A

Project Report On

"Linux Administration And Virtualization Implementation"

(Under- "Industry Internship Project" Track)

Submitted

in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Information Technology

by

Mr. Sumit Anand Padiyar (2010032)

at

Prodevans Technologies, Bengaluru

Under the Guidance of

Prof. R.B.Sadigale

Information Technology Department

K.E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(An Empowered Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

2023-2024

K. E. Society's

Rajarambapu Institute Of Technology, Rajaramnagar

(An Empowered Autonomous Institute, Affiliated to Shivaji University)

Department of Information Technology

CERTIFICATE

This is to certify that the project under Industry Internship & Project (IIP) track completed at **"Prodevans Technologies, Bengaluru"** is the bona fide work submitted by the following student, to the Rajarambapu Institute of Technology, Rajaramnagar during the academic year 2023-24, in partial fulfillment for the award of the degree of B. Tech in **Computer Science & Information Technology** under our supervision.The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Name of Student	**Roll Number**
Sumit Anand Padiyar	2010032

Date:

Place: RIT, Rajaramnagar

Prof. R.B.Sadigale
Industry Internship & Project Mentor(College)

Mr. Tausif Shaikh
Industry Internship & Project Mentor(Industry)

External Examiner

Prof. M. N. Mulla
Training & Placement Coordinator

Dr. A. C. Adamuthe
Head of Department

DECLARATION

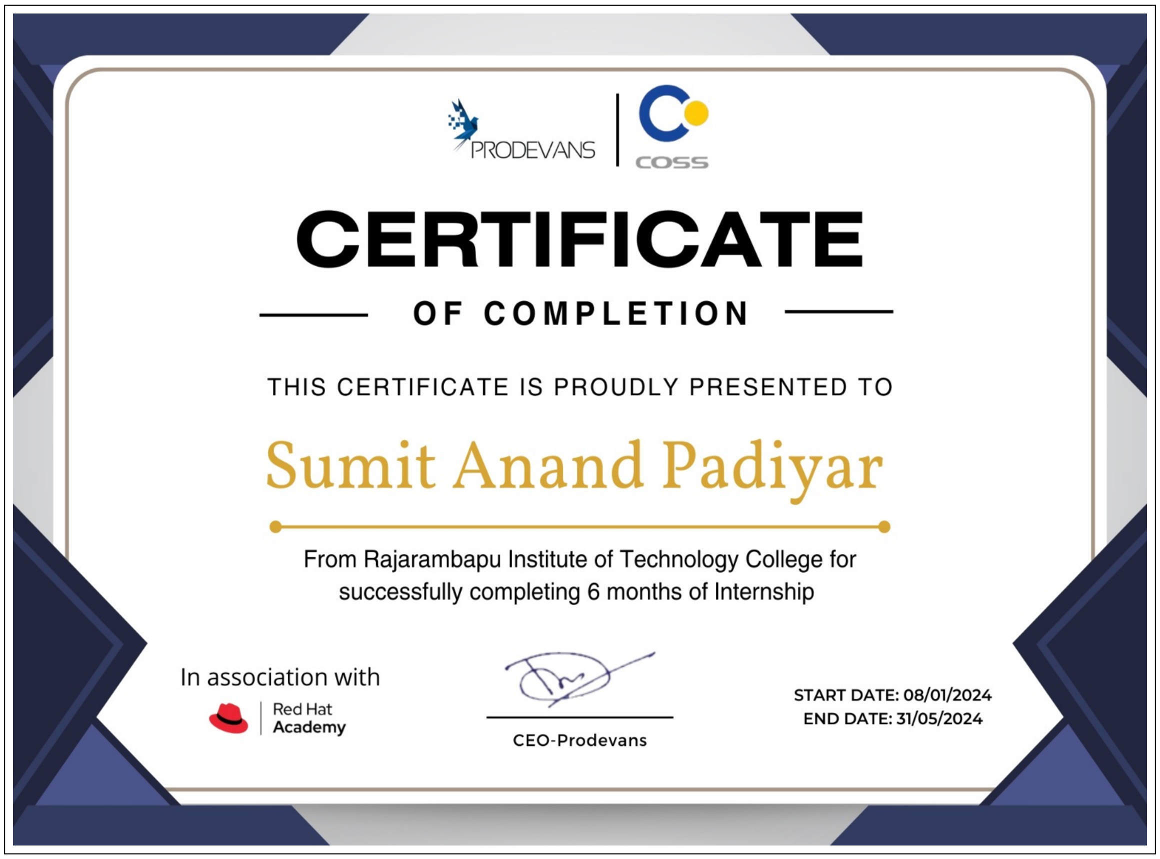
I declare that this report reflects my thoughts about the subject in my own words. I have sufficiently cited and referenced the original sources, referred or considered in this work. I have not plagiarized or submitted the same work for the award of any other degree. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute.

Sr. no	**Student Name**	**Roll No**	**Signature**
1	Sumit Anand Padiyar	2010032	

Date:

Place: RIT, Rajaramnagar

Project Completion Letter



PRODEVANS | COSS

CERTIFICATE

OF COMPLETION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Sumit Anand Padiyar

From Rajarambapu Institute of Technology College for successfully completing 6 months of Internship

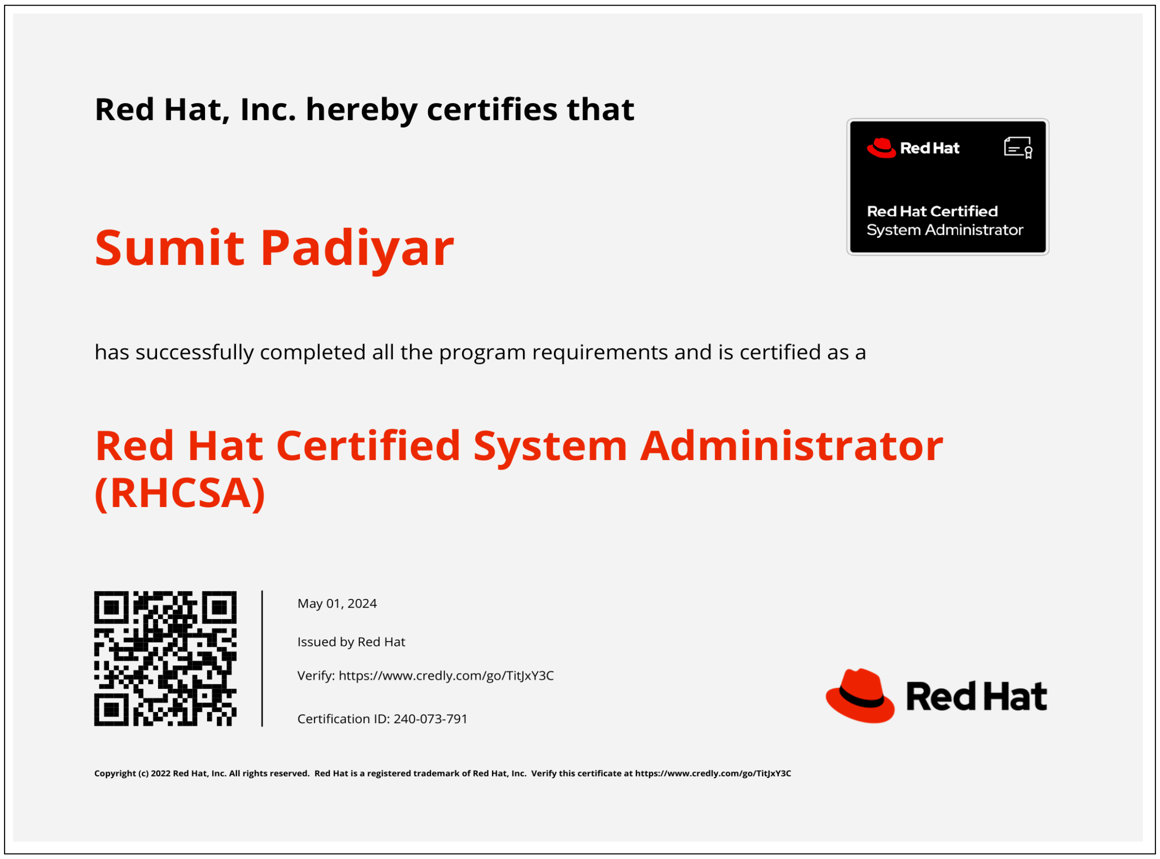
In association with Red Hat Academy

[illegible]

CEO-Prodevans

START DATE: 08/01/2024
END DATE: 31/05/2024

Red Hat System Administration (RHCSA)



Red Hat, Inc. hereby certifies that

Red Hat Certified System Administrator

Sumit Padiyar

has successfully completed all the program requirements and is certified as a

Red Hat Certified System Administrator (RHCSA)

May 01, 2024

Issued by Red Hat

Verify: https://www.credly.com/go/TitJxY3C

Certification ID: 240-073-791

Red Hat

Copyright (c) 2022 Red Hat, Inc. All rights reserved. Red Hat is a registered trademark of Red Hat, Inc. Verify this certificate at https://www.credly.com/go/TitJxY3C

ACKNOWLEDGEMENT

I take this opportunity to thank all those who have contributed to the successful completion of an Project Under the Industry Internship & Project (IIP) track at **“Prodevans Technologies ,Bengaluru”**. I sincerely wish to express my gratitude to Industry Internship & Project (IIP) Mentor **Prof. R.B. Sadigale** for full support, expert guidance, encouragement, and kind cooperation throughout the Internship work. I am greatly indebted to him for his help throughout project work. I express my sincere gratitude towards **Dr. A. C. Adamuthe**, Head of the Department, Information Technology, for providing the necessary facilities, guidance, and support.

I respect and thank **Mr. Tausif Shaikh** for providing me an opportunity to do a project in Prodevans Technologies, Bengaluru and giving us all support and guidance, which made me complete the Internship duly. I am extremely thankful to him for providing such nice support and guidance, although he had a busy schedule managing corporate affairs.

I thank **Prof. M. N. Mulla** for providing an Internship & Project Opportunity in an Industry.

I am thankful for and fortunate enough to get constant encouragement, support, and guidance from all teaching staff of the Information Technology Department, which helped me in successfully completing Internship.

Nevertheless, I express my gratitude toward my family members for their kind cooperation and encouragement which helped me in the completion of this Internship.

ABSTRACT

This project report summarizes key tasks completed during my internship, including the training for Red Hat Certified System Administrator (RHCSA) and Red Hat Certified Engineer (RHCE) certifications, focusing on Linux system administration. Shell scripts were developed to automate routine tasks such as backups, user management, and system monitoring, enhancing efficiency and reducing human error. A high-availability (HA) cluster was configured with multiple Linux nodes, incorporating failover mechanisms to ensure minimal downtime and reliable service. A DNS server using BIND was installed and configured, with DNS zone files managed and security measures implemented. Network connections for virtual machines (VMs) were established using NAT and bridge networking, with Bond networking enhancing redundancy and performance. A local repository was set up using the Apache web server, synchronized with an official package source to enable controlled and reliable software management for client systems. Client repositories were updated to use the local Apache-hosted repository, with package installations and updates tested and monitored to maintain security and availability. These tasks collectively improved system efficiency, reliability, and security, highlighting the importance of automation, redundancy, and secure configurations in system administration.

Contents

List of Figures

Chapter 1

Introduction

1.1 Project Overview

During my internship, I underwent extensive training to achieve the Red Hat Certified System Administrator (RHCSA) and Red Hat Certified Engineer (RHCE) certifications. This training provided a comprehensive understanding and practical experience in managing and securing Red Hat Enterprise Linux systems. Key skills acquired encompassed system configuration, network management, security implementation, and automation through scripting. This training served as a cornerstone in establishing a robust foundation in Linux system administration, enhancing my proficiency in managing intricate IT environments efficiently and securely.

Embarking on pivotal projects further enriched my expertise in Linux system administration. Firstly, I immersed myself in Shell Scripting Basics, crafting scripts tailored to automate routine tasks such as backups, user management, and system monitoring. These scripts, empowered by fundamental shell commands and control structures, not only optimized operational efficiency but also mitigated the risk of human error. Concurrently, I spearheaded the High-Availability Cluster Implementation, configuring a resilient HA cluster with failover mechanisms spanning across multiple Linux nodes. This initiative aimed to ensure uninterrupted service delivery by seamlessly transferring workloads in the event of node failures. Through these projects, I honed essential skills in automation, redundancy, and system reliability, highlighting their critical role in enterprise-grade systems administration.

Additionally, I undertook several significant tasks that augmented my proficiency in Linux system administration. Firstly, I executed the DNS Server Setup project, meticulously installing and configuring a DNS server using BIND. This entailed managing DNS zone files and implementing stringent security measures to safeguard against potential vulnerabilities. Simultaneously, I delved into KVM Networking and Storage, establishing network connections for virtual machines (VMs) through NAT and bridge networking techniques. Leveraging Bond networking further enhanced both redundancy and performance, ensuring dependable network connections for the VMs. Furthermore, I led the Local Repository Setup Using Apache project, orchestrating the setup of a local repository hosted on the Apache web server. Synchronized with an official package source, this repository facilitated controlled and dependable software management for client systems. I meticulously updated client repositories to utilize the local Apache-hosted repository, rigorously testing and monitoring package installations and updates to uphold security and availability standards. These endeavors collectively underscored my proficiency in system configuration, networking, and software management, emphasizing the significance of robust infrastructure and meticulous maintenance in enterprise environments.

1.2 Objectives

1. To obtain RHCSA and RHCE certifications.

2. To develop automation skills.

3. To configure high-availability systems.
4. To manage DNS servers.
5. To establish network connections for VMs.
6. To set up local software repositories.
7. To monitor and maintain systems.

1.3 Technologies used in company

1. **Linux**

   Linux is an open-source, Unix-like operating system built around the Linux kernel. It offers a highly flexible and customizable computing environment, making it suitable for a wide range of applications, from personal desktop computers to high-performance servers and supercomputers. Popular Linux distributions include Red Hat Enterprise Linux (RHEL), CentOS, Fedora, Debian, Ubuntu, and SUSE Linux Enterprise Server (SLES). Each distribution bundles the Linux kernel with a collection of software packages, utilities, and tools tailored for specific use cases, ranging from desktop environments to server deployments. Linux is renowned for its security, stability, and cost-effectiveness, which has contributed to its widespread adoption across various industries and organizations worldwide..

2. **Ansible**

   Ansible is an open-source IT automation engine that simplifies application deployment, cloud provisioning, and configuration management across diverse environments. It uses YAML-based playbooks to define automation tasks and roles to organize related tasks and configurations. Playbooks allow users to describe the desired state of their infrastructure, and Ansible takes care of the necessary steps to achieve that state. Roles provide a way to encapsulate and share configurations, making it easier to maintain and reuse code. Additionally, Ansible Collections are pre-packaged sets of roles, modules, and plugins that can be easily installed and used, promoting code reusability and collaboration within the Ansible community. With its agentless architecture and simple yet powerful automation framework, Ansible has become a popular choice for DevOps teams and system administrators.

3. **shell-scripting**

   Shell scripting is the art of automating tasks and streamlining workflows on Unix-like operating systems by writing a series of commands in a plain text file called a shell script. These scripts can include variables, control structures (such as conditionals and loops), functions, and various Unix utilities and commands. Shell scripts are typically written in languages like Bash (Bourne-Again SHell) or other shells like Zsh, Ksh, or the original Bourne shell (sh). They provide a powerful and flexible way to automate repetitive tasks, create custom tools and utilities, and simplify complex operations. Shell scripting is a core skill for system administrators, developers, and power users working with Unix-like systems, as it enables them to improve efficiency, productivity, and automation.

4. **Virtualization**

   Virtualization is the process of creating virtual versions of computing resources, such as operating systems, storage devices, and networking components. It involves the use of hypervisors, which can be classified as Type 1 (bare-metal) or Type 2 (hosted). Type 1 hypervisors, like VMware ESXi and Microsoft Hyper-V, run directly on the hardware, while Type 2 hypervisors, like Oracle VirtualBox and KVM (Kernel-based Virtual Machine), run on top of a host operating system. Hardware virtualization technologies, like Intel VT-x and AMD-V, enable efficient virtualization by providing hardware-level support for running virtual machines.

Virtualization in Linux allows multiple operating systems and applications to run on a single physical machine by abstracting the hardware resources. This is achieved through hypervisors, such as KVM (Kernel-based Virtual Machine), which is integrated into the Linux kernel, and other popular solutions like Xen and VMware. Linux virtualization supports both full and para-virtualization, enabling efficient resource management and isolation. It is widely used for server consolidation, development environments, and cloud computing. Tools like QEMU, libvirt, and virt-manager facilitate the creation, management, and monitoring of virtual machines, providing a robust and flexible virtualization platform.

Chapter 2

Requirement Specification

2.1 Problem Statement

Develop a comprehensive system administration solution integrating shell scripting for automating tasks like backups and user management. Configure a high-availability cluster with failover mechanisms to ensure continuous service availability. Implement DNS server security measures and optimize networking for VMs. Set up a local repository using Apache for reliable software management across the enterprise.

2.2 Traditional System

The previous system faced challenges due to manual intervention, leading to inefficiencies and increased error risks, while downtime during hardware failures or system errors was a concern, impacting service availability. Additionally, security vulnerabilities posed threats to system integrity and data protection, and networking configurations were suboptimal, hindering performance and scalability, especially within virtualized environments, and reliance on external sources for software management introduced dependencies and security risks. To overcome these challenges, we implemented modern system administration practices. Firstly, we automated routine tasks using shell scripting, reducing manual workload and error risks. Secondly, we deployed high-availability clustering with failover mechanisms, ensuring continuous service availability even during hardware failures. Thirdly, we enhanced security measures such as DNS server hardening and network segmentation to fortify system defenses against cyber threats. Fourthly, we optimized networking for virtual machines, improving performance and scalability within virtualized environments. Lastly, we established a local repository using Apache, enabling controlled software management and reducing reliance on external sources, thus enhancing security and availability.

2.3 Feature Specification

(a) Utilization of Basic Shell Commands.

(b) Networking Configuration for VMs.

(c) Apache Web Server Installation and Configuration.

(d) Repository Synchronization.

(e) Redundancy and Performance Enhancement.

(f) Automation of Routine Tasks.

(g) Failover Mechanism Configuration.

(h) BIND Setup and Configuration.

2.4 Software Requirements

(a) VM Ware / Virtual Box.

(b) Red Hat Enterprise Linux / Ubuntu / Fedora.

(c) Python.

(d) Vim Editor.

2.5 Hardware Requirements

(a) RAM: 8 GB.

(b) Processor: i3/i5/i7 and further series processors.

(c) Graphics card is optional but is recommended, to increase performance.

2.6 Preliminaries

(a) **Red Hat Enterprise Linux (RHEL)**

Red Hat Enterprise Linux (RHEL) is a trusted operating system used by businesses worldwide for its reliability and security. It provides long-term support, ensuring systems remain stable and secure for years. RHEL comes with advanced security features like SELinux to protect against threats. With tools like yum, managing software updates and configurations is easy. Plus, Red Hat offers excellent customer support for quick issue resolution. RHEL supports modern technologies like containers and cloud computing, making it adaptable to various IT environments. Overall, RHEL is a dependable choice for businesses seeking a robust operating system for their critical workloads.

(b) **Red Hat Certified System Administrator (RHCSA)**

The Red Hat Certified System Administrator (RHCSA) certification is a widely recognized credential in the field of Linux system administration. It validates the skills and knowledge required to configure, manage, and troubleshoot Red Hat Enterprise Linux (RHEL) systems efficiently and effectively. RHCSA candidates are tested on a range of tasks including system administration, user and group management, file system management, network configuration, security administration, and troubleshooting. Achieving RHCSA certification demonstrates proficiency in essential Linux system administration tasks, making it a valuable qualification for IT professionals seeking to advance their careers in Linux-based environments.

(c) **Red Hat Certified Engineer (RHCE)**

The Red Hat Certified Engineer (RHCE) certification is an advanced credential that builds upon the foundation laid by RHCSA. It validates the expertise of IT professionals in designing, deploying, managing, and troubleshooting complex Red Hat Enterprise Linux (RHEL) systems. RHCE candidates are tested on a broad range of topics including system configuration and management, network services, security, virtualization, and automation. Successful completion of the RHCE exam demonstrates the ability to handle advanced tasks such as setting up and configuring network services, implementing security measures, managing virtualization environments, and automating system administration tasks using scripting and other tools. RHCE certification is highly regarded in the industry and is a testament to an individual's proficiency in deploying and managing enterprise-grade Linux environments.

(d) **Python**

Python is a high-level, interpreted programming language known for its readability and simplicity. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it versatile for a wide range of applications. Python's extensive standard library and active community contribute to its popularity in web development, data analysis, artificial intelligence, scientific computing, and automation. Its clear syntax and ease of learning make Python an excellent choice for both beginners and experienced developers.

(e) **Virtualization**

Virtualization technology enables the creation of virtual instances of computer hardware, operating systems, storage devices, and network resources. It allows multiple virtual machines (VMs) to run on a single physical machine, thereby optimizing resource utilization, improving scalability, and enhancing flexibility in IT infrastructure management. Virtualization is widely used in data centers, cloud computing environments, and development/test environments. Key virtualization technologies include hypervisors like VMware vSphere, Microsoft Hyper-V, and KVM (Kernel-based Virtual Machine). Professionals proficient in virtualization possess skills in designing, deploying, managing, and troubleshooting virtualized environments. They are adept at configuring virtual machines, optimizing resource allocation, ensuring security, implementing backup and recovery strategies, and monitoring performance. Certification programs such as VMware Certified Professional (VCP) and Microsoft Certified: Azure Administrator Associate validate expertise in virtualization technologies, making them valuable credentials for IT professionals aiming to excel in the field of virtualization and cloud computing.

(f) **Shell Scripting**

Shell scripting is a powerful tool for automating tasks and managing system configurations in Unix-like operating systems. It involves writing scripts using shell commands and control structures to execute sequences of commands automatically. Shell scripts are utilized for various purposes including system administration, software deployment, log analysis, and data processing. Proficiency in shell scripting requires knowledge of shell scripting languages such as Bash, Python, or Perl, as well as familiarity with common Unix utilities and commands. Skilled shell scripters can streamline repetitive tasks, improve system efficiency, and ensure consistency in system configurations. Certifications such as the Linux Professional Institute Certification (LPIC) or Red Hat Certified Engineer (RHCE) often include shell scripting as part of their curriculum, recognizing its importance in the skillset of system administrators and DevOps professionals.

Chapter 3

System Design

3.1 System Architecture

The system architecture of a computer operating system is organized into several hierarchical layers. At the top, applications encompass user interfaces and utilities that facilitate direct interaction with the system. These applications run on top of the windowing system, which manages graphical user interfaces (GUIs), particularly in Linux environments. Beneath the windowing system lies the shell, such as Bash or Zsh, which provides a command-line interface for executing user commands and scripts. Supporting the shell are system libraries, like libc and libX11, which offer essential functions and application programming interfaces (APIs) that enable seamless communication between the shell and the kernel. The kernel, the core component of the operating system, manages critical system operations and acts as an intermediary between software and hardware. It includes device drivers that facilitate communication with hardware components such as the CPU, memory, and storage devices. The kernel also oversees memory management, ensuring efficient use of virtual memory, and process management, which handles the creation and execution of processes through mechanisms like fork and exec. File systems, including formats like ext4 and NTFS, manage data storage and retrieval, while the network stack, utilizing protocols like TCP/IP, manages network communications. This structured and layered architecture ensures an organized and efficient flow of data and commands from user-level applications down to the hardware level, facilitating smooth and effective system operation..

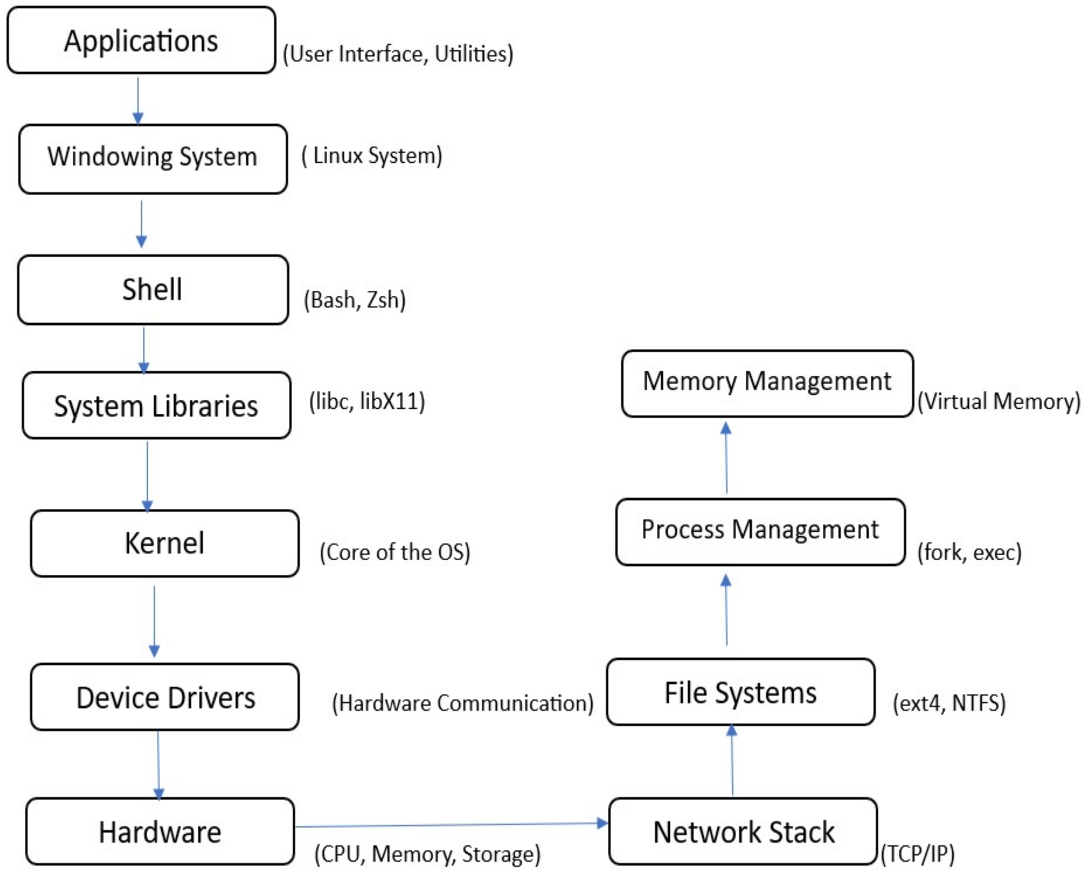


Figure 3.1: System Architecture

3.2 Data Flow Diagram

3.2.1 DFD of DNS Server configuration

The DFD for a DNS server shows the internal processes involved, including query processing, zone lookup, response generation, and their interactions with external entities like clients and DNS data sources. It illustrates the flow of DNS queries, lookup requests, DNS records, and responses within the server.

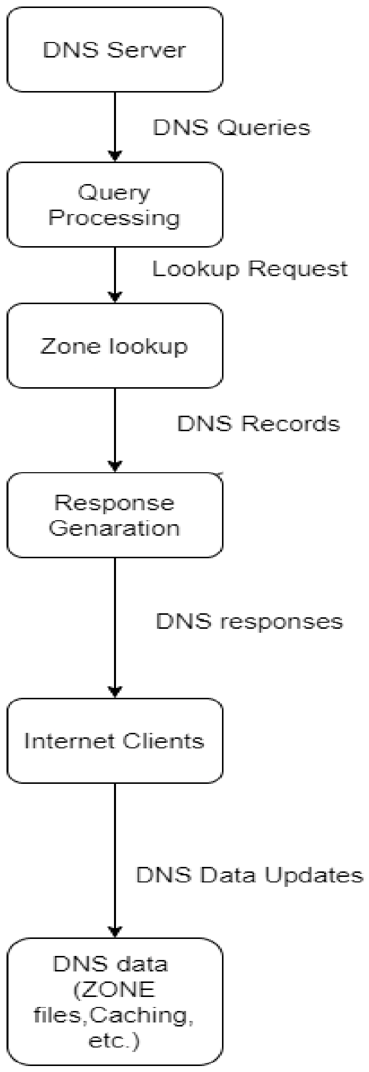


Figure 3.2: DFD for DNS SERVER

3.2.2 DFD of HA-Cluster

The DFD for setting up an HA cluster includes five main processes: Cluster Configuration, Resource Configuration, Cluster Node Setup, Validation Test, and Monitoring Management. It details data flows like Setup Request, Configuration Data, Resource Definitions, Node Status, and Cluster Status. Interactions with external entities, such as the HA Cluster Setup Tool and Shared Storage, are depicted, showcasing the setup, validation, and ongoing management tasks. This DFD outlines the systematic steps and data interactions involved in establishing and maintaining an HA cluster.

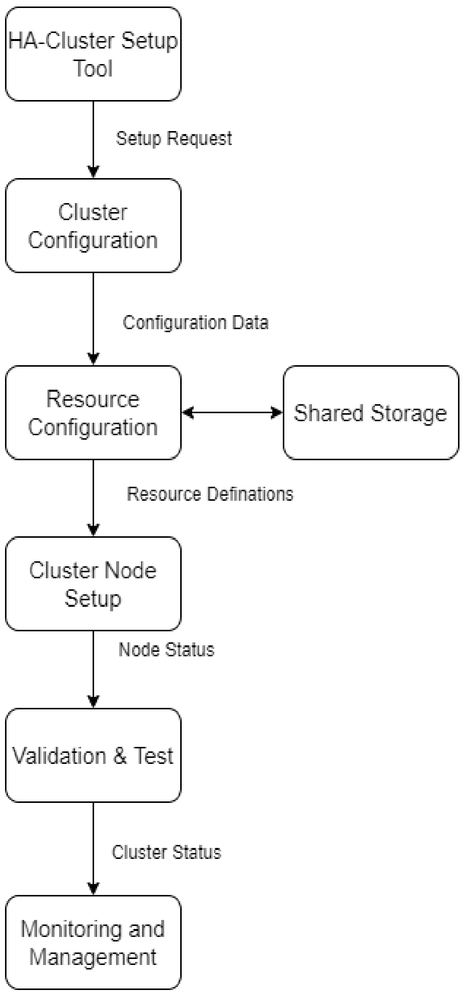


Figure 3.3: DFD for HA-Cluster

3.2.3 LocalRepo Config DFD

The diagram illustrates the process of setting up a local package repository hosted on an Apache server and its utilization by client systems. An admin user sets up the local repository, which is then hosted by the Apache server. The hosted package repository contains software packages that can be accessed and used by client systems for installing applications or packages locally.

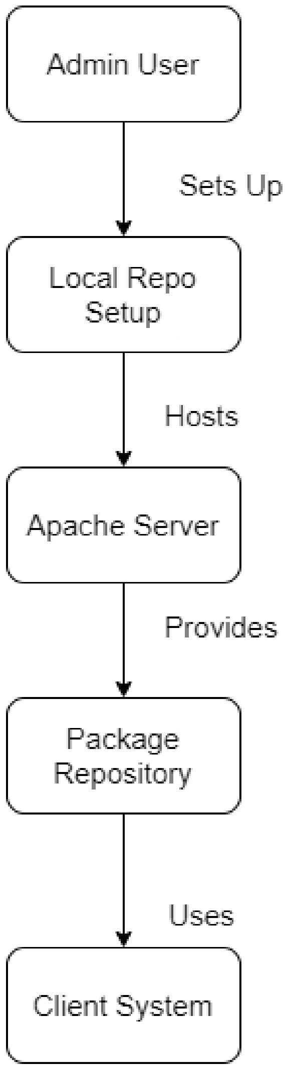


Figure 3.4: Repo config DFD

3.3 Use Case Diagram

The use case diagram illustrates the interactions between the System Administrator and the Automated Monitoring System to maintain the health and availability of a Linux-based web server for an e-commerce company. The System Administrator sets up and configures the automated system, which then takes over key maintenance tasks. The Automated Monitoring System continuously monitors system performance, ensuring optimal CPU, memory, disk usage, and network activity. It automatically executes regular updates to keep the server secure and up-to-date, and performs regular backups of critical data to prevent data loss. In case of anomalies or potential issues, the system generates alerts to notify the administrator. Additionally, it generates periodic performance and maintenance reports, providing insights into system health. The system also includes a capability to restore data from backups if needed, ensuring continuous website availability and reducing downtime. This automation minimizes manual intervention, reduces human error, and allows the company to focus more on its core business activities.

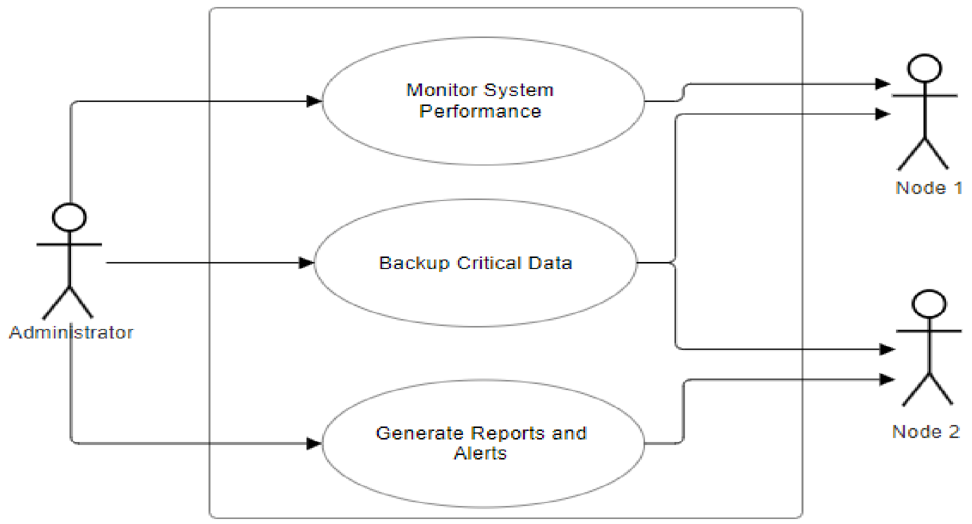


Figure 3.5: Use Case Diagram

Chapter 4

Implementation

4.1 Project Introduction

The project involves several key tasks as a Linux Administrator to manage systems efficiently. These tasks include learning the basics of shell scripting, which is essential for automating repetitive tasks and managing system configurations. Implementing a high-availability cluster on Linux ensures that services remain available and minimizes downtime. Setting up a DNS server on Linux involves various configurations to manage domain name resolution within the network.

Additionally, the project covers networking and storage in KVM (Kernel-based Virtual Machine), including understanding how network connectivity is achieved using NAT (Network Address Translation) and the Bond networking tool to enhance network performance and reliability. Another critical task is setting up a local repository using an Apache server, which serves as a centralized location for software packages. This local repository is then used by client systems to install and update software, ensuring consistent and reliable package management. These tasks collectively enhance system reliability, improve network management, and ensure efficient service delivery in a Linux environment.

4.2 Specifications

(a) **Shell Scripting Basics:** Write scripts to automate repetitive tasks such as backups, user management, and system monitoring. Utilize basic shell commands and control structures (loops, conditionals)..

(b) **High-Availability Cluster Implementation:** Configure multiple Linux nodes to function as a single HA cluster. Implement failover mechanisms to automatically transfer workloads to a backup node in case of failure.

(c) **DNS Server Setup:** Install and configure DNS server software (e.g., BIND).Create and manage DNS zone files for forward and reverse lookups. Secure the DNS server against common vulnerabilities and perform regular updates..

(d) **KVM Networking and Storage:** Set up network connectivity for VMs using NAT and bridge networking.Implement the Bond networking tool for network redundancy and performance improvement.

(e) **Local Repository Setup Using Apache:** Install and configure the Apache web server to host the repository. Synchronize the local repository with an official package source. Configure client systems to use the local repository for software installation and updates.

(f) **Client System Configuration:** Update client system repositories to point to the local Apache-hosted repository.Test package installations and updates from the local repository.Monitor repository usage and manage updates to ensure package availability and integrity..

4.3 Tasks

4.3.1 DNS Server Setup

Introduction to DNS The Domain Name System (DNS) is a hierarchical and decentralized naming system used to identify computers, services, and other resources reachable through the internet or private networks. It translates human-readable domain names (like www.example.com) into IP addresses (like 192.0.2.1) that networking equipment uses to locate and identify these resources.
DNS is crucial for the usability and functionality of the internet. Without DNS, users would need to remember complex IP addresses to access websites or online services. DNS ensures that when a user types a domain name into a web browser, the correct IP address is retrieved, and the user can access the desired site or service.

(a) **Planning the DNS Architecture:**
Identify the Purpose: Determine whether the DNS server will be authoritative, caching, or both.
Domain Structure: Define the hierarchy and structure of the domains you will be managing.
Server Resources: Ensure you have adequate hardware and network resources to support the DNS server.

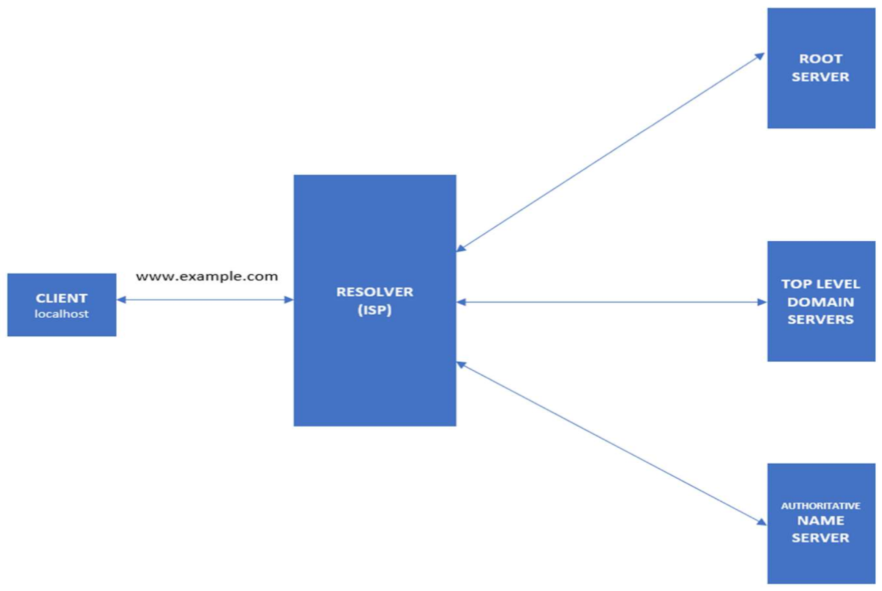


Figure 4.1: DNS Architecture

(b) **Choosing DNS Software:** Select Software: Choose the DNS software that fits your needs. Common choices include BIND (Berkeley Internet Name Domain), Microsoft DNS, and PowerDNS.

Version Compatibility: Ensure the software version is compatible with your operating system and meets your security requirements.

(c) **Configuring DNS Server:** Select Software: Install the DNS software on your server. This process typically involves downloading the software package and running the installer.

Initial Configuration: Configure the basic settings such as the IP address of the server, logging options, and access controls..

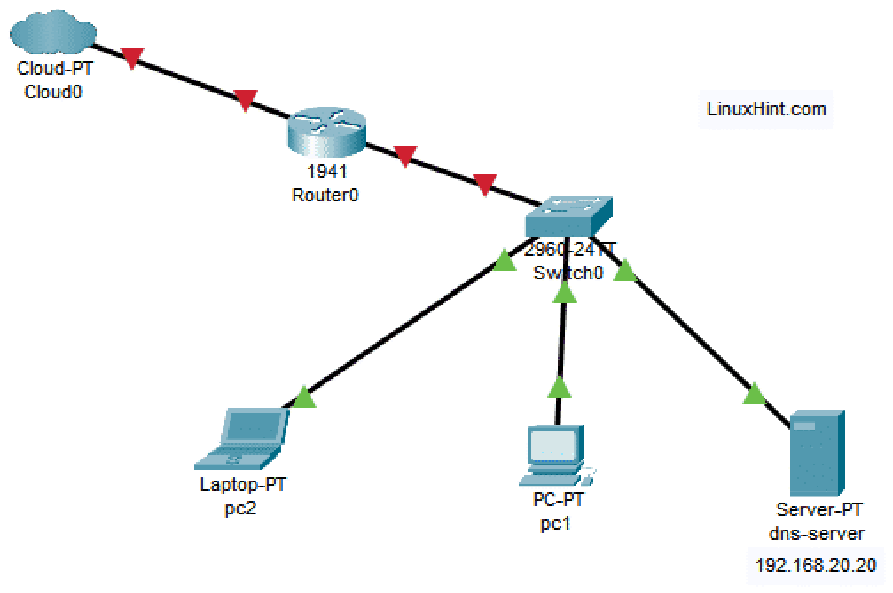


Figure 4.2: Configuring DNS

(d) **Testing and Validation:** Syntax Check: Validate the configuration files and zone files for syntax errors using the tools provided by your DNS software.

Functional Testing: Use DNS lookup tools to test whether your DNS server is correctly resolving domain names.

Performance Testing: Ensure that your DNS server can handle the expected load and performs efficiently.

(e) **Setting Up Forward and Reverse Zones :** Forward Zone: This is where you define mappings from domain names to IP addresses (A and AAAA records).

Reverse Zone: This is where you define mappings from IP addresses to domain names (PTR records).

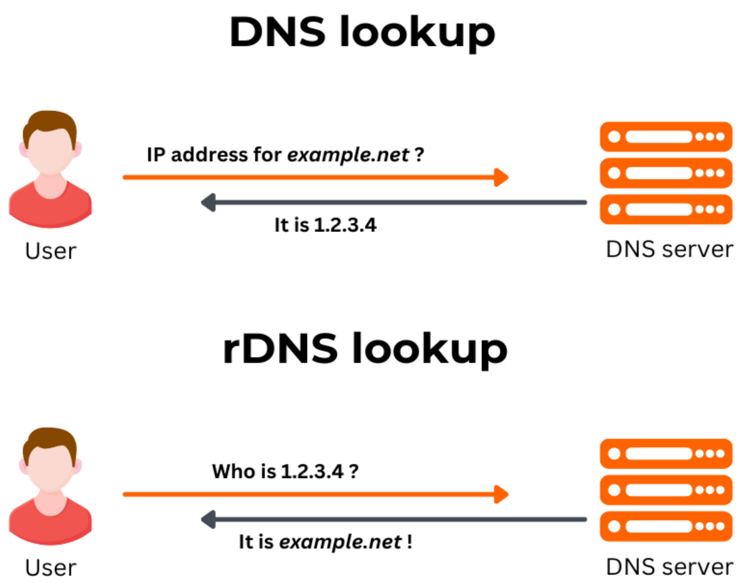


Figure 4.3: setting up reverse zone

4.3.2 Local repo setup by using apache server and that repo should be used by client:

A local repository can be created using Apache to host software packages. By installing Apache on a server and configuring a directory for packages, you can create a central location for clients to access. The client then points its package manager (like yum) to this repository using a configuration file, allowing it to install or update software directly from your local server. This offers benefits like faster downloads and improved control over software distribution.

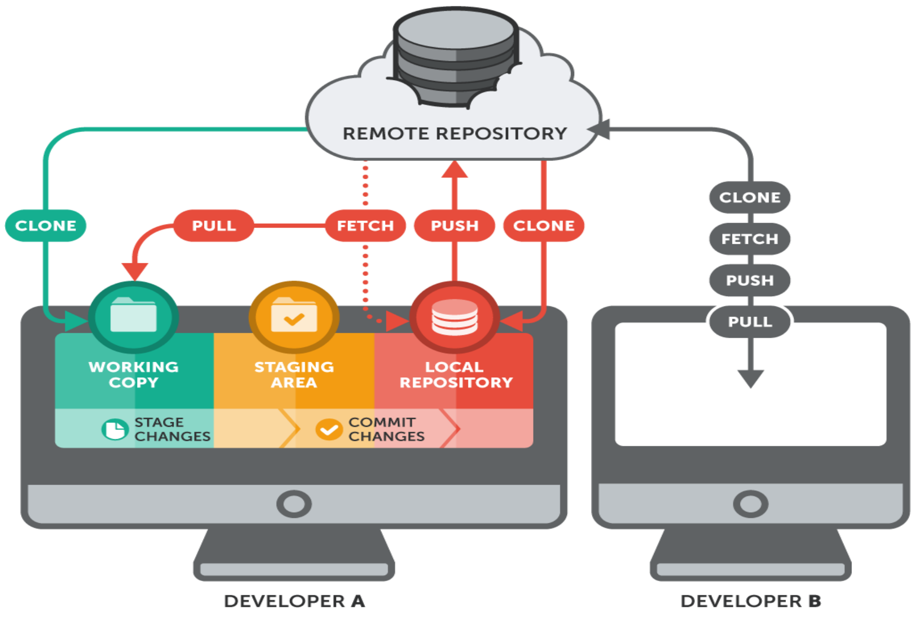
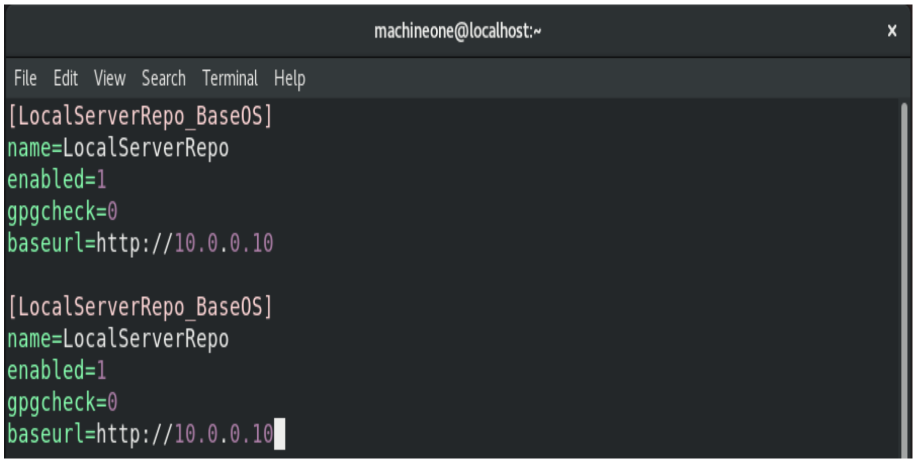


Figure 4.4: Remote repository



```
[LocalServerRepo_BaseOS]
name=LocalServerRepo
enabled=1
gpgcheck=0
baseurl=http://10.0.0.10

[LocalServerRepo_BaseOS]
name=LocalServerRepo
enabled=1
gpgcheck=0
baseurl=http://10.0.0.10
```

Figure 4.5: Local repository

4.3.3 Implement HA-Cluster

The figure 4.7 is a Classification high availability clusters provide continuous service by eliminating single points of failure. These clusters combine multiple servers (nodes) that work together. If one node fails, another healthy node takes over running critical services. This is achieved using cluster management software like Pacemaker, which monitors nodes, manages resource allocation (like databases), and ensures automatic failover for uninterrupted service. This redundancy enhances uptime and ensures applications remain accessible even during server outages.

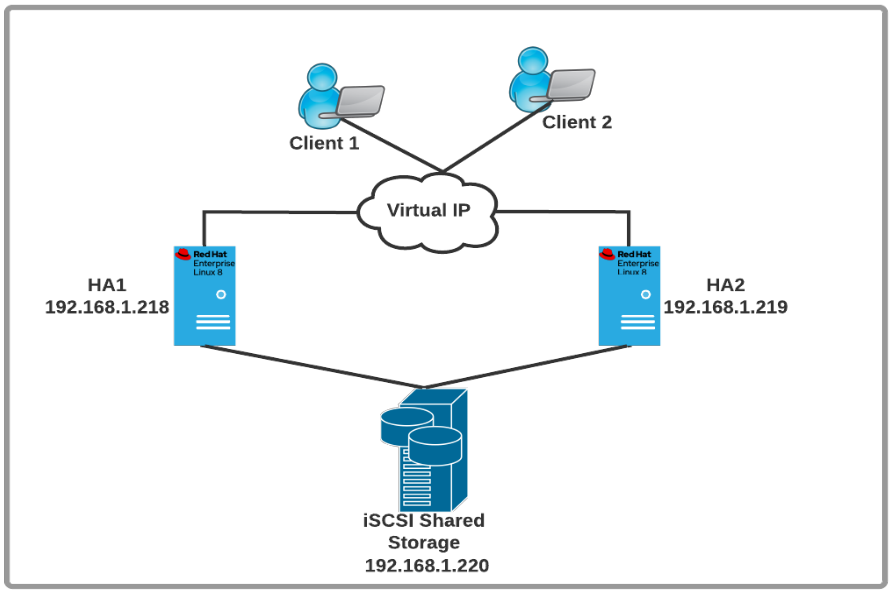


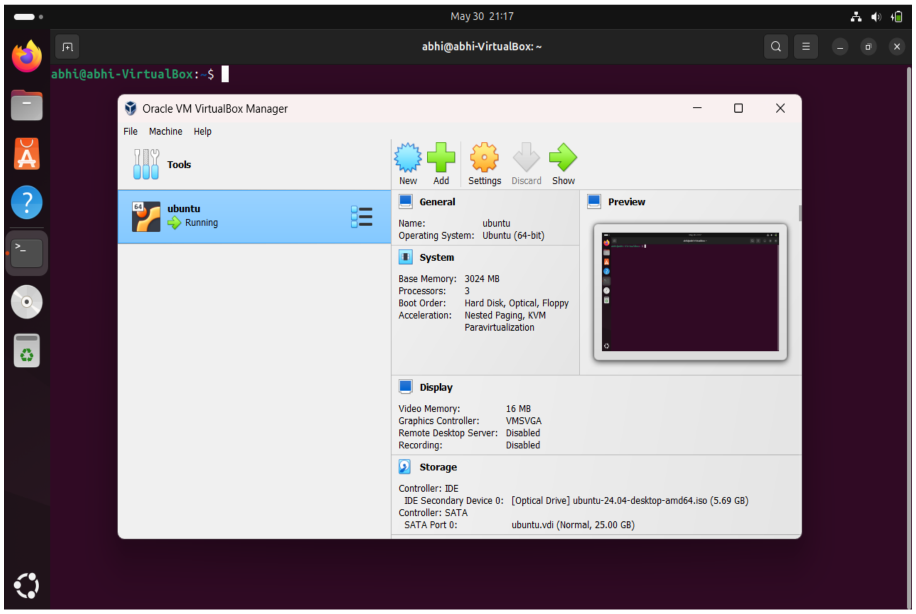
Figure 4.6: Implement HA-Cluster

4.3.4 Automated Monitoring and Maintenance for an E-commerce Website

An e-commerce company operating a Linux-based web server faces significant challenges in maintaining server health and ensuring continuous website availability. Manual monitoring and maintenance are time-consuming and susceptible to human error, jeopardizing reliability and uptime. To address these issues, the company needs an automated solution that can efficiently monitor system performance, execute regular updates, and back up critical data. Implementing such a solution will streamline server management, reduce the risk of downtime, and enhance the overall reliability and availability of the website, allowing the company to focus more on its core business activities and less on technical maintenance tasks.

(a) **Virtualized Environment Setup:**

The e-commerce company should start by setting up a virtualized environment using technologies like VMware, KVM, or VirtualBox. This approach allows multiple virtual machines (VMs) to run on a single physical server, optimizing resource use and simplifying management. Virtualization enhances scalability, enables quick backups and restorations, and provides isolation for different tasks, reducing the risk of system failures. This setup creates a flexible and efficient foundation for automating server management, ensuring the reliability and continuous availability of the website.



```
abhi@abhi-VirtualBox:~$
```

```
General
Name:               ubuntu
Operating System:   Ubuntu (64-bit)

System
Base Memory:   3024 MB
Processors:    3
Boot Order:    Hard Disk, Optical, Floppy
Acceleration:  Nested Paging, KVM
               Paravirtualization

Display
Video Memory:            16 MB
Graphics Controller:     VMSVGA
Remote Desktop Server:   Disabled
Recording:               Disabled

Storage
Controller: IDE
  IDE Secondary Device 0:   [Optical Drive] ubuntu-24.04-desktop-amd64.iso (5.69 GB)
Controller: SATA
  SATA Port 0:              ubuntu.vdi (Normal, 25.00 GB)
```

Figure 4.7: VM Configuration

(b) **Monitoring System Performance:**

The top command in Linux is a powerful tool for real-time system activity monitoring. It provides comprehensive process information, listing all running processes along with their process ID (PID), user, CPU usage, memory usage, and other vital statistics. At the top of the display, top shows a summary of CPU usage, including total CPU usage, and a breakdown of usage by user processes, system processes, and idle CPU time. Additionally, top offers detailed memory usage information, including total memory, used memory, free memory, and memory usage by individual processes. This makes top an essential tool for diagnosing performance issues and monitoring system health.

(c) **Automate System Updates:**

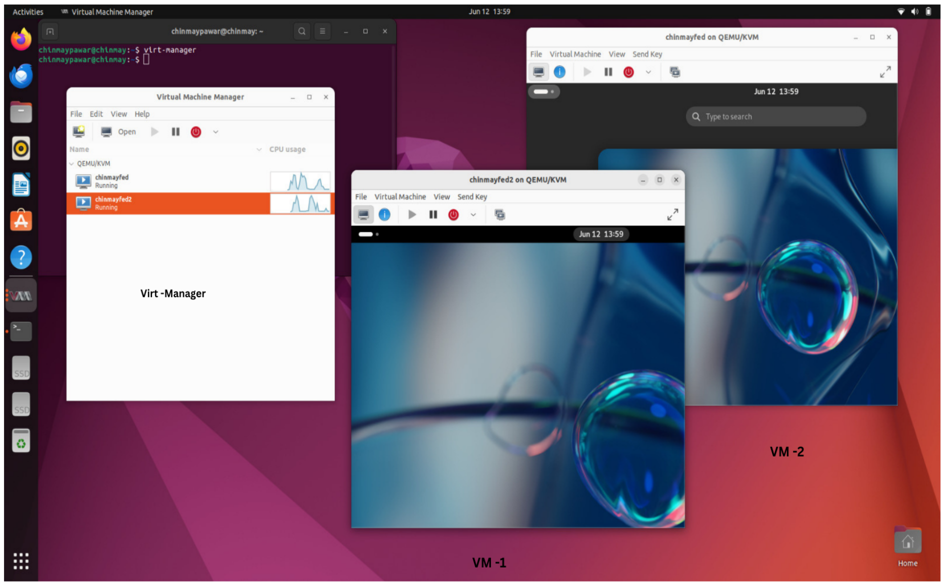
To automate system updates, the e-commerce company should create an update script and schedule it to run regularly. First, they need to create a script named update-system.sh that updates the package lists and upgrades all installed packages, ensuring the system is up-to-date. They should then make the script executable. To schedule this script to run daily at 2 AM, they can use cron by opening the cron table with crontab -e and adding the appropriate scheduling line. This cron job ensures the script runs automatically at the specified time, keeping the system updated without requiring manual intervention.

(d) **Automate Data Backup:**

To automate data backup, the e-commerce company should create a backup script and schedule it to run weekly. First, they need to create a script named backup.sh that archives the website files into a compressed file, ensuring data is safely stored. They should then make the script executable. To schedule this script to run every Sunday at 3 AM, they can use cron by opening the cron table with crontab -e and adding the appropriate scheduling line. This cron job ensures the backup script runs automatically at the specified time, regularly securing the website data without requiring manual intervention..

4.3.5 Virtualization by KVM

The figure 4.9 Kernel-based Virtual Machine (KVM) is an open source virtualization technology built into Linux®. Specifically, KVM lets you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environments called guests or virtual machines (VMs).Creating a KVM virtual machine on Linux involves installing the required packages (e.g., qemu-kvm, libvirt), defining the virtual machine's specifications (CPU, memory, disk, network), and then using a tool like virt-manager or virsh to provision the VM. You can either create a new virtual disk image or use an existing one (ISO or disk image). After configuring the VM, you can install the desired operating system, connect to the VM's console, and start using it just like a physical machine. KVM's tight integration with the Linux kernel allows for efficient virtualization and resource management on the host system.

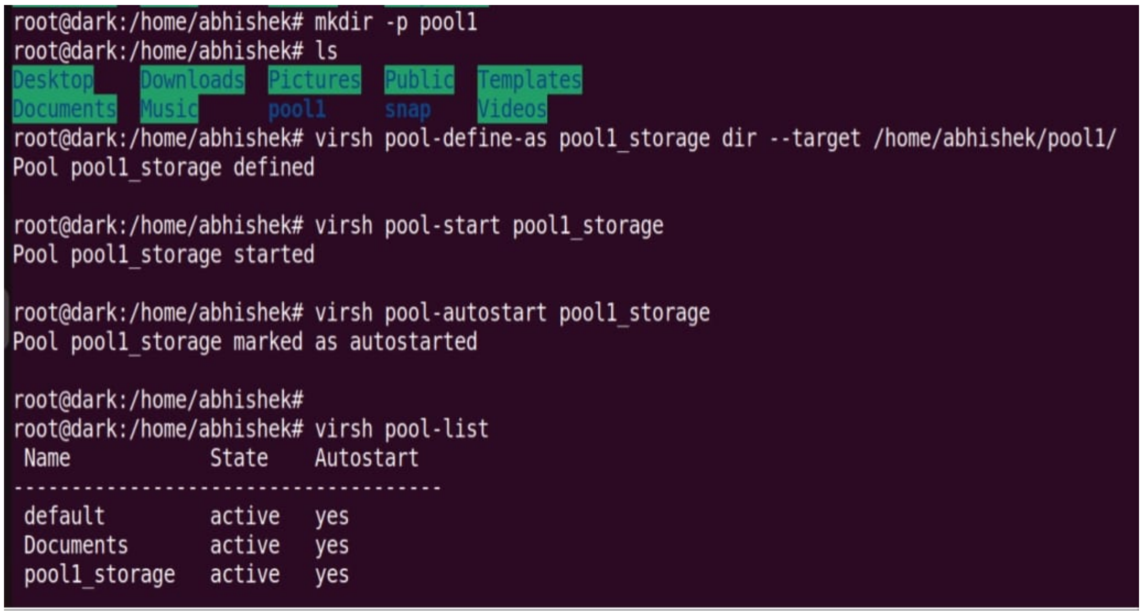


```
chinmaypawar@chinmay:~$ virt-manager
chinmaypawar@chinmay:~$ 
```

Name	CPU usage
QEMU/KVM	
chinmayfed Running	
chinmayfed2 Running	

Figure 4.8: Vm's Created by KVM

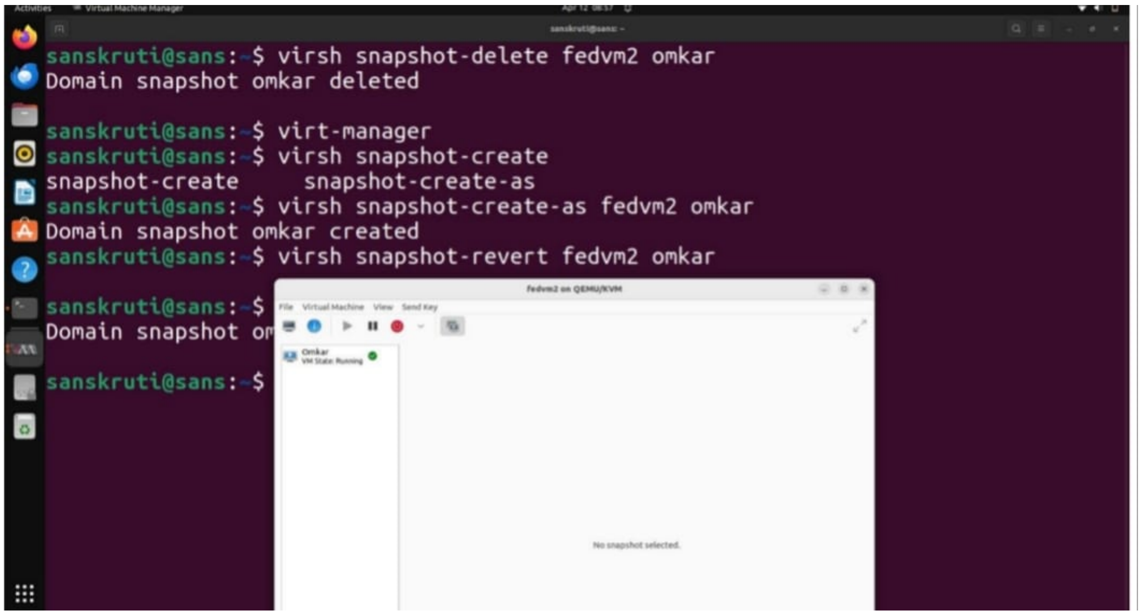
The figure 4.10 is a Storage virtualization in KVM involves abstracting physical storage resources and presenting them as virtualized storage pools that can be efficiently allocated and managed. KVM supports creating storage pools from various backend storage technologies like local disks, network file systems (NFS), iSCSI, and others. These storage pools act as centralized repositories for storing disk images and other data associated with virtual machines. Storage virtualization enables features like thin provisioning, where storage is dynamically allocated as needed, snapshots for creating point-in-time copies, and live migration of VMs between hosts while maintaining storage connectivity. By virtualizing storage, KVM simplifies storage management, improves resource utilization, and provides flexibility in choosing the appropriate storage backend based on performance, availability, and cost requirements.



```
root@dark:/home/abhishek# mkdir -p pool1
root@dark:/home/abhishek# ls
Desktop    Downloads  Pictures  Public  Templates
Documents  Music      pool1     snap    Videos
root@dark:/home/abhishek# virsh pool-define-as pool1_storage dir --target /home/abhishek/pool1/
Pool pool1_storage defined

root@dark:/home/abhishek# virsh pool-start pool1_storage
Pool pool1_storage started

root@dark:/home/abhishek# virsh pool-autostart pool1_storage
Pool pool1_storage marked as autostarted

root@dark:/home/abhishek#
root@dark:/home/abhishek# virsh pool-list
 Name            State    Autostart
-------------------------------------
 default         active   yes
 Documents       active   yes
 pool1_storage   active   yes
```

Figure 4.9: Storage Virtulization

The figure 4.11 is a KVM snapshot, snapshots are a powerful feature that allow you to capture the complete state of a virtual machine at a specific point in time, including its disks, memory, and virtual devices. This creates a frozen copy of the VM's data, which can be used for various purposes, such as creating backups, testing software changes, or rolling back to a previous state if needed. Snapshots are particularly useful in scenarios where you need to revert to a known good state, experiment with different configurations, or simply preserve a specific VM state for future reference. KVM's snapshot functionality leverages copy-on-write technology, which means that only the changes made after the snapshot are stored, minimizing storage overhead. With snapshots, you can quickly restore a VM to its previous state, enabling rapid recovery from failures, data corruption, or other issues, thereby enhancing the overall reliability and resilience of your virtualized infrastructure.



```
sanskruti@sans:~$ virsh snapshot-delete fedvm2 omkar
Domain snapshot omkar deleted

sanskruti@sans:~$ virt-manager
sanskruti@sans:~$ virsh snapshot-create
snapshot-create     snapshot-create-as
sanskruti@sans:~$ virsh snapshot-create-as fedvm2 omkar
Domain snapshot omkar created
sanskruti@sans:~$ virsh snapshot-revert fedvm2 omkar

sanskruti@sans:~$
Domain snapshot om

sanskruti@sans:~$
```

Figure 4.10: Snapshot

Chapter 5

Result and Discussion

The Linux operating system has proven to be a reliable and versatile platform across industries, contributing to effective management of Linux-based assets and infrastructure. Ansible's agentless architecture and automation framework have streamlined processes like vendor onboarding and purchase order management, enhancing transparency and operational efficiency. KVM, integrated with the Linux kernel, has optimized resource allocation and workload consolidation by enabling virtual machine management, facilitating improved efficiency and cost savings. Shell scripting has automated repetitive tasks, streamlined workflows, and integrated systems within the Linux ecosystem. The utilization of Linux, Ansible, KVM, and Shell Scripting has enabled significant improvements in procurement, asset management, inventory control, and overall performance through enhanced transparency, informed decision-making, and financial control.

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

The hands-on learning experience encompassed a diverse range of topics, providing a comprehensive understanding of Linux administration, automation, and virtualization. Mastering shell scripting fundamentals enabled the automation of repetitive tasks, enhancing efficiency and productivity. Implementing a high-availability (HA) cluster facilitated the configuration of failover mechanisms, ensuring business continuity and resilience. Setting up a DNS server involved configuring DNS software, managing zone files, and securing the server against vulnerabilities. Exploring KVM virtualization covered networking and storage aspects, including NAT, bridged networking, bonding for redundancy, and storage management. The local repository setup using Apache allowed for centralized package distribution, streamlining software installation and updates for client systems.equipping learners with the skills and knowledge to effectively manage and maintain Linux-based infrastructure in real-world scenarios.

6.2 Future Scope

The skills and knowledge acquired through these hands-on learning experiences open up a wide range of future opportunities and scope for further growth. With a strong foundation in shell scripting, automation can be extended to more complex scenarios, such as continuous integration and deployment pipelines, enabling streamlined software delivery processes. Proficiency in setting up high-availability clusters paves the way for implementing advanced clustering solutions like Kubernetes, enabling efficient container orchestration and scalability. Expertise in DNS server configuration and management can be leveraged for exploring DNS-based load balancing and traffic management strategies. Additionally, the experience with KVM networking and storage virtualization can be applied to cloud computing environments, enabling seamless workload migration and hybrid cloud architectures. Furthermore, the local repository setup can be expanded to include package mirroring and caching mechanisms, optimizing software distribution across geographically dispersed locations. As technology continues to evolve, these learnings can be built upon to stay ahead of the curve, embracing emerging trends in areas such as serverless computing, edge computing, and artificial intelligence, ultimately positioning learners as versatile and adaptable professionals in the ever-changing IT landscape.

References

[1] Red Hat Certified System Administrator (RHCSA) RH124,RH134

[2] A Red Hat® Certified Engineer (RHCE®) RH294

[3] GFG Linux Tutorials. https://www.geeksforgeeks.org/linux-tutorial/

[4] GFG Virtualization Tutorials. https://www.geeksforgeeks.org/virtualization-cloud-computing-types/

[5] Ansible Galaxy. https://galaxy.ansible.com/ui/

[6] RedHat Training's. https://www.redhat.com/en/services/training/red-hat-academy