To build a V-Profile project.

A Project Work Synopsis

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE WITH SPECIALIZATION IN DEVOPS

Submitted by:

22BDO10012 Vishwachi Pandey

22BDO10022 Anand Tiwari

22BDO10025 Shivam

Under the Supervision of:

Ankur Gupta

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

Abstract

The architecture of the automated setup involves orchestrating the integration of various services and components through Vagrant scripts. Git Bash is employed as the primary command-line tool, and Visual Studio Code (VS Code) serves as the integrated development environment (IDE) for code development and collaboration.

The project's multi-tier nature provides a comprehensive environment for developers to work on different aspects of the application, from server-side scripting with PHP/Python/Perl, managing databases with MySQL, to handling web server configurations using Apache and NGINX. Additional components such as Tomcat, RabbitMQ, and Memcached are seamlessly integrated to support diverse functionalities and improve the overall performance of the application.

This comprehensive setup not only streamlines the local development process but also allows for thorough research and experimentation. Developers can use this environment to conduct R&D activities, test new features, and explore innovative solutions before deploying changes to the production environment. The inclusion of version control with Git ensures collaboration and version tracking, while Vagrant and VirtualBox provide a flexible and scalable infrastructure for local development.

In summary, this project establishes an automated, code-driven, and modular local environment for multi-tier web application development, empowering developers to conduct efficient research and development using a diverse set of services and tools. The combination of LAMP stack, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL within a WordPress-based application creates a robust platform for innovation and collaborative development.

Table of Contents

Title Page i

Abstract ii

1. Introduction

    1.1 Problem Definition

    1.2 Project Overview

    1.3 Hardware Specification

2. Problem Formulation

3. Research Objective

4. Methodologies

5. Experimental Setup

6. Conclusion

7. Tentative Chapter Plan for the proposed work

1. INTRODUCTION

In the dynamic landscape of web application development, creating a robust and efficient environment for research and development (R&D) is imperative. Our project revolves around addressing the challenges encountered in the development lifecycle of a multi-tier web application powered by a stack comprising Linux, Apache, MySQL, PHP/Python/Perl (LAMP), NGINX, Tomcat, RabbitMQ, Memcached, and MySQL, with the WordPress content management system.

The current development ecosystem presents hurdles related to the complexity of local setups, the non-reproducibility of environments, limited comfort in implementing real changes, inefficient R&D capabilities, and a reliance on manual configurations. Recognizing these challenges, our project aims to revolutionize the development workflow by introducing an automated, repeatable, and standardized local setup.

By leveraging tools such as Vagrant, VirtualBox, Git Bash, and Visual Studio Code (VS Code), our approach seeks to streamline the local development environment, making it accessible and consistent across various machines. This initiative not only enhances the ease of setting up individual components but also fosters collaboration, accelerates the development cycle, and promotes a more comfortable space for experimentation and innovation.

The project's multi-tier architecture, encompassing LAMP, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL within a WordPress framework, serves as a comprehensive platform for developers. It enables them to address the intricacies of server-side scripting, database management, web server configurations, and the integration of additional components seamlessly.

Through the establishment of an automated, code-driven local environment, our project empowers developers to conduct meaningful R&D activities. It facilitates the creation of isolated test environments, ensuring that changes are thoroughly tested before implementation in the production environment. The adoption of version control with Git enhances

collaboration, while the use of Vagrant and VirtualBox provides a scalable infrastructure for local development.

In summary, our project seeks to redefine the development experience by addressing existing challenges, fostering a more comfortable and standardized local setup, and providing developers with the tools and environment needed for efficient research, development, and innovation in the realm of multi-tier web applications.

1.1 Problem Definition

The current development environment for our multi-tier web application, powered by the LAMP stack, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL with WordPress, poses several challenges that hinder the efficiency and ease of the development process. The primary issues include:

1. Complex Local Setup:

The existing local setup for the project lacks simplicity and ease of use. Developers face challenges in configuring and establishing a consistent environment on their machines. The intricacies involved in setting up individual components such as Apache, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL make the process error-prone and time-consuming.

2. Non-Reproducible Environments:

The current local setup lacks repeatability. Developers often encounter difficulties in reproducing the same environment across different machines, leading to inconsistencies and variations in the development environment. This variability hampers collaboration and makes it challenging to ensure that all team members are working with identical configurations.

3. Limited Comfort in Making Real Changes:

Developers express discomfort when implementing changes directly in the production environment due to the absence of a robust local testing environment. The lack of a reliable

local setup hinders the ability to confidently experiment, test, and iterate on new features, ultimately slowing down the development lifecycle.

4. Inefficient Research and Development (R&D):

The current environment does not adequately support R&D activities. Developers struggle to conduct meaningful experiments and explorations due to the complexities in setting up isolated and controlled test environments. This limitation inhibits innovation and the ability to validate new ideas before deployment.

5. Dependency on Manual Configurations:

The current setup relies heavily on manual configurations, contributing to a lack of standardization and consistency. This dependency on manual interventions increases the likelihood of errors, creates a steep learning curve for new team members, and makes it challenging to scale the development process effectively.

In light of these challenges, there is a critical need to address the complexities in the local setup, enhance repeatability, and provide a more comfortable and standardized environment for developers to make real changes and conduct efficient research and development. The solution aims to streamline the development workflow, foster collaboration, and empower developers to work with confidence in both local and production environments.

1.2 Problem Overview

The current development environment for our web application, utilizing the LAMP stack, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL with WordPress, presents challenges in complexity and reproducibility. Setup is time-consuming, lacking user-friendliness, and struggles with non-reproducibility across machines, causing collaboration issues. The absence of a reliable local testing environment hampers confident feature development, resulting in a slower lifecycle. Research and Development activities are hindered by difficulties in setting up isolated test environments, stifling innovation. Manual configurations contribute to a lack of standardization, increasing the risk of errors

and impeding scalability. To address these challenges, our proposed solution focuses on simplifying the setup, improving repeatability, and establishing a standardized development environment. This aims to streamline workflows, enhance collaboration, and empower developers to work confidently in both local and production environments, fostering a more efficient development process.

1.3 Hardware Specification

Laptop/Desktop

1.4 Software Overview

NGINIX:

NGINX, an open-source software, serves as a versatile platform for web hosting, reverse proxying, caching, load balancing, media streaming, and other functionalities. Initially developed as a high-performance and stable web server, NGINX has evolved to encompass a broader spectrum of capabilities. Beyond its prowess as an HTTP server, NGINX serves as a proxy for email protocols(IMAP, POP3, and SMTP), and operates as a reverse proxy and load balancer for HTTP, TCP, and UDP servers.

TOMCAT:

Tomcat is an open-source web server and servlet. The Apache Software Foundation has developed it. It is used widely for hosting Java-based applications on the web. It is built on Java technologies and implements the Java Servlet and JavaServer Pages (JSP) specifications. Tomcat acts as a bridge between web servers and Java-based applications, facilitating the execution of dynamic content and processing client requests.

RABBITMQ:

RabbitMQ, an open-source message-broker software often referred to as message-oriented middleware, was initially designed to implement the Advanced Message Queuing Protocol (AMQP). Over time, it has been enhanced through a plug-in architecture to accommodate additional protocols such as the Streaming Text Oriented Messaging Protocol (STOMP), MQ Telemetry Transport (MQTT), and others.

MEMCACHED:

Memcached stands as a high-performance memory caching system that is both free and open-source. Its primary purpose is to cache database data, API calls, or page rendering components in RAM, thereby enhancing application performance.The data it stores can range from small numbers to entire HTML pages. Designed for TCP access, Memcached can operate on separate servers and can be distributed across multiple servers, creating a large hash table to manage data storage.

MYSQL:

MySQL, an RDBMS (Relational Database Management System) developed by Oracle, relies on Structured Query Language (SQL) for its operations.

A database represents a structured compilation of data, spanning from a basic shopping list to a comprehensive picture gallery or a repository for vast corporate network information. Specifically, a relational database serves as a digital repository that organizes data according to the relational model. In this model, tables are composed of rows and columns, with the relationships between data elements adhering to a rigorous logical structure. An RDBMS encompasses the suite of software tools utilized to implement, maage, and query such a database

2.PROBLEM FORMULATION

The existing development environment for our multi-tier web application, comprising the LAMP stack, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL, with WordPress, presents several challenges that hinder the seamless progression of the development lifecycle. The formulation of these problems serves as a foundation for our project objectives:

1. Complexity in Local Setup:

- Problem: The intricate nature of the current local setup process leads to challenges in configuring and establishing a consistent environment on developers' machines. The complexity introduces errors and consumes valuable time during the setup phase.

- Objective: Simplify the local setup by introducing automated processes, reducing the learning curve, and ensuring a straightforward configuration process for individual components.

2. Non-Reproducibility of Environments:

- Problem: Inconsistencies across different machines arise due to the lack of repeatability in the current setup. Variations in environments hinder collaboration and create challenges in maintaining uniform configurations.
- Objective: Implement an automated system to ensure the reproducibility of development environments, fostering consistency and eliminating variations across diverse development machines.

3. Discomfort in Making Real Changes:

- Problem: Developers express discomfort in implementing changes directly in the production environment due to the absence of a reliable and controlled local testing environment. This leads to hesitation and a lack of confidence in making real-time modifications.
- Objective: Establish a comfortable and secure local testing environment that empowers developers to make real changes with confidence. Enable thorough testing before deploying changes to the production environment.

4. Inefficiencies in Research and Development (R&D):

- Problem: The current development environment lacks the necessary support for efficient research and development activities. Developers encounter challenges in conducting meaningful experiments and explorations.
- Objective: Enhance the R&D capabilities by providing tools and resources for developers to create isolated test environments, facilitating experimentation and innovation within a controlled setting.

5. Dependency on Manual Configurations:

- Problem: Heavy reliance on manual configurations introduces inconsistencies, increases the likelihood of errors, and creates a steep learning curve for new team members.
- Objective: Minimize dependency on manual configurations through the introduction of automated processes, ensuring standardization, reducing errors, and providing a smoother onboarding experience for new developers.

By strategically addressing these formulated problems, our project aims to streamline the development workflow, foster collaboration, and empower developers to work with confidence in both local and production environments.

3.OBJECTIVES

The objectives of implementing the automated, repeatable, and standardized local development environment for the multi-tier web application are defined to address the identified challenges and enhance the overall development process. These objectives guide the project towards achieving a more efficient and developer-friendly environment:

1. Simplify Local Setup.
2. Ensure Reproducibility.
3. Enable Comfortable Real Changes.
4. Enhance R&D Capabilities.
5. Minimize Dependency on Manual Configurations.

4.METHODOLOGY

To systematically address the identified problems and achieve the project objectives, a structured methodology will be employed. The following steps outline the approach to be taken in the development of an automated, repeatable, and standardized local setup for the multi-tier web application:

1. Analysis and Requirement Gathering:

- Conduct a comprehensive analysis of the current development environment.

- Gather requirements from developers, project managers, and other stakeholders to understand their needs and expectations.

2. Technology Stack Selection:

- Evaluate and select appropriate technologies and tools for automation, considering factors such as compatibility, scalability, and ease of integration.

- Choose scripting languages and configuration management tools suitable for the specific requirements of the project.

3. Infrastructure as Code (IaC) Implementation:

- Implement Infrastructure as Code principles using tools like Vagrant and VirtualBox to automate the provisioning of development environments.

- Create scripts that define the infrastructure components, ensuring consistency and repeatability across different machines.

4. Configuration Management:

- Implement configuration management tools to automate the setup and configuration of individual components such as LAMP stack, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL.

- Use tools like Ansible or Puppet to enforce consistent configurations.

5. Local Testing Environment Setup:

- Establish a local testing environment that mirrors the production setup, enabling developers to test changes in a controlled and secure environment.

- Integrate tools for version control, continuous integration, and automated testing to enhance the development and testing process.

6. Documentation and Training:

- Document the automated setup processes, including step-by-step guides for developers.

- Provide training sessions to familiarize the development team with the new automated environment and processes.

7. User Feedback and Iterative Development:

- Gather feedback from developers during and after the implementation phase.

- Iterate on the automated setup based on user feedback, addressing any issues or enhancements suggested by the development team.

8. Integration with Development Workflow:

- Integrate the automated local setup seamlessly into the existing development workflow, ensuring minimal disruption to daily activities.

- Develop plugins or extensions for popular IDEs like Visual Studio Code to enhance developer experience.

9. Security Considerations:

- Implement security best practices in the automated setup, ensuring that the local testing environment is secure and isolated.

- Conduct security audits and implement necessary measures to protect sensitive data.

10. Monitoring and Maintenance:

- Implement monitoring tools to track the performance and health of the local development environment.

- Establish regular maintenance procedures to address updates, patches, and changes in the technology stack.

5.EXPERIMENTAL SETUP

The experimental setup for validating the effectiveness and efficiency of the automated, repeatable, and standardized local development environment involves a step-by-step process:

1. Hardware and Software Requirements:

Hardware:

- Development machines with a minimum of 8GB RAM, multi-core processors, and sufficient storage for virtualization and development activities.

Software:

- Hypervisor: VirtualBox or VMware.

- Vagrant for automating the provisioning of virtual machines.

- Cli: git bash.

- Development tools: Visual Studio Code (VS Code), Git, and relevant plugins/extensions.

2. Development Environment Setup:

Installation and Configuration:

- Install VirtualBox or VMware on development machines.

- Install Vagrant and configure its settings.

- Clone the project repository containing automated setup scripts and configurations.

- Install necessary development tools and plugins/extensions.

3. Automated Setup Execution:

Execute Automation Scripts:

- Navigate to the project directory.

- Run Vagrant commands to initiate the automated setup process.

```bash
vagrant up
```

- Monitor the setup process for any errors or issues.

4. Development and Testing:

Feature Development:

- Use VS Code for coding and Git for version control.

- Develop and test features within the local development environment.

5. Performance and Stability Testing:

Metrics and Evaluation:

- Use tools like Apache JMeter or Gatling for performance testing.

- Measure key performance metrics (response times, resource utilization).

- Evaluate system stability under varying workloads.

Summary and Recommendations:

- Prepare a comprehensive report summarizing experimental setup, findings, and recommendations.

- Include insights into the effectiveness and efficiency of the automated local development environment.

By following this experimental setup, the project aims to validate the seamless integration of the automated local development environment, ensuring its effectiveness, reliability, and user-friendliness for multi-tier web application development.

6.CONCLUSION

The V-Profile project aims to streamline the local setup process for multi-tier web applications, facilitating seamless research and development activities on laptops or desktops. Addressing the challenges of complexity and non-repeatability in local setups, the project provides an automated solution through code, enabling users to make changes comfortably in a simulated environment.

Utilizing tools such as Oracle VM VirtualBox for hypervisor, Vagrant for automation, Git Bash for CLI, and VS Code as the IDE, the project's objective is to automate VM setup locally, empowering users to conduct extensive research and development activities on their machines.

The architecture of the project services encompasses crucial components such as NGINX, TOMCAT for Java web applications, RABBITMQ, MEMCACHED, and MYSQL, fostering a comprehensive development environment.

The automated setup architecture revolves around Vagrant, VirtualBox, and Git Bash, orchestrating an efficient environment for project development and experimentation.

In conclusion, the V-Profile project offers a robust framework for setting up and conducting research and development activities locally, enhancing user comfort, repeatability, and efficiency in managing project runtime services.

7. TENTATIVE CHAPTER PLAN FOR THE PROPOSED WORK

CHAPTER 1: INTRODUCTION

Our project addresses challenges in multi-tier web application development by revolutionizing the workflow through automated, repeatable, and standardized local setups. Leveraging tools like Vagrant, VirtualBox, Git Bash, and Visual Studio Code, we streamline the development environment, enhancing accessibility and consistency across machines. This initiative accelerates the development cycle, fosters collaboration, and promotes a comfortable space for experimentation. The multi-tier architecture, including LAMP, NGINX, Tomcat, RabbitMQ, Memcached, and MySQL with WordPress, offers a comprehensive platform for addressing server-side scripting, database

management, web server configurations, and seamless component integration. Through a code-driven local environment, developers engage in meaningful R&D activities, creating isolated test environments for thorough testing before production implementation. Git version control enhances collaboration, while Vagrant and VirtualBox provide a scalable infrastructure. Overall, our project redefines the development experience by overcoming challenges, establishing standardized local setups, and empowering developers for efficient research, development, and innovation in multi-tier web applications.

CHAPTER 2: OBJECTIVE

The project's objectives revolve around enhancing the multi-tier web application development lifecycle by implementing an automated, repeatable, and standardized local development environment. Goals include simplifying the local setup, ensuring reproducibility, enabling confident real changes through thorough testing, and enhancing R&D capabilities within isolated test environments. By minimizing dependency on manual configurations and integrating seamlessly into the existing workflow, the project aims to improve collaboration and minimize disruptions. Additional objectives focus on optimizing performance and stability, enhancing security measures, providing comprehensive documentation, gathering and acting on user feedback, and establishing a framework for continuous improvement. Through these objectives, the project strives to create a developer-friendly platform, contributing to the success of the multi-tier web application development lifecycle by fostering efficiency, reliability, and user satisfaction.

CHAPTER 3: METHODOLOGIES

The methodology for implementing the automated, repeatable, and standardized local setup for the multi-tier web application is structured and systematic. It begins with a comprehensive analysis of the current development environment and gathers requirements from stakeholders. The selection of appropriate technologies and tools, including Infrastructure as Code principles, configuration management, and local testing environment setup, follows. Documentation and training ensure a smooth transition for the development team.

User feedback plays a crucial role, with iterative development addressing suggestions and issues. Integration with the existing development workflow, security considerations, and monitoring tools contribute to a seamless and secure environment. The methodology concludes with a focus on ongoing maintenance, ensuring the continued efficiency and relevance of the automated setup. This approach aims to deliver a user-friendly and secure local development environment, promoting collaboration, innovation, and efficiency in multi-tier web application development.

CHAPTER 4: EXPERIMENTAL SETUP

The experimental setup for validating the automated, repeatable, and standardized local development environment involves configuring development machines with a minimum of 8GB RAM, multi-core processors, and necessary storage. Key software components include VirtualBox or VMware for virtualization, Vagrant for automation, Git bash for command-line operations, and Visual Studio Code with relevant extensions for development. The setup process includes installing and configuring these tools, cloning the project repository with automated scripts, and installing necessary development tools.

Automated setup execution is initiated by running Vagrant commands, closely monitoring for errors. Feature development occurs within Visual Studio Code, utilizing Git for version control. Performance and stability testing involve using tools like Apache JMeter, assessing key metrics, and evaluating system stability under varying workloads.

The experimental findings are summarized in a comprehensive report, highlighting insights into the effectiveness and efficiency of the automated local development environment. The report includes recommendations based on the observed performance and stability metrics, providing valuable guidance for further optimization and utilization in multi-tier web application development.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

The V-Profile project streamlines local setup for multi-tier web applications, employing Oracle VM VirtualBox, Vagrant, Git Bash, and VS Code. By automating VM setup, it empowers users for comfortable research and development on laptops/desktops. The architecture includes NGINX, TOMCAT, RABBITMQ, MEMCACHED, and MYSQL, fostering a comprehensive development environment. Automated setup, centered around Vagrant, VirtualBox, and Git Bash, offers efficiency for project development. In conclusion, V-Profile provides a robust framework, enhancing user comfort, repeatability, and efficiency in local research and development activities.