LINUX: Project Assignment

Automated Monitoring and Maintenance for an E-commerce Website

Problem Statement

An e-commerce company runs its website on a Linux-based web server. The company faces challenges in maintaining the server's health and ensuring the website is always up and running. Manual monitoring and maintenance tasks are time-consuming and prone to human error. The company needs an automated solution to monitor system performance, perform regular updates, and back up critical data to ensure continuous availability and reliability of the website.

User Stories

User Story 1

As a system administrator, I want to set up a virtualized environment for the web server so that I can easily manage and configure the server without impacting the physical hardware.

Acceptance Criteria:

- A virtual machine (VM) is created using VirtualBox.

- Ubuntu Server is installed on the VM.

- The web server (Apache) is installed and configured on the VM.

User Story 2

As a system administrator, I want to monitor the web server's performance using basic Linux commands so that I can identify and resolve performance issues promptly.

Acceptance Criteria:

- The system administrator can use `top` to monitor CPU and memory usage.

- Disk space usage can be checked using `df -h`.

- Network activity is monitored using `ip a` and `netstat -tuln`.

- Logs for the web server can be accessed and monitored.

User Story 3

As a system administrator, I want to automate system updates to ensure the web server is always running the latest security patches and software updates.

Acceptance Criteria:

- A shell script is created to update the system.

- The script is scheduled to run daily using cron.

- The system is updated automatically without manual intervention.

User Story 4

As a system administrator, I want to automate the backup of web content to ensure that the website's data is securely backed up and can be restored in case of data loss.

Acceptance Criteria:

- A shell script is created to back up the web server's data.

- The backup script is scheduled to run weekly using cron.

- Backups are stored in a designated backup directory with date-stamped filenames.

User Story 5

As a system administrator, I want to ensure that all backup and update scripts are properly documented so that other team members can understand and maintain them.

Acceptance Criteria:

- Detailed documentation is provided for each script.

- The documentation includes explanations of the commands used, the purpose of the script, and how to execute and schedule the scripts.

- Screenshots of the setup and execution processes are included in the documentation.

Implementation Plan

User Story 1: Virtualized Environment Setup

- Create a New Virtual Machine:

- Name: 'EcommerceWebServer'

- Type: Linux

- Version: Ubuntu 64-bit

- Memory: 4096 MB

- Virtual Hard Disk: 40 GB

- Install Ubuntu Server:

- Download the Ubuntu Server ISO.

- Start the VM and select the ISO as the startup disk.

- Follow the installation prompts to set up Ubuntu Server.

- Install Apache web server:

- sudo apt update

- sudo apt install apache2 -y

- sudo systemctl start apache2

- sudo systemctl enable apache2

User Story 2: Monitoring System Performance

- Monitor System Performance:

- Use `top` to monitor CPU and memory usage.

- Check disk usage with `df -h`.

- Monitor network activity with `ip a` and `netstat -tuln`.

- Access Apache logs:

- sudo tail -f /var/log/apache2/access.log

- sudo tail -f /var/log/apache2/error.log

User Story 3: Automate System Updates

- Create an Update Script:

- Create a script named `update_system.sh`:

-
  ```bash

- #!/bin/bash

- sudo apt update && sudo apt upgrade -y

- echo 'System update completed.'

- ```

- Make the script executable:
  ```

- chmod +x update_system.sh

- Schedule the Script:

- Open the cron table:

- crontab -e

- Add the following line to schedule the update script daily at 2 AM:

- 0 2 * * * /path/to/update_system.sh

User Story 4: Automate Data Backup

- Create a Backup Script:

- Create a script named `backup.sh`:

-
  ```bash

- #!/bin/bash

- tar -czvf /var/backups/web_backup_$(date +%F).tar.gz /var/www/html

- echo 'Backup completed.'

- ```

- Make the script executable:

- chmod +x backup.sh

- Schedule the Script:

- Open the cron table:

- crontab -e

- Add the following line to schedule the backup script weekly on Sundays at 3 AM:

- 0 3 * * 0 /path/to/backup.sh

## User Story 5: Documentation

- Document Each Script:

- Create a detailed document for each script.

- Include the purpose of the script, a breakdown of the commands used, and step-by-step instructions for execution and scheduling.

- Provide screenshots of the setup and execution processes.
  ```

Summary

By completing these user stories, the e-commerce company will have an automated system for monitoring and maintaining their web server. This ensures the server's performance and reliability, reduces manual workload, and minimizes the risk of human error. The comprehensive documentation will help the team understand and manage the automated tasks effectively.

Deliverables

1. **Documentation**:

   - Detailed steps with screenshots.

   - Descriptions of the commands and scripts used.

2. **Script Files**:

   - `update_system.sh`

   - `backup.sh`

3. **Final Report**:

   - Compile the documentation and scripts into a ZIP file named `WebServerAutomation_YourName.zip`.

4. **Submission**:

   - Submit the ZIP file via the designated submission platform or email it to the instructor.

Evaluation Criteria

1. **Completeness**: All steps are completed and documented.

2. **Accuracy**: Commands and scripts are correct and functional.

3. **Clarity**: Documentation is clear, with detailed explanations and screenshots.

4. **Automation**: Scripts effectively automate the tasks described.