios web interface

Nagios provides a web interface that allows you to view the status of your monitored resources, as well as any problems or alerts that have been generated. You can use this interface to monitor the health and performance of your EC2 instance in real-time.

# CHAPTER 4 RESULT AND DISCUSSION

When we applied the principles of DevOps to our project, which combined software development (Dev) and information technology operations, we discovered that delivering frequently in close alignment with business objectives and shortening the life cycle of systems development were two significant benefits.Some common benefits DevOps practices provided us include:

1. Continuous integration: our developers can regularly merge code changes into a central repository, which is then built and tested automatically.
2. Continuous delivery: our code changes are automatically built, tested, and prepared for deployment, allowing teams to release new updates quickly and reliably.
3. Continuous deployment: our code changes are automatically deployed to production, eliminating the need for manual intervention.
4. Infrastructure as code: our infrastructure is treated as code, allowing it to be versioned, reviewed, and managed like application code.
5. Monitoring and logging: real-time monitoring and logging of our application and infrastructure are used to identify and troubleshoot issues quickly.
6. Enhanced collaboration: our cross-functional team works together, with a focus on communication and collaboration between our developers and IT operations staff.
7. Better security: DevOps practices helps to ensure that security is built into the system from the start, rather than being an afterthought.

Using DevOps principles, we were able to improve the speed, reliability, and quality of software development and delivery, and respond more quickly to changing business needs.

Top of Form

Bottom of Form

**A picture containing text, indoor

Description automatically generated**

Figure 4.1 Login homepage

The figure above shows the login page where the user (or administrator) inputs the username and password to access the leave management system. The user information is being compared to the user list o the database to restrict access from unauthorised users.

Graphical user interface, text, application

Description automatically generated

Figure 4.2 Administrator module

The Administrator account has the priviledge to add and register all employees in the organization's database. Additionally, it has access to the Employee's information, which includes their leave status and type; Based on the employee, a search could be conducted, and a report could also be created.

**Graphical user interface, text, application, email

Description automatically generated**

Figure 4.3 Staff homepage

In this module, employee has the privilege to use his username and password to login into the system. He can then make leave request, view the status of existing requests, check the number of days remaining for leave and update his personal information.

Graphical user interface, text, application

Description automatically generated

Figure 4.4 Leave successfully approved

This automated system of leave management system empowers employees to check their own leave balances and apply them easily, thus it will increase employee satisfaction and affects the organization’s financial performance positively. Furthermore, this solution will introduce transparency, flexibility and convenience that will benefit the employees and the organization to achieve business goal overall.

**Graphical user interface, text, application

Description automatically generated**

Figure 4.5 Jenkins build a dashboard

The figure above reveals the project was successfully built and the build stability indicates no recent builds failed

## 4.1 TASK DIVISION

|  |  |  |  |
| --- | --- | --- | --- |
| S/N | STUDENT ID | NAME | TASK |
| 1 | 2214630 | AUSTIN ORUMWENSE | * Planning: Project research, requirement gathering * Operation: GitHub Setup, Ansible Installation, Monitoring * Report: Background study, result discussion and Conclusion |
| 2 | 2212656 | DAVISON ISESELE | * Planning: Project Research, Development Tools * Operation: AWS EC2 configuration, GitHub Setup, Visual Studio * Report: Project Methodology, Build & Development Tools |
| 3 | 2214077 | PETER BLESSING | * Planning: Project Research, Literature Review * Operation: GitHub Setup, Nagios Setup * Report: Project Introduction and abstract |
| 4 | 2215831 | TOYIN SAMAD | * Planning: Project Research, Build Tools * Operation: Web Development, Jenkins Installation, Instances Creation, GitHub Setup, Database Setup * Report: CI/CD Pipeline, Gantt Chart |

Table 4‑1 Task Division

## 4.2 PERSONAL REFLECTIONS

DAVISON ISESELE 2212656

My initial responsibility as a team member was involving myself during the planning phase, where I had the chance to contribute significantly. My understanding of the DevOps culture was broadened by participating in various phases of this project.

AUSTIN ORUMWENSE 2214630

Prior to embarking on this project, my knowledge of DevOps was basically theoretical, thinking it is a one-time implementation and basically a role. However, having participated in every stage of the project from planning to deployment and reporting, my mindset now has a cultural shift that DevOps is more of a philosophy and it involves a continuous process of improvement, it can be implemented incrementally, and it is a way to collaborate more efficiently.

PETER BLESSING 2214077

One of the most significant takeaways for me was the realization of how important collaboration and communication are when working in a development team. By working closely with my team members and keeping an open line of communication, we were able to overcome any obstacles that arose and deliver a successful project. I also gained a deeper understanding of the importance of Agile methodologies and DevOps practices in software development. By embracing an Agile approach and using the DevOps tools like Git, Jenkins, Ansible, Nagios and AWS EC2 we were able to move quickly and adapt to changes easily.

TOYIN SAMAD 2215831

A key aspect of the project was the use of agile methodologies, which emphasized the need for flexibility and the ability to adapt to changes quickly. This was particularly evident when it came to testing and implementing new features, as well as troubleshooting and resolving any issues that arose. The use of continuous integration and deployment tools such as Jenkins and Ansible helped to automate many of these processes and made the development process much more efficient. This experience is going to help me in my future career and help me to be a better team player and communicator.

## 4.3 LEARNING REFLECTIONS

AUSTIN ORUMWENSE 2214630

Working on this project has immensely broadened my knowledge on the DevOps culture, I have learnt how to handle a project right from the planning stage to delivery and then to reporting. I witnessed how DevOps tools work together in a cohesive manner to support continuous delivery and deployment of software. I have now learnt how crucial it is to work and collaborate within a team, learning a whole lot from each member that I wouldn’t have known working alone. Working with the team also taught me how to receive and accept criticism during a project delivery.

DAVISON ISESELE 2212656

The functionality of DevOps technologies in the CI/CD pipeline especially interests me. I gained a lot of expertise configuring AWS EC2 instance and exposure to various DevOps tools while working on this project.

PETER BLESSING 2214077

Throughout this DevOps leave management project, I have learned a great deal about the importance of collaboration and communication within a development team. One of the most significant takeaways for me was the realization that development and operations must work together seamlessly in order to ensure a successful project. I had previously thought of these two areas as separate entities, but this project made it clear that they are interconnected and rely on one another. This experience is going to help me in my future career and help me to be a better team player and communicator.

TOYIN SAMAD 2215831

Through this project, I have gained first-hand experience visualising the implementation of a DevOps CI/CD pipeline that includes all stages, from planning to monitoring, while utilising different tools and skill sets for each stage and working with team members to coordinate tasks to achieve success.

# CHAPTER 5 CONCLUSION AND FUTURE SCOPE

## 5.1 FUTURE SCOPE

The future of DevOps in a leave management system will depend on the specific needs and goals of the organization. However, some trends and developments in the field of DevOps that may impact a leave management system include:

1. Increased automation: As organizations continue to adopt DevOps practices, there may be an increase in the use of automation to streamline tasks and processes. This could include the use of tools such as Jenkins and Terraform to automate the build, test, and deployment process, or the use of machine learning to optimize the performance and availability of the leave management system.
2. Greater emphasis on security: As organizations become more reliant on software and digital systems, there will likely be a greater emphasis on security in DevOps practices. This could include the implementation of secure coding practices, the use of vulnerability scanners and penetration testing tools, and the adoption of secure infrastructure as code practices.
3. Increased use of cloud platforms: Many organizations are moving their applications and infrastructure to cloud platforms such as AWS, Azure, and Google Cloud. This trend is likely to continue, and DevOps practices will need to be adapted to take advantage of the unique features and capabilities of these platforms.
4. Greater focus on cultural and organizational changes: As organizations adopt DevOps practices, there will likely be a greater focus on cultural and organizational changes to support collaboration and communication between development and operations teams. This could include things like agile methodologies, DevOps training and certification programs, and the adoption of new roles and responsibilities within the organization.

Other futuristic features that can be incorporated into the leave management system include:

1. Integration with other HR systems: Leave management systems may be integrated with other HR systems, such as payroll and performance management, to provide a more seamless and efficient experience for users.
2. Predictive analytics: Leave management systems may use AI to analyze past leave patterns and predict future leave requests, allowing companies to better plan and allocate resources.
3. Virtual assistants: Leave management systems may incorporate virtual assistants that can handle tasks such as scheduling leave, answering employee questions, and providing real-time status updates.
4. Mobile apps: Leave management systems may be made available as mobile apps, allowing employees to request and manage their leave from their smartphones.
5. Automated approvals: Leave management systems may use AI to automate the approval process for leave requests, freeing up HR staff to focus on other tasks.

Overall, by combining DevOps practices with Artificial intelligence, the future of leave management systems is likely to focus on making the process more efficient, convenient, and user-friendly for both employees and HR staff.

**Top of Form**

**Bottom of Form**

## 5.2 CONCLUSION

We have successfully created a leave management system by using web-technologies and incorporating DevOps tools and principles. This web portal can be used by employees of any organization to easily manage all their leave-related work like requesting leave applications and getting notifications on whether their leave request is granted or not. Furthermore, it will provide both flexibility and convenience for the employees of an organization. Ultimately, the goal of using DevOps in a leave management system is to deliver a better product to users. By using the practices outlined above, it is possible to deliver a more reliable, stable, and user-friendly system, which can lead to increased user satisfaction.

The Leave Management System's operations are further improved by applying the DevOps approach. By automating the processes, the system will be more adaptable for the administrative department and user-friendly for employees in a business. Additionally, this solution will introduce simplicity, flexibility, and transparency, all of which will help the firm as a whole achieve its overall business objectives and benefit its people. The end goal of every project is to strive to improve performance, cut expenses, and create better software that fails less frequently; a DevOps strategy makes this more than possible.

# REFERENCES

Adisa, A. O. (2016). *Design And Implementation of An Electronic Leave Management System (A Case Study of Al-Hikmah University).* MSc. Al-Hikmah University, Ilorin, Nigeria

Battina, D.S., 2021. The Challenges and Mitigation Strategies of Using DevOps during Software Development. *International Journal of Creative Research Thoughts (IJCRT), ISSN*, pp.2320-2882.

Danielle Ellis, 2022. *Git vs GitHub: What’s the difference* [Online] Available from: <https://blog.hubspot.com/website/git-vs-github> [29 December 2022]

DevOps Institute, (2018). *State of DevOps Reports* [Online] Available from: <https://services.devopsinstitue.com/state-of-devops-report/> [31 December 2022]

Dybå, T., Dingsøyr, T. and Moe, N.B., 2014. Agile project management. In *Software project management in a changing world* (pp. 277-300). Springer, Berlin, Heidelberg.

Geeksforgeeks*,* 2022. *Version Control System* [Online]Available from <https://www.geeksforgeeks.org/version-control-systems/> [Accessed 29 Dec 2022]

Intellipaat, 2022. *Everything about Git* [Online] Available from <https://intellipaat.com/blog/what-is-git/#no4> [Accessed 31 December 2022]

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.* IT Revolution Press

Kurkela, M., 2020. *DevOps Capability Assessment in a Software Development Team.* KAMK University of Applied sciences

Lwakatare, L.E., Kuvaja, P. and Oivo, M., 2016*. An exploratory study of devops extending the dimensions of devops with practices.* *ICSEA*, *p104.*

Mahir Mohmmad, 2021. *Jenkins* [Online] Available from: <https://linkshortner.net/lEXEP> [Accessed 02 January 2023]

Microsoft, 2022. *vscode* [Online] Available from: <https://github.com/Microsoft/vscode/> [Accessed 14 December 2022]

Novo Solutions, (2020) *GIS Asset Management Software* [Online] Available from: <https://novosolutions.com/gis-asset-management-software-6-ideas-for-making-it-more-accessible/> [Accessed 31 December 2022]

Orviz Fernández, P., 2022. *Introduction to Git & GitHub* [Online] Available from: <https://digital.csic.es/bitstream/10261/266837/1/Git%26GitHub%20%283%29.pdf> [Accessed 29 December 2022]

Rao, S. and Satoa, K.J., 2013. An attendance monitoring system using biometrics authentication. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(4), pp.379-383.

Reshma Ahmend, 2022. *Git Tutorial – Commands and Operations in Git* [Online] Available from: https://www.edureka.co/blog/git-tutorial [Accessed 1 January 2022]

Scott, C. and Ben, S., (2014). *Pro Git 2nd Edition* [online] Available from: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control [Accessed 29 December 2022]

Serengeti., 2020. *Introduction to Git and types of Version Control Systems* [Online] Available from: <https://serengetitech.com/tech/introduction-to-git-and-types-of-version-control-systems> [Accessed 29 December 2022]

Shahin, M., Babar, M.A. and Zhu, L., 2017. *Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices*. *IEEE Access*, *5*, pp.3909-3943.

Zhang, Y., Vasilescu, B., Wang, H. and Filkov, V., 2018. One size does not fit all: an empirical study of containerized continuous deployment workflows. *In Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 295-306).

# APPENDIX

**About Us**

Graphical user interface, text

Description automatically generated with medium confidence

**Source Codes**

* Home page

Text

Description automatically generated

* About Us

Calendar

Description automatically generatedText

Description automatically generated