RTB | RWANDA TVET BOARD

RQF LEVEL 5



NITLS501

NETWORKING AND INTERNET TECHNOLOGIES

Linux Server Administration

TRAINEE'S MANUAL

October, 2024

LINUX SERVER ADMINISTRATION

KOICA Korea International Cooperation Agency

TQUM TVET Quality Management Project

2024

AUTHOR'S NOTE PAGE (COPYRIGHT)

The competent development body of this manual is Rwanda TVET Board ©, reproduce with permission.

All rights reserved.

- This work has been produced initially with the Rwanda TVET Board with the support from KOICA through TQUM Project
- This work has copyright, but permission is given to all the Administrative and Academic Staff of the RTB and TVET Schools to make copies by photocopying or other duplicating processes for use at their own workplaces.
- This permission does not extend to making of copies for use outside the immediate environment for which they are made, nor making copies for hire or resale to third parties.
- The views expressed in this version of the work do not necessarily represent the views of RTB. The competent body does not give warranty nor accept any liability
- RTB owns the copyright to the trainee and trainer's manuals. Training providers may reproduce these training manuals in part or in full for training purposes only. Acknowledgment of RTB copyright must be included on any reproductions. Any other use of the manuals must be referred to the RTB.

© Rwanda TVET Board

Copies available from:

- *HQs: Rwanda TVET Board-RTB*
- *Web: www.rtb.gov.rw*
- **KIGALI-RWANDA**

Original published version: October 2024

ACKNOWLEDGEMENTS

The publisher would like to thank the following for their assistance in the elaboration of this textbook:

Rwanda TVET Board (RTB) extends its appreciation to all parties who contributed to the development of the trainer's and trainee's manuals for the TVET Certificate V in Networking and Internet Technologies, specifically for the module **"NITLS501: Linux Server Administration."**

We extend our gratitude to KOICA Rwanda for its contribution to the development of these training manuals and for its ongoing support of the TVET system in Rwanda

We extend our gratitude to the TQUM Project for its financial and technical support in the development of these training manuals.

We would also like to acknowledge the valuable contributions of all TVET trainers and industry practitioners in the development of this training manual.

The management of Rwanda TVET Board extends its appreciation to both its staff and the staff of the TQUM Project for their efforts in coordinating these activities

This training manual was developed:

Under Rwanda TVET Board (RTB) guiding policies and directives



Under Financial and Technical support of



COORDINATION TEAM

RWAMASIRABO Aimable

MARIA Bernadette M. Ramos

MUTIJIMA Asher Emmanuel

PRODUCTION TEAM

Authoring and Review

NIYITURAGIYE Vedaste

HABIMANA Joel

Validation

HANYURWIMFURA Dieudonne

MANIRAGUHA Denys

HATEGEKIMANA Vincent

Conception, Adaptation and Editorial works

HATEGEKIMANA Olivier

GANZA Jean Francois Regis

HARERIMANA Wilson

NZABIRINDA Aimable

DUKUZIMANA Therese

NIYONKURU Sylvestre

BYUKUSENGE Protais

Formatting, Graphics, Illustrations, and infographics

YEONWOO Choe

SUA Lim

SAEM Lee

SOYEON Kim

WONYEONG Jeong

SHYAKA Emmanuel

Financial and Technical support

KOICA through TQUM Project

TABLE OF CONTENT

AUTHOR'S NOTE PAGE (COPYRIGHT)---iii

ACKNOWLEDGEMENTS---iv

TABLE OF CONTENT ---vii

ACRONYMS---viii

INTRODUCTION ---1

MODULE CODE AND TITLE: NITLS501 LINUX SERVER ADMINISTRATION ---2

Learning Outcome 1: Prepare Linux server Environment ---3

Key Competencies for Learning Outcome 1 : Prepare Linux server Environment ---4

Indicative content 1.1: Identification of Linux server requirements ---6

Indicative content 1.2: Selection of Tools and Equipment---12

Indicative content 1.3: Installation of Linux OS---18

Learning outcome 1 end assessment ---32

References ---35

Learning Outcome 2: Manage Files and Directories ---36

Key Competencies for Learning Outcome 2 : Manage Files and Directories ---37

Indicative content 2.1: Applying Basic Linux Commands---39

Indicative content 2.2: Creation of Linux files and Directories---55

Indicative content 2.3: Management of User and Groups ---62

Indicative content 2.4: Application of files and directories permissions ---77

Learning Outcome 2 end assessment---83

References ---88

Learning Outcome 3: Manage Linux services---89

Key Competencies for Learning Outcome 3: Manage Linux services---90

Indicative content 3.1: Configuration of DNS Service ---92

Indicative content 3.2: Configuration of DHCP service ---98

Indicative content 3.3: Configuration of File Server Service---102

Indicative content 3.4: Configuration of Web Server ---108

Indicative content 3.5: Configuration of Mail Server Service ---113

Learning outcome 3 end assessment ---119

References ---122

ACRONYMS

ACL: Access Control List
BIOS: Basic Input Output System
CD: Change Directory
CHMOD: Change Mode
CHOWN: Change Ownership
CIFS: Common Internet File System
CPU: Central Processing Unit
DEB: Debian Package
DF: Disk Free
DHCP: Dynamic Host Configuration Protocol
DNF: Dandified YUM
DNS: Domain Name System
DU: Disk Usage
EFI: Extensible Firmware Interface
EXT: Extended File System
FS: File System
FTP: File Transfer Protocol
GID: Group Identifier
GPG: GNU Privacy Guard
GRUB: Grand Unified Bootloader
GUI: Graphical User Interface
HTTP: Hypertext Transfer Protocol
HTTPS: Hypertext Transfer Protocol Secure
I/O: Input/output
IDS: Intrusion Detection System
IMAP: Internet Message Access Protocol
INODE: Index Node
IP: Internet Protocol
IPS: Intrusion Prevention System
IPv4: Internet Protocol version 4
IPv6: Internet Protocol version 6
ISP: Internet Service Provider
JOURNALD: Journal Daemon
KILL: Terminate a Process
KVM: Kernel-based Virtual Machine
LDAP: Lightweight Directory Access Protocol
LOM: Lights Out Management

LS: List Directory Contents
LVM: Logical Volume Manager
LXC: Linux Containers
MAC: Media Access Control
MDA: Mail Delivery Agent
MKDIR: Make Directory
MTA: Mail Transfer Agent
NAS: Network-Attached Storage
NAT: Network Address Translation
NFS: Network File System
NIC: Network Interface Card
OS: Operating System
POP: Post Office Protocol
PS: Process Status
PWD: Print Working Directory
RAID: Redundant Array of Independent Disks
RAM: Random Access Memory
RPM: Red Hat Package Manager
RTB: Rwanda TVET Board
SFTP: Secure File Transfer Protocol
SHA: Secure Hash Algorithm
SMB: Server Message Block
SMTP: Simple Mail Transfer Protocol
SSH: Secure Shell
SSL: Secure Sockets Layer
SYSCTL: System Control
TCP: Transmission Control Protocol
TLS: Transport Layer Security
TQUM Project: TVET Quality Management Project
UDP: User Datagram Protocol
UID: User Identifier
UPS: Uninterruptible Power Supply
YUM: Yellowdog Updater Modified

INTRODUCTION

This trainee's manual includes all the knowledge and skills required in Network and Internet Technologies specifically for the module of **"Linux Server Administration".** Trainees enrolled in this module will engage in practical activities designed to develop and enhance their competencies. The development of this training manual followed the Competency-Based Training and Assessment (CBT/A) approach, offering ample practical opportunities that mirror real-life situations.

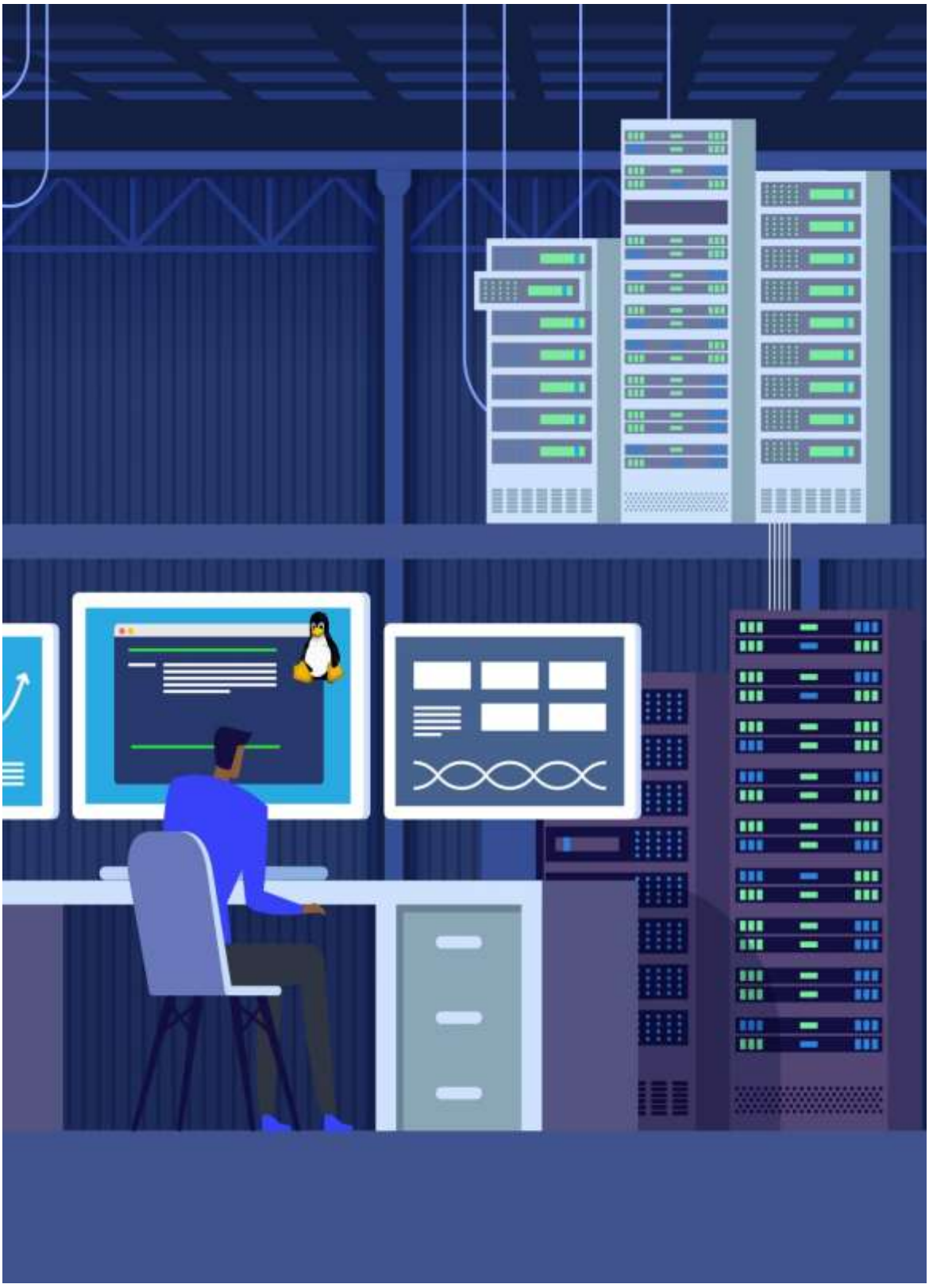
The trainee's manual is organized into Learning Outcomes, which is broken down into indicative content that includes both theoretical and practical activities. It provides detailed information on the key competencies required for each learning outcome, along with the objectives to be achieved.

As a trainee, you will start by addressing questions related to the activities, which are designed to foster critical thinking and guide you towards practical applications in the labor market. The manual also provides essential information, including learning hours, required materials, and key tasks to complete throughout the learning process.

All activities included in this training manual are designed to facilitate both individual and group work. After completing the activities, you will conduct a formative assessment, referred to as the end learning outcome assessment. Ensure that you thoroughly review the key readings and the 'Points to Remember' section.

MODULE CODE AND TITLE: NITLS501 LINUX SERVER ADMINISTRATION

Learning Outcome 1: Prepare Linux server Environment
Learning Outcome 2: Manage Files and Directories
Learning Outcome 3: Manage Linux services



Indicative contents

1.1 Identification of Linux server requirements
1.2 Selection of Tools and Equipment
1.3 Installation of Linux OS

Key Competencies for Learning Outcome 1 : Prepare Linux server Environment

Knowledge	Skills	Attitudes
• Description of Linux server • Description of the Linux server selection criteria	• Selecting Tools and Equipment • Installing Linux OS • Testing Linux OS	• Being Honest • Having Creativity • Having a Teamwork spirit • Being a Problem Solver • Having Patience • Being a Critical thinker



Duration: 20 hrs

Learning outcome 1 objectives:



By the end of the learning outcome, the trainees will be able to:

1. Describe clearly Linux server OS based on server type.
2. Describe correctly Linux server selection based on organization's technical and operational needs.
3. Select properly Tools and Equipment based on system requirements
4. Install properly Linux OS based on business requirements
5. Test properly Linux OS based on business requirements



Resources

Equipment	Tools	Materials
● Computer ● UPS	● Flash disc ● Linux Distributions (Ubuntu, CentOS, Red hat) ● Virtualization software (VMware, Virtual Box, Hyper-V) ● Disk image utility software	● CD/DVD ● Internet ● Ethernet cables

Indicative content 1.1: Identification of Linux server requirements

Duration: 5 hrs

Theoretical Activity 1.1.1: Description of Linux server Operating System

Tasks:

1: Answer the following questions

i. Define Linux server OS

ii. List and explain three key features of Linux server operating systems.

iii. What are the essential components of a Linux server operating system, a nd what roles do they play?

iv. What are the main advantages of using a Linux server operating system o ver other server operating systems?

v. What are some common limitations or challenges associated with Linux s erver operating systems?

vi. What are some common applications or use cases for Linux servers in ent erprise environments?

2: Present your findings to trainer and the whole class or one of your colleagues.

3: Read the Key readings 1.1.1 in this manual.

4: Ask for clarifications if necessary.

Key readings 1.1.1: Description of Linux server Operating System

1. Definition of Linux Server OS

A Linux server OS is an open-source operating system designed to manage servers, applications, and network resources. It provides high performance, stability, and security, which makes it a popular choice for web servers, database servers, and other network services.

2. Key Features of Linux Server Operating Systems

- **Security**: Linux is known for its robust security features, including built-in firewalls (e.g., iptables), user account management, and access controls.
- **Stability and Reliability**: Linux servers are highly stable and can run for long periods without crashing or requiring reboots, making them ideal for critical enterprise applications.
- **Open source and customization**: Linux is open-source, meaning administrators can customize the system to suit their needs. There are various Linux distributions (e.g., Ubuntu, CentOS, Red Hat) tailored to different use cases.

3. **Essential components of a Linux server operating system and their roles are:**
    - **Kernel**: The core component that manages hardware resources and ensures secure communication between hardware and software.
    - **Shell**: Provides a command-line interface for users to interact with the system, execute commands, and run scripts.
    - **File System**: Manages how data is stored, accessed, and organized on storage devices, ensuring file security and efficient access.
    - **Daemon**: Background services (like web servers, mail servers, etc.) that run continuously to perform tasks such as serving web pages, sending emails, or running backups.
4. **Advantages of Using a Linux Server Operating System Over Other Server Operating Systems**
    - **Cost Efficiency**: Linux is open-source and free to use, which significantly reduces costs for businesses compared to proprietary systems like Windows Server.
    - **Security**: Linux offers excellent security features with frequent updates and community support for patches, making it harder for malware and hackers to exploit the system.
    - **Customization and Flexibility**: Linux provides a wide range of distributions and configuration options, allowing users to tailor the system to meet specific needs.
5. **Common Limitations or Challenges Associated with Linux Server Operating Systems**
    - **Steep Learning Curve**: Administrators may need specialized skills and experience, especially for command-line management, which can be difficult for beginners.
    - **Limited Support for Certain Applications**: Some proprietary software and applications may not be natively supported on Linux, requiring workarounds like virtualization or compatibility layers (e.g., Wine).
    - **Hardware Compatibility**: While Linux supports a broad range of hardware, some drivers or hardware-specific features may not be as well-supported as they are on other systems, particularly Windows.
6. **Common Applications or Use Cases for Linux Servers in Enterprise Environments**
    - **Web Hosting and Web Servers**: Linux powers a majority of the world's web servers, often using software like Apache, Nginx, and MySQL.
    - **Database Servers**: Many enterprises use Linux for database servers running PostgreSQL, MySQL, or MongoDB.
    - **Cloud Infrastructure**: Linux is widely used in cloud computing, providing the backbone for platforms like Amazon Web Services (AWS), Google Cloud, and Microsoft Azure.

- **Virtualization**: Linux servers are used to create virtual machines for running multiple instances of operating systems on a single physical machine.



Theoretical Activity 1.1.2: Description of the Linux server selection



Tasks:

1: Answer the following questions:

i. What are the typical hardware requirements for deploying a Linux server?
ii. What software prerequisites must be considered before installing a Linux serve r operating system?
iii. What are the networking requirements for setting up a Linux server?
iv. What are the essential security requirements for a Linux server to protect again st unauthorized access and vulnerabilities?
v. What performance factors should be taken into account when configuring a Linux server?
vi. How does Linux support scalability in server environments, and what considerat ions are needed for scaling up or out?
vii. What measures can be taken to ensure high availability and reliability of Linux s ervers

2: Present your findings to trainer, workshop assistant, the whole class or one of your colleagues.

3: Read the Key readings 1.1.1 in this manual.

4: Ask to the trainer for clarifications if necessary.



Key readings: 1.1.2: Description of the Linux server selection criteria

1. **Typical hardware requirements for deploying a Linux server**
   - **Processor (CPU)**: A multi-core processor is recommended for better performance, especially for handling multiple concurrent users or applications.
   - **Memory (RAM)**: Minimum 1 GB of RAM is typically required for basic operations; however, 4 GB or more is recommended for better performance, especially for running applications or services.

- **Storage**: Sufficient hard disk space for the operating system, applications, and data. A minimum of 10 GB is common, but larger installations may require 100 GB or more.
- **Network Interface Card (NIC)**: At least one NIC is required for network connectivity; multiple NICs may be needed for load balancing or redundancy.
- **Backup Power Supply**: Uninterruptible Power Supply (UPS) is recommended to protect against power failures.

2. **Software requirements to consider before installing a Linux server operating system**
   - **Supported Distribution**: Choose a Linux distribution that meets your requirements (e.g., Ubuntu Server, CentOS, Debian).
   - **Compatibility with Applications**: Ensure that any applications you plan to run are compatible with the chosen Linux distribution.
   - **Software Dependencies**: Identify and install any necessary dependencies or libraries required for your applications before installation.
   - **Updates and Patches**: Check for the latest updates or patches for the Linux distribution to ensure security and stability.
3. **Networking requirements for setting up a Linux server**
   - **Static IP Address**: Assign a static IP address for easier access and management of the server.
   - **Domain Name System (DNS)**: Configure DNS settings for name resolution, allowing users to connect using domain names rather than IP addresses.
   - **Firewall Configuration**: Set up firewall rules to allow or restrict traffic to and from the server.
   - **Network Services**: Depending on the server role, configure necessary network services (e.g., DHCP, FTP, HTTP).
4. **Essential security requirements for a Linux server to protect against unauthorized access and vulnerabilities**
   - **User Account Management**: Implement strong password policies and limit user privileges to reduce the risk of unauthorized access.
   - **Firewall Setup**: Use tools like iptables or firewall to configure firewall rules that restrict unauthorized access to server ports.
   - **Regular Updates**: Keep the operating system and installed software up-to-date with security patches and updates.
   - **Intrusion Detection/Prevention Systems (IDS/IPS)**: Consider implementing tools to monitor for suspicious activity and potential breaches.
5. **Performance factors to consider when configuring a linux server**
   - **CPU and Memory Usage**: Monitor CPU and RAM usage to ensure resources are not being over-utilized, potentially leading to slow performance.

- **Disk I/O**: Optimize disk read/write operations to improve data access speeds. Consider using SSDs for faster performance.
- **Network Bandwidth**: Ensure sufficient network bandwidth for the server's expected traffic load to avoid bottlenecks.
- **Load Balancing**: Distribute traffic across multiple servers to improve performance and avoid overload.

6. **Linux supports scalability in server environments**
   - **Vertical Scaling**: Linux can efficiently utilize hardware upgrades, such as adding more RAM or CPUs to a single server.
   - **Horizontal Scaling**: Multiple Linux servers can be deployed to handle increased load, often using clustering or load-balancing techniques.
   - **Containerization**: Technologies like Docker allow for lightweight, isolated applications that can be easily scaled up or down.
   - **Considerations**: Plan for resource allocation, network configurations, and application architecture to ensure effective scaling.
7. **Measures to ensure high availability and reliability of Linux servers**
   - **Redundancy**: Implement hardware and software redundancy (e.g., RAID, clustering) to minimize downtime in case of failure.
   - **Load Balancing**: Use load balancers to distribute incoming traffic across multiple servers, improving performance and availability.
   - **Regular Backups**: Schedule regular backups to prevent data loss and ensure quick recovery in case of system failure.
   - **Monitoring Tools**: Implement monitoring solutions to track server health and performance, allowing for proactive management and issue resolution.



Points to Remember

Description of Linux server Operating System

- A Linux server OS is an open-source operating system designed to manage servers, applications, and network resources. It provides high performance, stability, and security, which makes it a popular choice for web servers, database servers, and other network services.
- The key features of Linux server operating systems are security, stability and reliability, and open source and customization
- There are essential components of a Linux server operating system such as kernel, shell, file system.
- The main advantages of using a Linux server operating system over other server operating systems are cost efficiency, security, and customization and flexibility.

- Common limitations or challenges associated with Linux server operating systems are steep learning curve, limited support for certain applications and hardware compatibility.
- Common applications or use cases for Linux servers in enterprise environments are web hosting and web servers, database servers, cloud infrastructure, and virtualization.

Description of the Linux server selection

- Typical hardware requirements for deploying a Linux server such processor (CPU), memory (RAM), storage, network interface card (NIC), and backup power supply.
- Software prerequisites to consider before installing a Linux server operating system like choosing a Linux distribution that meets your requirements (e.g., Ubuntu Server, CentOS, Debian), ensure that any applications you plan to run are compatible with the chosen Linux distribution, identifying and installing any necessary dependencies or libraries required for your applications before installation and check for the latest updates or patches for the Linux distribution to ensure security and stability.
- Assign a static IP address for easier access and management of the server, configure DNS settings for name resolution, allowing users to connect using domain names rather than IP addresses, set up firewall rules to allow or restrict traffic to and from the server, depending on the server role, and configure necessary network services (e.g., DHCP, FTP, HTTP).
- Security requirements for a Linux server to protect against unauthorized access and vulnerabilities such as user account management, firewall setup, regular updates and intrusion detection/prevention systems (IDS/IPS)
- Performance factors to consider when configuring a Linux server are CPU and memory usage, Disk I/O, Network bandwidth, and Load balancing.
- Linux supports scalability in server environments like vertical scaling, horizontal scaling, containerization, and considerations
- Ensure high availability and reliability of Linux servers are redundancy, load balancing, regular backups, and monitoring tools.



Application of learning 1.1.

The XYZ Programming Company Ltd intends to deploy a server to manage its services for both students and employees. Before purchasing a Linux server, you have been asked to provide advice and draft the technical specifications for the server to ensure it meets the company's needs and operational requirements

Indicative content 1.2: Selection of Tools and Equipment

Duration: 5 hrs

Theoretical Activity 1.2.1: Identification of Tools and Equipment

Tasks:

1: answer the following questions:

i. What are the minimum hardware specifications required for a basic Linux server setup?

ii. What Linux distributions are commonly recommended for server use, and what are their distinguishing features?

iii. Give three examples of Physical equipment are necessary to set up a Linux server environment?

2: Present your findings to the whole class or one of your colleagues.

3: Read the Key readings 1.2.1 in this manual.

4: Address any questions or concerns

5: Ask to the trainer for clarifications if necessary.

Key readings: 1.2.1: Identification of tools and equipment:

1. Server Hardware

- **Processor (CPU)**: Choose a multi-core processor to handle multiple concurrent tasks efficiently. Intel Xeon and AMD EPYC processors are popular choices for server environments due to their performance and scalability.
- **Memory (RAM)**: Aim for at least 4 GB of RAM for basic server functions, but 8 GB or more is recommended for more demanding applications, such as database management or hosting multiple services.
- **Storage**:
  - **Type**: Consider using SSDs for faster read/write speeds and better performance, especially for applications that require quick data access.
  - **Capacity**: Assess storage needs based on anticipated data volume; a minimum of 100 GB is common, but this may vary based on specific applications and data requirements.
  - **RAID Configuration**: Implement RAID (Redundant Array of Independent Disks) for data redundancy and improved performance. RAID 1 (mirroring) or RAID 5 (striping with parity) are popular configurations.
- **Network Interface Card (NIC)**: A Gigabit NIC is essential for efficient network connectivity. Consider dual NICs for redundancy and load balancing if necessary.

2. Physical Equipment

- **Rackmount Server**: Opt for a rackmount server for space efficiency and better airflow in a data center environment. Ensure it meets the required specifications for processing power and memory.
- **Network Switch**: A managed switch allows for better control over network traffic and provides features like VLAN support, monitoring, and enhanced security.
- **Uninterruptible Power Supply (UPS)**: A UPS is critical to protect against power outages and surges. It provides backup power to the server, allowing for safe shutdowns and preventing data loss.

3. Software Tools

- **Linux Distribution**: Select a stable Linux distribution tailored for server use, such as:
  - **Ubuntu Server**: User-friendly and well-documented, ideal for beginners.
  - **CentOS**: Known for stability and long support cycles, suitable for enterprise environments.
  - **Debian**: Offers a robust and customizable platform, great for production systems.
- **Web Server Software**: Choose web server software based on your hosting needs. Apache and Nginx are popular choices, each with its strengths (Apache for extensive features and Nginx for performance).
- **Database Management System (DBMS)**: Select a DBMS based on the applications you plan to run. MySQL and PostgreSQL are commonly used for web applications and data management.
- **Backup Solutions**: Implement automated backup solutions such as:
  - **rsync**: For file synchronization and backups.
  - **Bacula**: A more comprehensive backup solution for large environments.
- **Performance Monitoring Tools**: Utilize tools like:
  - **Nagios**: For system and network monitoring.
  - **Zabbix**: For monitoring the performance of servers, applications, and network devices.

4. Security Measures

- **Firewalls**: Configure a firewall (like iptables or firewalld) to control incoming and outgoing traffic based on predetermined security rules.
- **Intrusion Detection Systems (IDS)**: Consider using an IDS (such as Snort) to monitor network traffic for suspicious activity.
- **Regular Updates**: Ensure that the Linux distribution and all installed software receive regular updates to mitigate security vulnerabilities.

5. Documentation and Support

- **Documentation**: Maintain thorough documentation of the server setup, configurations, and procedures for troubleshooting. This is essential for future maintenance and updates.

- **Community and Vendor Support**: Evaluate the availability of community support (forums, documentation) or vendor support if using a commercial distribution. This can be crucial for resolving issues quickly.

6. Testing and Validation

- **Test Environment**: Consider setting up a test environment to validate the server setup before deploying it in production. This can help identify any potential issues with hardware compatibility or software configurations.



Practical Activity 1.2.2: Selecting tools and equipment for a Linux web server setup installation

Task:

1: Individually, Perform the following task:

You have been tasked by a school to identify and select the appropriate tools and equipment for setting up and managing a Linux server for a small office.

2: Read the key reading 2.2.1

3: select tools and equipment used to set and manage Linux server

4: Present your work to the trainer, workshop assistant or your classmate

5: Ask clarification, or concerns if any and where necessary

6: Perform the task provided in application of learning 2.2



Key readings: 1.2.2:

Selecting tools and equipment for a Linux web server

1. Server Hardware

This includes the physical equipment on which the Linux web server will run.

- **CPU**: Choose a server with a processor capable of handling expected web traffic (Intel Xeon or AMD EPYC).
- **RAM**: Sufficient memory is necessary for smooth operation, typically starting at 8GB or more, depending on the traffic.
- **Storage**: provide faster read/write speeds for hosting websites and databases.
- **Network Interface Card (NIC)**: High-speed Ethernet cards (e.g., 10Gbps) ensure fast data transfer between server and clients.

2. Linux Distribution

The operating system that runs the server. Common Linux distributions for web servers include:

- **Ubuntu Server**: User-friendly and widely supported, with a large community.
- **CentOS**: Known for stability, especially for enterprise applications.

- **Red Hat Enterprise Linux (RHEL)**: Commercial option with professional support, ideal for critical environments.
- **Debian**: Known for its stability and extensive repository of software packages.

3. **Web Server Software**
   - **Apache HTTP Server**: One of the most popular web server platforms, open-source and highly configurable. And it is suitable for hosting dynamic websites and handling large traffic.
   - **Nginx:** A lightweight, high-performance web server, ideal for static content and reverse proxy services, efficient memory usage, load balancing, and reverse proxy capabilities. And often used for serving static content or as a reverse proxy for handling heavy traffic.
4. **Database Management Systems (DBMS)**
   - **MySQL**: One of the most widely used relational databases, open-source with a large community. And suitable for dynamic web applications that need to manage structured data.
   - **PostgreSQL**: An advanced open-source database known for its scalability, performance, and extensibility. And ideal for complex data handling and enterprise applications.
5. **Domain Name System (DNS) Tools**
   - **BIND**: A widely used DNS server that manages domain name resolution and critical for mapping domain names to IP addresses and handling DNS queries.
   - **Dnsmasq**: Lightweight DNS forwarder and DHCP server and ideal for small to medium-sized web server setups.
6. **Security Tools**
   - **UFW (Uncomplicated Firewall)**: A user-friendly firewall configuration tool for managing firewall rules on Ubuntu and Debian-based servers. And essential for restricting access to the server and securing web traffic.
   - **iptables**: A powerful firewall tool that is part of the Linux kernel, used to filter network traffic. And provides more advanced options for controlling incoming and outgoing traffic.
   - **Fail2ban**: A security tool that scans log files and bans IP addresses showing signs of malicious activity (e.g., brute force attacks). And helps protect the server from unauthorized access attempts.
7. **Backup Tools**
   - **Rsync**: A powerful file synchronization and backup tool. And automates the process of copying and synchronizing files and directories between servers or local/remote locations.
   - **Duplicity**: Backup software that supports encrypted and incremental backups. And ideal for creating encrypted, secure backups of server data to remote storage.

8. Performance Monitoring Tools

- **Nagios**: A widely used open-source system and network monitoring tool. And monitors server health, alerts you to performance issues, and ensures uptime.
- **Zabbix**: A full-featured open-source monitoring tool for networks and servers and provides detailed performance metrics and alerting for system resources like CPU, memory, and disk usage.
- **HTOP**: An interactive system monitor tool for real-time tracking of server performance and shows a dynamic overview of processes, memory usage, and system load.

9. Configuration Management Tools

- **Ansible**: An open-source automation tool for configuration management, deployment, and orchestration and automates the installation, configuration, and management of server software, ensuring consistent server environments.
- **Puppet**: A powerful configuration management tool for automating server infrastructure and helps manage multiple servers and keep their configurations consistent over time.

10. Web Hosting Control Panels

- **cPanel**: A paid, user-friendly control panel for managing web hosting, email, databases, and security settings and simplifies the management of web hosting environments, making it easier to control multiple websites and domains.
- **Webmin**: A free, open-source control panel for Linux server administration and offers a graphical interface for managing Linux systems, ideal for those less comfortable with command-line operations.

11. Virtualization and Containerization Tools

- **Docker**: A containerization platform for deploying and managing applications in isolated containers and helps run web applications in a portable, scalable way, with minimal overhead.
- **KVM (Kernel-based Virtual Machine)**: A full virtualization solution for Linux on x86 hardware and allows you to create and manage virtual machines on your Linux server, useful for hosting multiple isolated environments.



Points to Remember

Identification of tools and equipment used for Linux server environment.

- A basic Linux server setup requires a dual-core processor, at least 1 GB of RAM, a minimum of 10 GB of storage, and a network interface card for connectivity.
- Recommended Linux distributions for server use include Ubuntu Server for its user-friendliness, CentOS for its enterprise-grade stability, and Debian for its robust and customizable nature.
- Essential physical equipment includes a rackmount server for efficient housing, a network switch for connectivity, and an uninterruptible power supply (UPS) to ensure continuous operation during power outages.

While Selecting tools and equipment for a Linux server setup installation you consider

- Hardware tools such as sufficient CPU, RAM, and storage capacity.
- Software tools such as Linux distribution (like Ubuntu Server or CentOS), automated backup solutions (such as rsync or Bacula) , and performance monitoring tools (like Nagios or Zabbix) to ensure optimal operation.
- Physical equipment such as rackmount server for efficient housing, a network switch for connectivity, and an uninterruptible power supply (UPS) to ensure continuous operation during power outages.



Application of learning 1.2.

You have been hired as an IT consultant by a small company that is setting up a Linux-based server to host their internal website, manage employee data, and provide secure file sharing between departments. Your task is to identify and recommend the tools and equipment necessary to set up and manage this Linux server environment.

Indicative content 1.3: Installation of Linux OS

Duration: 10 hrs

Practical Activity 1.3.1: Installing Ubuntu Linux Server OS

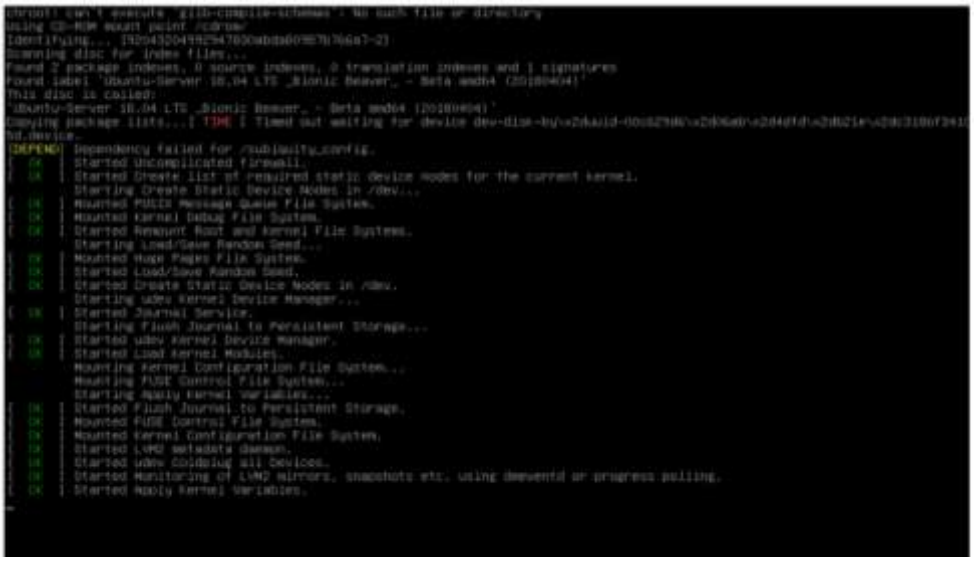
Task:

1**:** Individually, you are requested to perform the following task:
You are hired by NYK School located in Ruhango district to install Linux operating system in their computers, which will be used in teaching.
2: Read the key reading 1.3.1
3: Perform installation process
4: Present your work to the trainer or your classmate
5: Perform the task provided in application of learning 1.3

Key readings: 1.3.1: Installing Ubuntu Linux Server OS

1. **Boot from install media**
   - Put the Ubuntu DVD into your DVD drive (or insert the USB stick or other install media).
   - Restart your computer.

   After a few moments, you should see messages like those shown below on the screen

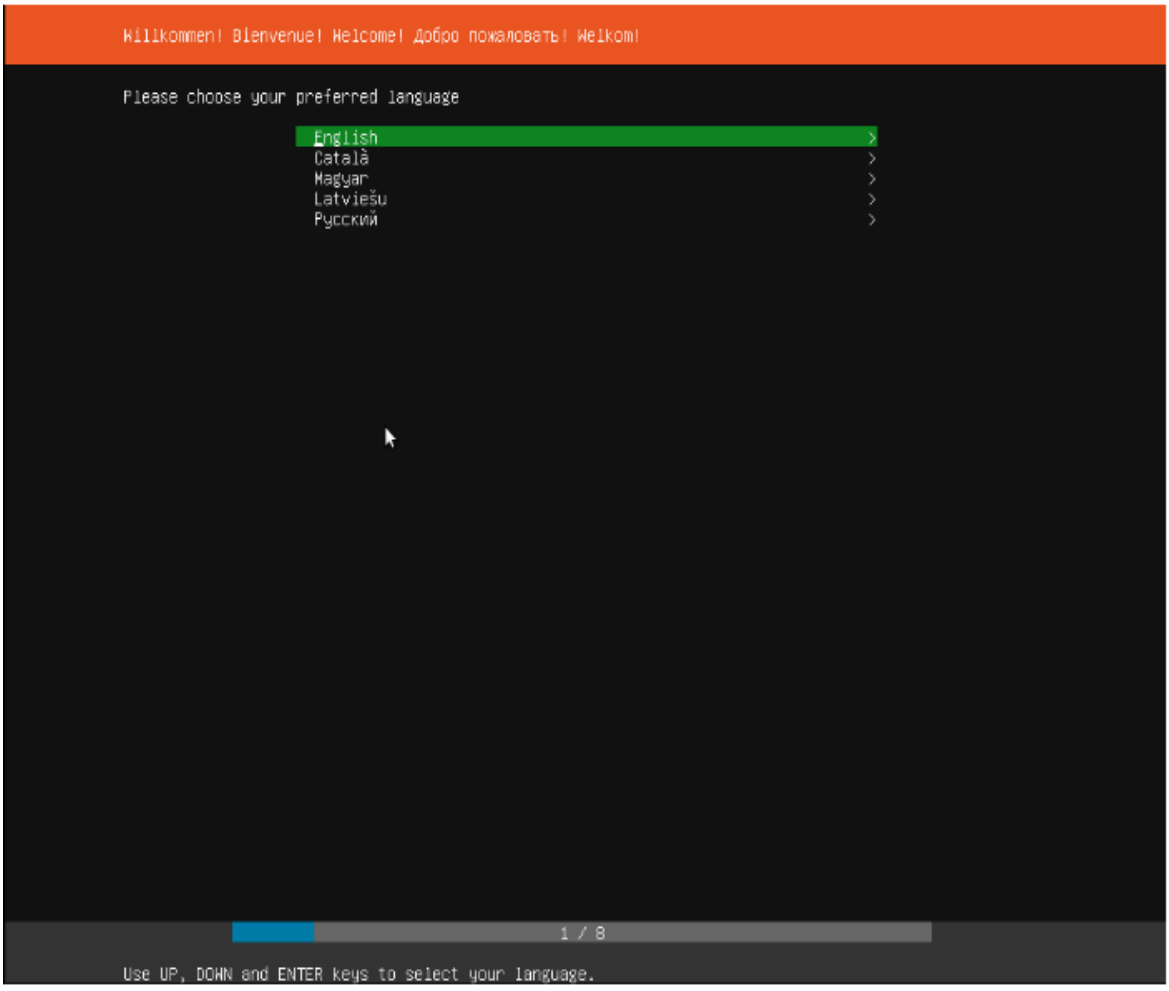


Most computers will automatically boot from USB or DVD, though in some cases this is disabled to improve boot times. If you don't see the boot message and the "Welcome" screen which should appear after it, you will need to set your computer to boot from the install media.

There should be an on-screen message when the computer starts telling you what key to press for settings or a boot menu. Depending on the manufacturer, this could be Escape, F2, F10 or F12. Simply restart your computer and hold down this key until the boot menu appears, then select the drive with the Ubuntu install media.

2. **Choose your language**

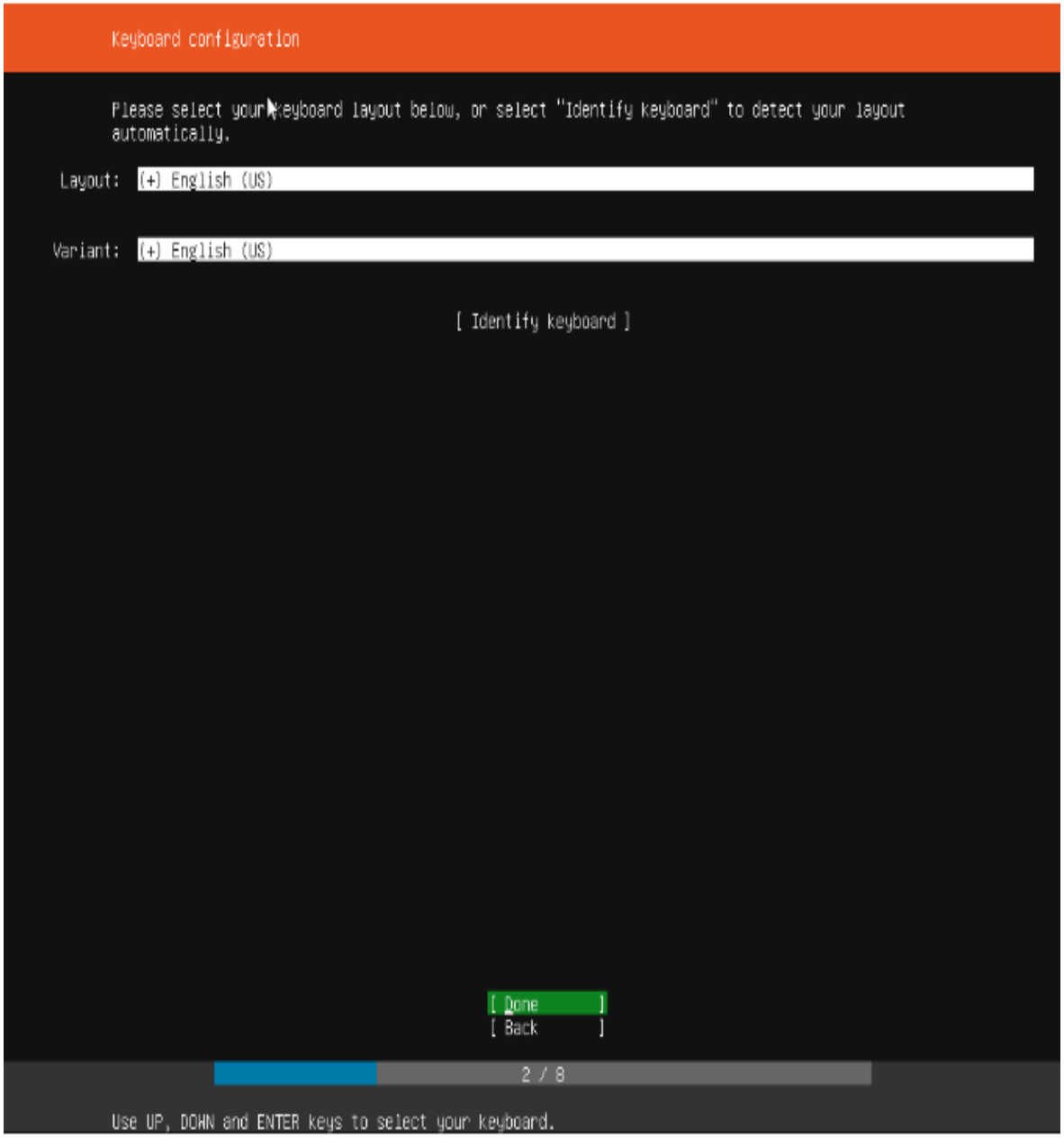
After the boot messages appear, a 'Language' menu will be displayed.



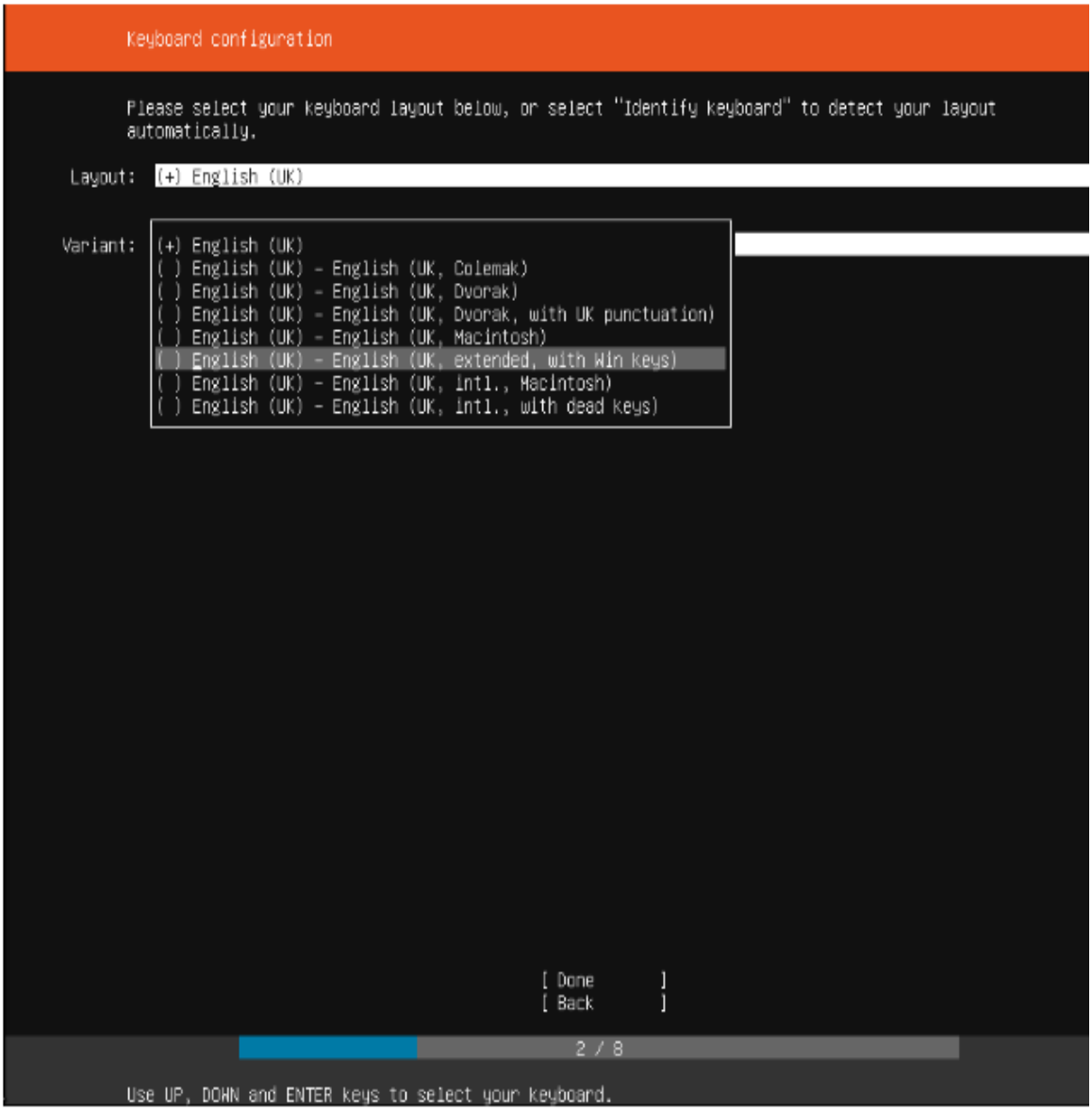
As the message suggests, use the Up, Down and Enter keys to navigate through the menu and select the language you wish to use.

3. **Choose the correct keyboard layout**

Before you need to type anything in, the installer will next display a menu asking you to select your keyboard layout and, if applicable, the variant.

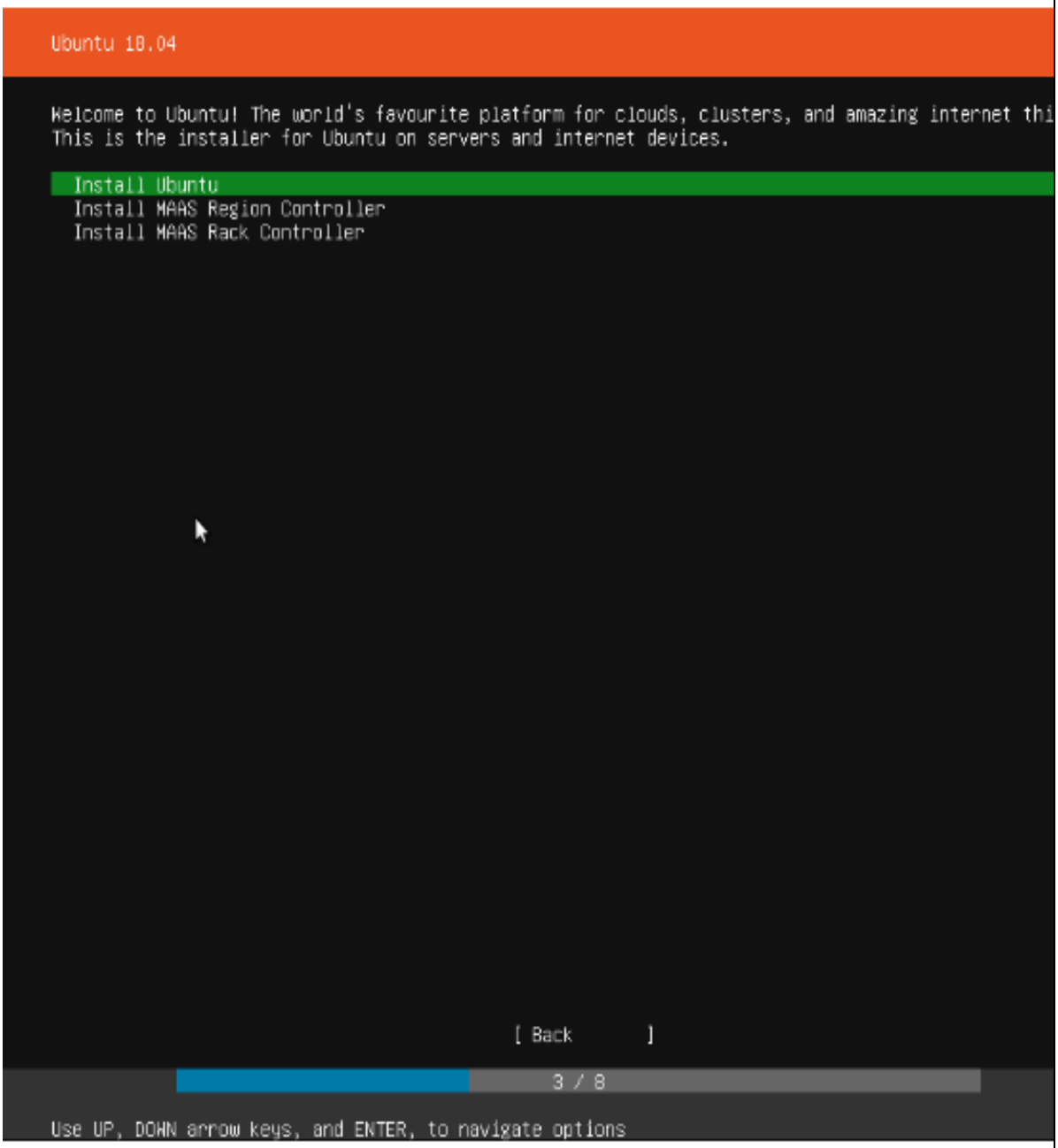


If you don't know which particular variant you want, just go with the default - when Ubuntu Server has been installed you can test and change your preferences more easily if necessary.



4. Choose your install

Now we are ready to select what you want to install. There are three options in the menu :



The bottom two options are used for installing specific components of a Metal As A Service (MAAS) install.

5. **Networking**

The installer will automatically detect and try to configure any network connections via DHCP.



This is usually automatic and you will not have to enter anything on this screen, it is for information only

6. Configure storage

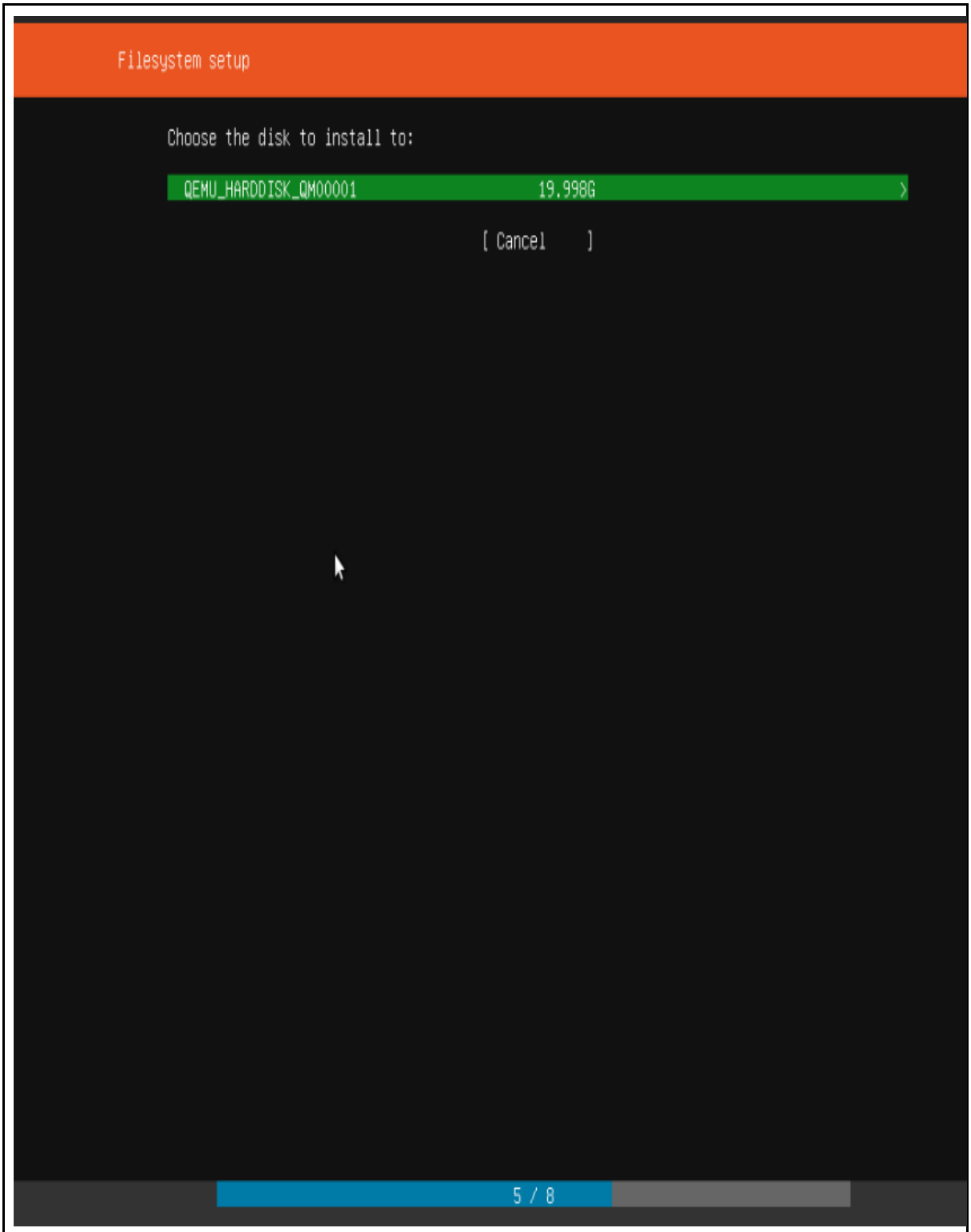
The next step is to configure storage. The recommended install is to have an entire disk or partition set aside for running Ubuntu



If you need to set up a more complicated system, the manual option will allow you to select and reorganise partitions on any connected drives.

7. Select a device

This menu will allow you to select a disk from the ones detected on the system.



To help identification, the drives will be listed using their system ID. Use the arrow keys and enter to select the disk you wish to use.

8. Confirm partitions

With the target drive selected, the installer will calculate what partitions to create

and present this information…



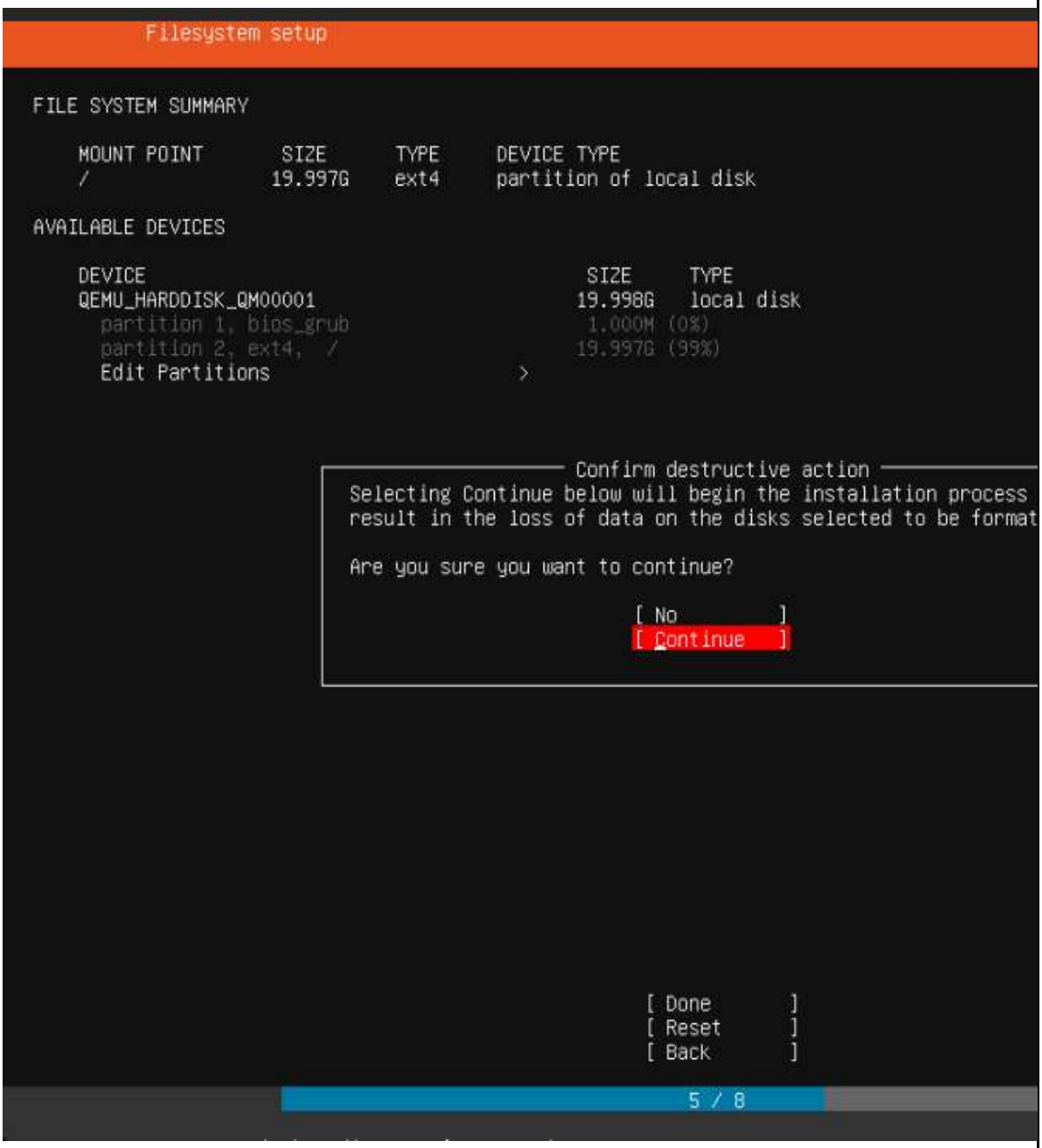
If this isn't what you expected to see (e.g., you have selected the wrong drive), you should use the arrow keys and enter to select Back from the options at the bottom of the screen. This will take you back to the previous menu where you can select a different drive.

It is also possible to manually change the partitions here, by selecting Edit Partitions. Obviously you should only select this if you are familiar with how partitions work.

When you are happy with the disk layout displayed, select Done to continue.

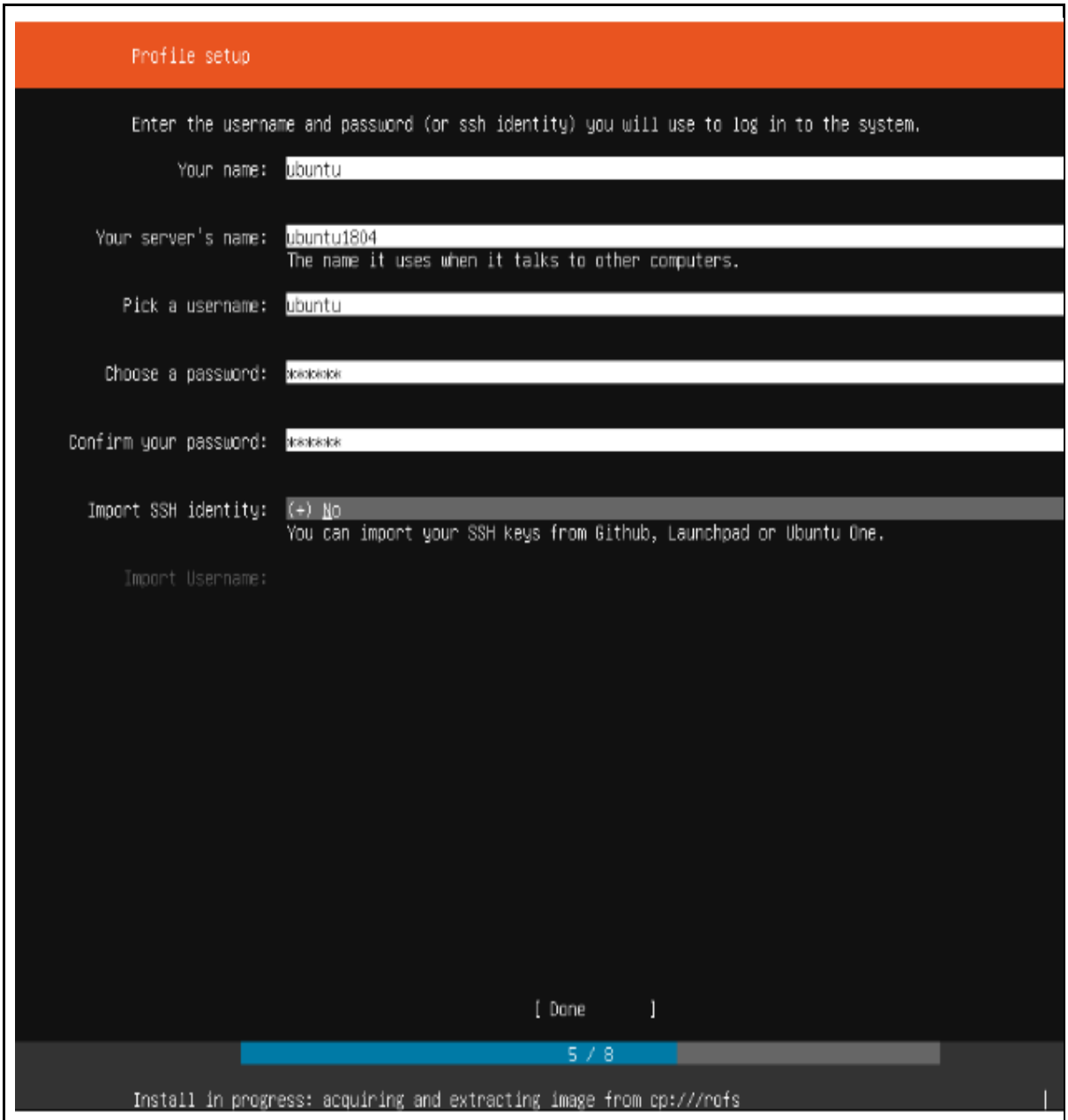
9. Confirm changes

Before the installer makes any destructive changes, it will show this final confirmation step. Double check that everything looks good here and you aren't about to reformat the wrong device!



10. Set up a Profile

The software is now being installed on the disk, but there is some more information the installer needs. Ubuntu Server needs to have at least one known user for the system, and a hostname. The user also needs a password.



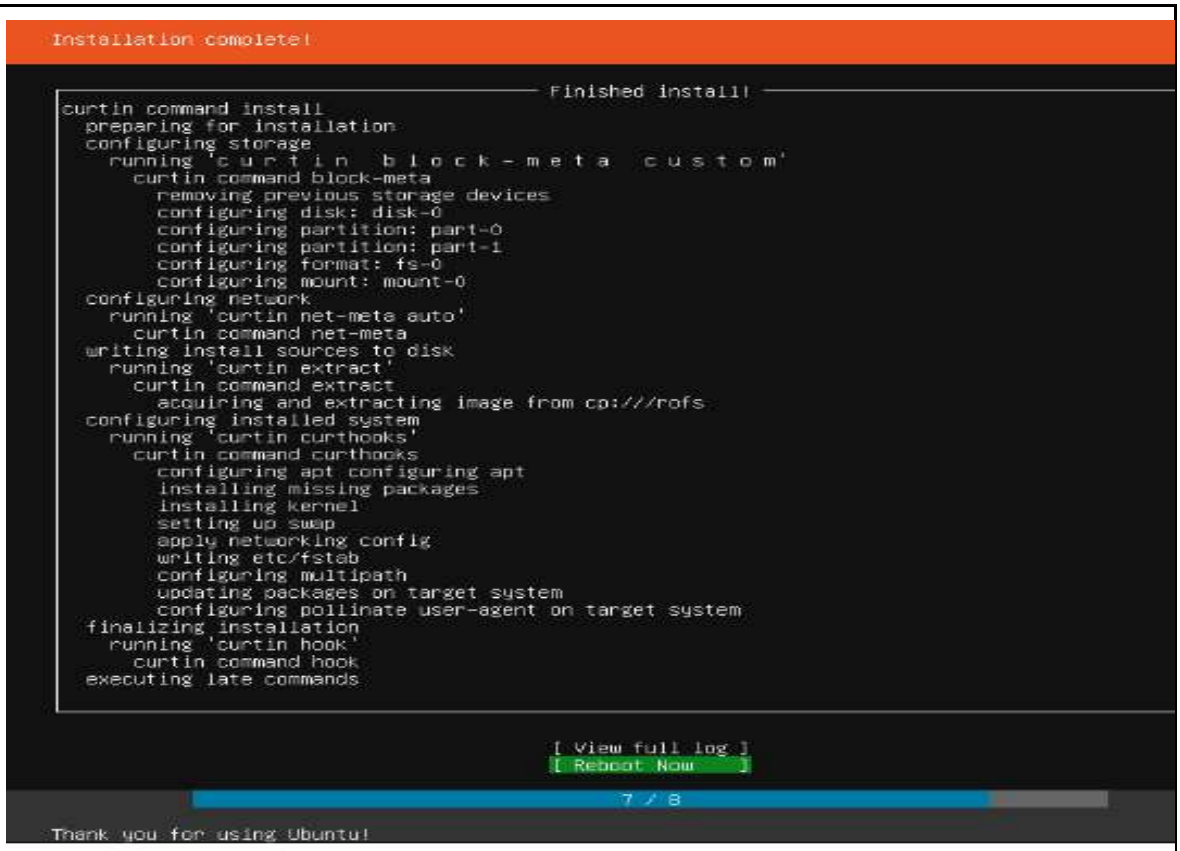
11. Install software

Once you have finished entering the required information, the screen will now show the progress of the installer. Ubuntu Server now installs a concise set of useful software required for servers. This cuts down on the install and setup time dramatically. Of course, after the install is finished, you can install any additional software you may need.



12. Installation complete

When the install is complete, you will see a message like this on the screen.



Remember to remove the install media, and then press enter to reboot and start the server. Welcome to Ubuntu!

Points to Remember

While installing Ubuntu Linux server OS you pass through the following steps:

Step1: Boot from install media

Step2: Choose your language

Step3: Choose the correct keyboard layout

Step4: Choose your install

Step5: Networking

Step6: Configure storage

Step7: Select a device

Step8: Confirm partitions

Step9: Confirm changes

Step10: Set up a Profile

Step11: Install software

Step12: Installation complete



Application of learning 1.3.

The school wants to install the Ubuntu Linux Server OS on its staff's and students' computers, and it has chosen you to handle the installation. This project will enhance the school's technological capabilities and provide a reliable and flexible system for both students and staff. Your task is to install the Linux operating system.



Learning outcome 1 end assessment

Written assessment

Choose the correct answer:

1. Linux server OS is:

a) A system used for personal computing tasks like web browsing and gaming.
b) An open-source operating system used to manage server resources, applications, and services.
c) A proprietary software system for running databases and spreadsheets.
d) A system exclusively designed for mobile phones.

2. The main components of a Linux server operating system are:

a) Linux Kernel, Shell, and Package Management System.
b) Browser, File Explorer, and Antivirus Program.
c) Network Drivers, Email Client, and Social Media Apps.
d) Word Processor, Image Editor, and Video Player.

3. What are the three key features of Linux server operating systems?

a) Multitasking, Security, and Open-Source License.
b) Video Streaming, File Compression, and Gaming Support.
c) Search Engine Optimization, Social Media Integration, and Cloud Storage.
d) Proprietary License, No Customization, and High Cost.

4. How does the open-source nature of Linux contribute to its features and capabilities?

a) It restricts the usage of Linux to only a few certified developers.
b) It allows anyone to view, modify, and contribute to the source code, enhancing flexibility and security.
c) It makes Linux harder to customize or improve over time.
d) It provides Linux with a fixed set of unchangeable features.

5. Which Linux distribution is known for its stability and is commonly used in enterprise environments?

a) Ubuntu
b) CentOS
c) Fedora
d) Arch Linux

6. What is the primary purpose of the **iptables** tool in Linux?
   a) To manage software packages
   b) To configure network interfaces
   c) To manage firewall rules
   d) To monitor system performance

7. Which web server software is known for its lightweight design and high performance?
   a) Apache
   b) Nginx
   c) IIS
   d) Tomcat

8. What tool would you use for secure remote command-line access to a Linux server?
   a) FTP
   b) Telnet
   c) SSH
   d) HTTP

9. Which tool is commonly used for monitoring server performance and resource usage?
   a) MySQL
   b) Nagios
   c) Docker
   d) Git

10. Which database management system is open-source and widely used with Linux servers?
   a) Microsoft SQL Server
   b) Oracle DB
   c) MySQL
   d) IBM Db2

11. Read the following statement and fill the correct answer from the brackets:

- A Linux server operating system is designed to manage _____ (**hardware resources, software**) and provide various network services such as file sharing, _____ (**website hosting, File hosting**), and database management.
- The Linux server OS consists of several main components, including the _____ (**Kernel, Process**), which is the core of the system, and the _____ (**File, Shell),** which allows users to interact with the system.

- **Match the following terms with their correct descriptions:**

Term	Description	Answers
1. Disk Partitioning	A. Software used for configuring and managing firewall rules.	**1……..**
2. File System	B. A process that divides a disk into separate, isolated regions.	**2……..**
3. Swap Space	C. The area in memory that Linux uses as virtual memory to improve performance.	**3……..**

4. LVM (Logical Volume Manager)	D. Used for organizing and storing files on disk.	**4……**
5. Network Interface Configuration	E. A virtualized partitioning system that allows for dynamic disk management.	**5……**
6. SSH (Secure Shell)	F. Command-line tool used to securely access a remote server.	**6……**
7. Package Manager	G. A system used to install, update, and remove software packages in Linux.	**7……**
8. Firewall Configuration	H. The process of configuring network interfaces such as IP addresses and gateways.	**8……**

Practical assessment

The Tech Innovators Company is a software development company which has around 50 employees, including developers, project managers, and support staff. It wants to migrate their internal systems and applications to Ubuntu Linux Server due to its stability, cost-effectiveness, and flexibility. You are hired to Install Ubuntu Linux Server OS which will help the company to serve its customers.



References

Nemeth, E. (2017). *UNIX and Linux system administration handbook* (5th ed.). Prentice Hall.

Nemeth, E. (2020). *Linux administration handbook* (6th ed.). Prentice Hall.

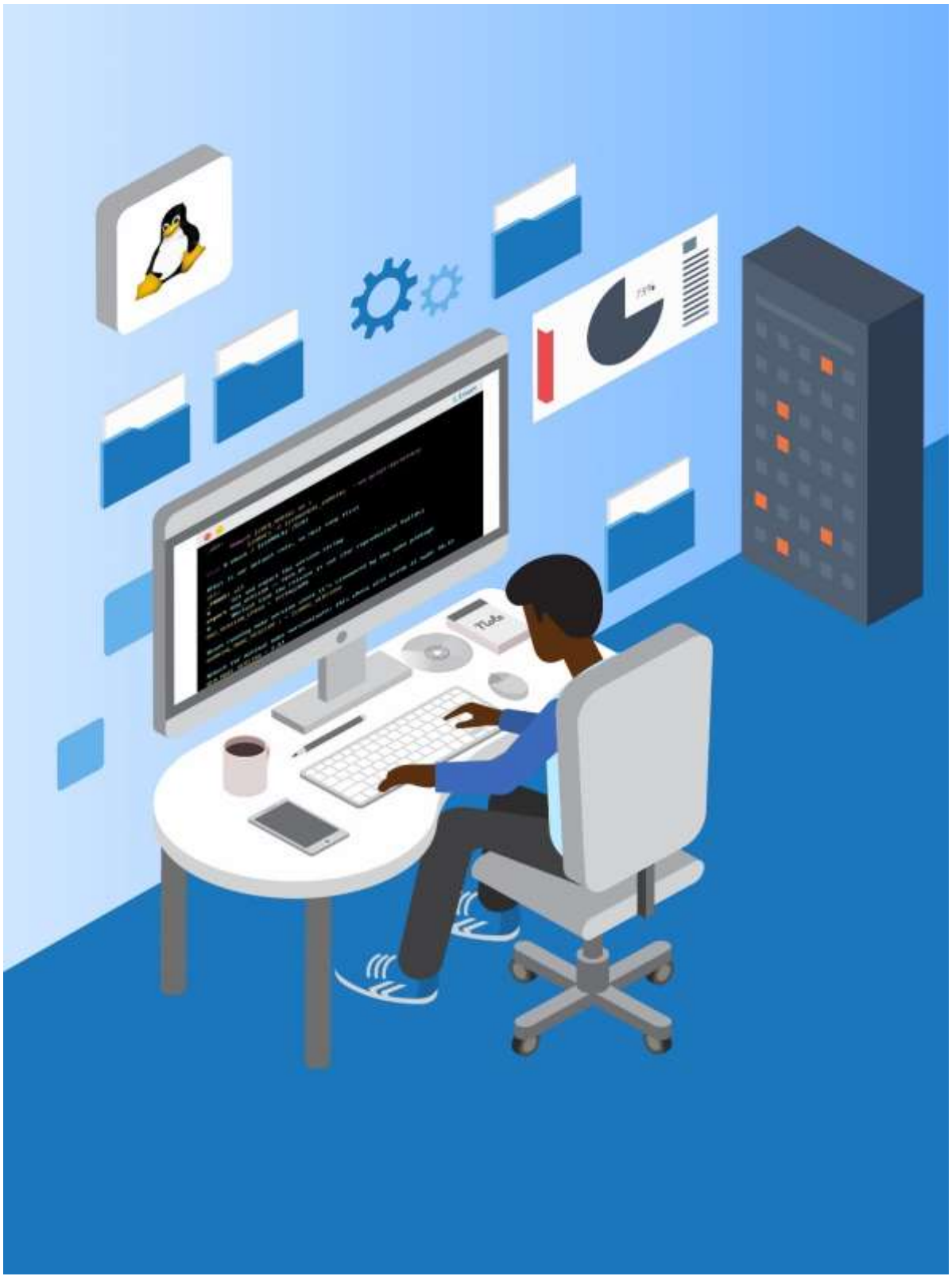
Smith, R. (2018). *Linux essentials* (2nd ed.). Wiley.

Ward, B. (2021). *How Linux works: What every superuser should know* (3rd ed.). No Starch Press.

Study.com. (n.d.). *Linux installation, hardware requirements & distribution*. Retrieved from https://study.com/academy/lesson/linux-installation-hardware-requirements-distribution.html

IntroServ. (n.d.). *Top 10 tools for Linux server administrators*. Retrieved from https://introserv.com/blog/top-10-tools-for-linux-server-administrators/

GoAbacus. (n.d.). *How to set up a small business server and network*. Retrieved from https://goabacus.com/how-to-set-up-a-small-business-server-and-network/



Indicative contents
2.1 Applying basic Linux commands **2.2 Creation of Linux files and directories** **2.3 Management of User and Groups** **2.4 Application of files and directories permissions**

Key Competencies for Learning Outcome 2 : Manage Files and Directories

Knowledge	Skills	Attitudes
• Description of Linux basic commands • Description of file system navigation • Description of basic commands for viewing current user information • Description of Current network information • Description of Linux files and directories • Description of user management	• Install package management • Apply Linux basic commands • Creating Linux files and directories • Creating user accounts in Linux server • Managing user and groups • Applying Files and directories permissions	• Demonstrate team spirit while working with others. • Being rapid during work execution • Take responsibility • Being organized to achieve the required result • Establish resilience in your daily activities • Being Attentive to details



Duration: 45 hrs

Learning outcome 2 objectives:



By the end of the learning outcome, the trainees will be able to:

1. Describe clearly Linux basic commands based on their purpose/function
2. Install correctly Linux package manager based on functionality
3. Apply correctly Linux basic commands based on their purpose/function
4. Create appropriately Linux files and directories based on their file types and directory Size
5. Create correctly user accounts in Linux server based on user roles and access requirements.
6. Manage user and groups correctly based on their roles and responsibilities
7. Applying Files and directories permissions based on user roles and access requirements

**Resources**

Equipment	Tools	Materials
● Computer ● UPS	● Ubuntu server ● VMware or VirtualBox ● Disk image utility software	● Internet

Indicative content 2.1: Applying Basic Linux Commands

Duration: 10 hrs

Theoretical Activity 2.1.1. Description of File system navigation

Tasks:

1: As a trainee you are asked to answer the following questions:

i. What is the purpose of the **`ls`** command in Linux?

ii. In most Linux systems, what does the command `l` refer to? What is its function?

iii. What information does the **`ls -la`** command provide, and how is it different from `ls` alone?

iv. What does the **`pwd`** command do, and what does the acronym stand for?

v. Describe the function of the **`cd`** command.

2: Present your findings to the whole class or one of your colleagues.

3: Read the Key readings 2.1.1. for more clarifications.

4: Ask questions where necessary.

Key readings 2.1.1: Description of File system navigation

- **ls (List Directory Contents):** Lists the files and directories in the current working directory.
  - ✓ **Common Options**:
    - **ls**: Basic listing of files and directories.
    - **ls -l**: Lists files with detailed information (permissions, owner, size, timestamp).
    - **ls -a**: Shows all files, including hidden files (those starting with.).
    - **ls -la**: Combines -l and -a to show detailed information for all files, including hidden ones.
  - ✓ **Examples:**
    - **ls:** Lists visible files and directories in the current directory.
    - **ls -la /home/user:** Shows detailed info for all files in the /home/user directory, including hidden ones.
- **l (Alias for ls -l)**: l is often an alias for ls -l in some Linux distributions or user-defined configurations. It provides detailed information about files and directories, such as permissions, ownership, size, and modification date.
- **pwd (Print Working Directory)**: Displays the full path of the current working directory.

- **cd (Change Directory)**: Changes the current working directory to the specified directory.
  - ✓ **Common Usages**:
    - **cd**: Changes to the user's home directory.
    - **cd /path/to/directory**: Moves to a specified directory.
    - **cd ..**: Moves up one directory level (parent directory)



Theoretical Activity 2.1.2: Description of basic commands for viewing current user information

Tasks:

1: As a trainee you are asked to answer the following questions:

i. What does the **id** command display in a Linux system, and what information can you obtain from its output?
ii. Explain the purpose of the **getent** command and give an example of how it can be used.
iii. What information does the **whoami** command provide, and how does it differ from the **who** command?
iv. What does the **who** command show, and what kind of information can you gather about users currently logged into the system?
v. Describe the function of the **w** command and the type of information it provides about logged-in users.

2: Present your findings to the whole class or one of your colleagues.

3: Read the Key readings 2.1.2 for more clarifications.

4: Ask questions where necessary.



Key readings 2.1.2: Description of basic commands for viewing current user information

id: The id command displays the user ID (UID) and group ID (GID) of the current user, along with supplementary group memberships.

1. **getent:** The getent command retrieves entries from system databases configured in /etc/nsswitch.conf. It can be used to obtain information about users, groups, and other system entities.
2. **Whoami:** The whoami command returns the username of the current user. It is a simplified way to find out who is logged in.

3. **Who:** The who command shows who is currently logged into the system. It provides information about the login time, terminal, and originating IP or hostname.
4. **w**: The w command provides a summary of the users currently logged in and what they are doing. It includes information about their activity, idle time, and source of login.



Theoretical Activity 2.1.3: Description of Processes management for Linux commands

Tasks:

1: Answer the following questions:

i. What information does the ps command provide, and how can it be used to view running processes?
ii. How does the top command differ from ps, and what real-time information does it display about system processes?
iii. What are the advantages of using htop over top, and what additional features does htop provide for managing processes?
iv. How can the pgrep command be utilized to find the process IDs of running processes based on specific criteria?
v. Describe the function of the kill command and how it is used to terminate processes in Linux. What are some common signals that can be sent with this command?

2: Present your findings to the whole class or one of your colleagues.

3: Read the Key readings 2.1.3 for more clarifications.

4: Ask questions where necessary.



Key readings 2.1.3: Description of Processes management for Linux commands

1. **Ps:** The ps (process status) command displays information about active processes. It provides a snapshot of current processes and their status.
   Common Options:
   ps aux: Shows detailed information about all processes for all users.
   ps -ef: Displays processes in a full-format listing.

2. **top:** The top command provides a real-time, dynamic view of system processes. It displays system summary information and a list of processes sorted by various criteria like CPU usage, memory usage, etc.
3. **htop**: htop is an interactive process viewer similar to top, but with a more user-friendly, ncurses-based interface. It allows for more intuitive management of processes and sorting options.
4. **pgrep:** The pgrep command searches for processes based on criteria like name or other attributes and returns their process IDs.
5. **kill:** The kill command sends signals to processes, typically to terminate them. The most commonly used signal is SIGTERM (default), which asks the process to terminate gracefully. SIGKILL can be used to forcefully kill a process.



Theoretical Activity 2.1.4: Description of Package management



Tasks:

1: asked to answer the following questions:

i. Define the term "package" in the context of a Linux environment.
ii. What commands would you typically use to uninstall software package in Linux server?
iii. What commands would you use to view a list of all installed packages?

2: Present your findings to the whole class or one of your colleagues.
3: Ask questions where necessary.
4: Read the Key readings 2.1.4 for more clarifications.



Key readings 2.1.4: Description of Package management

- **Installation**

Installation involves adding new software packages to your system using a package manager. Most package managers retrieve software from official repositories, which are collections of software packages maintained and verified for the operating system. The package manager ensures that all necessary dependencies (software components required by the package) are installed alongside the desired package.

✓ **Steps Involved:**
- The package manager resolves dependencies (other required software).
- Downloads the package and dependencies from the repository.
- Installs them on the system, setting up configuration files and system services as needed.

✓ **Types of Installations:**

- **Repository Installation**: Installing from official or third-party repositories.
- **Local Installation**: Installing packages from local files, like .deb (Debian) or .rpm (Red Hat) packages.

- **Uninstallation**

Uninstallation refers to removing a software package from the system. This process deletes the files associated with the package but might leave behind configuration files or logs. In some cases, a "purge" operation can be performed, which removes all associated files, including configuration data.

✓ **Steps Involved:**

- The package manager checks if the package is installed.
- Uninstalls the package, leaving dependencies if other programs still require them.
- Optionally, it can also remove dependencies and unused files with a command like autoremove.

- **Checking installed package**

This process involves querying the system to see what packages are currently installed. This is useful for verifying if certain software is present or for troubleshooting system configurations. You can also check package versions and get detailed information on each package.

✓ **Checking Package Versions:** If you need to check the version of an installed package: apt list nginx --installed

✓ **Detailed Package Information:** To view details such as the installation date, dependencies, and description of a package: apt show nginx



Theoretical Activity 2.1.5: Description of current network information



Tasks:

1: As a trainee you are asked to answer the following questions:

i. What is the purpose of the ifconfig command in Linux, and what type of information does it display about network interfaces?

ii. Explain the function of the ping command in network troubleshooting. How does it help diagnose connectivity issues?

iii. Describe what the traceroute command does. How can it be used to identify routing problems in a network?

iv. What information does the netstat command provide regarding network connections and interfaces, and why is it useful for system administrators?

v. What is the purpose of the nslookup command, and how can it be utilized to troubleshoot DNS-related issues?

2: Present your findings to the whole class or one of your colleagues.

3: Read the Key readings 2.1.1.e for more clarifications.

4: Ask questions where necessary.



Key readings 2.1.5: Description of current network information

- **Ifconfig** (Interface Configuration)**:** is a command-line tool used in Unix-based systems (like Linux) to configure network interfaces and view network settings such as IP address, subnet mask, and MAC address.
- **Ping (Packet Internet Groper):** ping is a utility used to test the reachability of a network host. It sends ICMP (Internet Control Message Protocol) echo requests to the target host and waits for replies. This helps verify whether the target host is reachable.
- **Traceroute**: traceroute is a network diagnostic tool that tracks the path packets take from your system to a destination, showing each router or hop along the way. It is useful for identifying where delays or issues occur on the network.
- **Netstat (Network Statistics)**: netstat is a command-line tool that displays network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. It is useful for monitoring network activity and troubleshooting.
- **Nslookup (Name Server Lookup)**: nslookup is a command-line tool used to query DNS (Domain Name System) to obtain the domain name or IP address mapping. It helps troubleshoot DNS-related issues.



Theoretical Activity 2.1.6 Description of System configuration



Tasks:

1: As a trainee you are asked to answer the following questions:

i. What information does the uname -a command provide about the system.
ii. Explain the purpose of the hostnamectl command and describe what information it can display or modify.
iii. What does the df -h command show, and how does the -h option affect its output?
iv. What information does the free -h command provide regarding system memory.
v. What does the sysctl -a command display, and how can it be used to adjust system parameters in Linux?

2: Present your findings to the whole class or one of your colleagues

3: Ask questions where necessary..

4: Read the Key readings 2.1.6 for more clarifications.



Key readings 2.1.6 : Description of System configuration

1. **uname -a**: uname -a is a command used to display detailed information about the system, including the kernel version, hardware, and operating system details such as **Kernel name** (e.g., Linux), **Network node hostname** (the system's name), **Kernel version** and release date, **Machine hardware architecture** (e.g., x86_64 for 64-bit) and **Operating system** (e.g., GNU/Linux).
2. **hostnamectl**: hostnamectl is a tool used in Linux to query or change the system's hostname and display related system information, such as the operating system, kernel version, architecture, and more.

   Usage:
   a. hostnamectl: Displays detailed information about the system's static hostname, pretty hostname, operating system, and architecture.
   b. hostnamectl set-hostname [new-hostname]: Changes the system's hostname.
3. **Common Fields**:
   a. Static hostname: The permanent hostname of the system.
   b. Pretty hostname: A more descriptive hostname (optional).
4. **df -h**: df -h is used to display the disk usage of file systems in a human-readable format, showing sizes in units like MB, GB, etc.
5. **free -h**: free -h is used to display information about the system's memory (RAM) and swap usage in a human-readable format.

   Output:
   a. **Total memory**: The total installed RAM.
   b. **Used memory**: RAM currently being used by processes.
   c. **Free memory**: RAM that is completely free.
   d. **Available memory**: Memory available for new applications, which includes free memory and cached memory.
   e. **Swap**: Information about the swap space, including total, used, and free swap.

   Usage:
   f. free -h: Displays memory usage in human-readable form (MB, GB, etc.).
6. **system_profiler (macOS-specific)**: system_profiler is a macOS command-line tool that provides detailed system information about hardware, software, and network configurations.
7. **sysctl -a**: sysctl -a is a command in Unix-like systems (primarily Linux) that displays all kernel parameters and their current values. It's mainly used for kernel tuning and system configuration.
   - ✓ **Usage**:
     - sysctl -a: Lists all system control parameters related to the kernel.

- sysctl [parameter_name]: Retrieves a specific kernel parameter (e.g., sysctl net.ipv4.ip_forward).
- sysctl -w [parameter_name]=[value]: Temporarily sets a kernel parameter.

✓ **Example**:

- sysctl -a | grep ipv4: Filters kernel parameters related to IPv4 networking.



Theoretical Activity 2.1.7: Description of task automation



Tasks:

1: As a trainee you are asked to answer the following questions:

i. What is the primary function of the at command in Linux, and in what scenarios would you typically use it?

ii. How does anacron differ from cron, and what advantages does it provide for running scheduled tasks on systems that may not be powered on continuously?

iii. Describe how cron works for scheduling repetitive tasks in Linux, and explain the significance of the crontab file in this process.

2: Present your findings to the whole class or one of your colleagues.

3: Read the Key readings 2.1.1.g for more clarifications.

4: Ask questions where necessary.



Key readings 2.1.7: Description of task automation

1. **at Command:** The at command is used for **one-time task scheduling**. It allows users to execute commands or scripts at a specific time in the future. Once the task is executed, it will not repeat, unlike cron or anacron.
2. **anacron Command:** Anacron is designed for scheduling **periodic tasks** on systems that are not guaranteed to be running all the time, such as laptops or desktops that may be turned off. Unlike cron, anacron will run missed jobs once the system is powered back on.
3. **cron Command:** Cron is one of the most widely used tools for scheduling **repetitive tasks**. It allows users to schedule commands to run at specified intervals such as minutes, hours, days, weeks, or months. Unlike anacron, cron assumes the system will always be running, and missed jobs are not executed later.



Practical Activity 2.1.8: Apply basic Linux commands



Task:

1: Individual, perform the following activity:

You are working as a Linux system administrator for a small company that hosts its website on a Linux-based web server. Recently, users have complained that the website has become slow and unresponsive. Your task is to diagnose the issue using basic Linux commands and perform some routine maintenance to ensure the server is running optimally.

2. Read the key reading 2.1.8
3: Perform the task provided above
4. Present your work to the trainer and your classmate
5: Perform the task provided in application of learning 2.1.



Key readings 2.1.8: Apply basic Linux commands

1. **System Information and Monitoring**
   - ✓ **uname -a**: Provides detailed system information, including the kernel version, architecture, and OS.
   - ✓ **hostnamectl**: Displays and allows changes to the hostname, and provides additional system information.
   - ✓ **Use these commands to verify system configurations and ensure you're working on the correct machine.**
2. **Disk and Memory Usage**
   - ✓ **df -h**: Checks disk usage for each mounted filesystem in a human-readable format. Helps you identify if the system is running out of disk space.
   - ✓ **free -h**: Shows memory (RAM) and swap usage. Use this to monitor available and used memory.
   - ✓ **Keep an eye on disk and memory usage to prevent performance issues caused by lack of resources.**
3. **Performance Monitoring**
   - ✓ **top**: Provides a real-time overview of system processes, CPU, and memory usage. Use it to monitor system performance and identify resource-intensive processes.
   - ✓ **htop**: (if installed) is a more user-friendly version of top with an intuitive interface.
   - ✓ Regularly monitor performance to ensure no process is monopolizing system resources.
4. **Network Connectivity**

- ✓ **ping**: Tests network connectivity by sending ICMP packets to a remote host. Useful for verifying if a network host or server is reachable.
- ✓ **traceroute**: Shows the path data takes to reach a destination, helpful in diagnosing network issues.
- ✓ Use network commands to troubleshoot connectivity and latency issues on the network.

5. Kernel Parameters and Tuning

- ✓ **sysctl -a**: Lists all system and kernel parameters, which are essential for system tuning and optimization.
- ✓ Make sure to understand the impact of modifying kernel parameters to optimize system performance without causing instability.

6. Managing File systems

- ✓ **lsblk**: Displays block devices (disks) and their mount points. Use it to identify available storage devices.
- ✓ **mount / umount**: Mount or unmount file systems when needed.
- ✓ Regularly check and manage file systems to prevent data loss or corruption.

7. File and Directory Management

- ✓ **ls, cp, mv, rm**: Use these commands to list files, copy/move files, and remove files/directories.
- ✓ **chmod, chown**: Change file permissions and ownership for better security.
- ✓ Handle files carefully, especially when deleting or changing permissions.

8. Command Usage Best Practices

- ✓ **Use man (manual pages)**: Always refer to the manual (e.g., man df) to understand command options and usage.
- ✓ **Command options**: Learn the key options (e.g., -h for human-readable output, -r for recursive operations) for better control over command execution.
- ✓ **Practice caution when using commands that modify the system (rm, chmod, chown, etc.)** to avoid accidental system changes.

9. Scripting and Automation

- ✓ **Basic shell scripting**: Automate repetitive tasks by creating simple shell scripts.
- ✓ Use scheduling tools like cron to run regular maintenance tasks like backups or system updates.

10. Regular Maintenance

- ✓ **Monitor system resources (CPU, memory, disk, and network)** regularly to prevent slowdowns.
- ✓ **Clean up unnecessary files** (e.g., old logs, temporary files) to free up disk space.
- ✓ **Ensure updates and patches are applied** to maintain security and stability.



Points to Remember

Description of file system navigation

- The ls command is used to list the contents of a directory.
- By default, it shows the names of files and directories in the current working directory.
- It can be modified with options (like -l, -a, etc.) to provide additional details about the files, such as file permissions, sizes, and hidden files.
- The command l is not a standard command in Linux by itself. However, it may be an alias for ls -l in many systems.
- If defined as an alias, it would typically list files in the long format, displaying detailed information such as permissions, owner, group, size, and last modification date.
- The ls -la command provides a detailed list of all files and directories, including hidden files (those beginning with a dot .).
- It displays information in a long format, including file permissions, number of links, owner, group, size, and timestamp of the last modification.
- In contrast, ls alone only shows the names of non-hidden files and directories in a simple list format.
- The pwd command stands for "print working directory."
- It displays the full path of the current directory in which the user is located.
- It helps users understand their location within the file system hierarchy.
- The cd command stands for "change directory."
- It is used to navigate between directories in the file system.
- Users can specify the path to the target directory, allowing them to move up, down, or to a different branch of the directory structure.
- Common usages include cd .. to move to the parent directory and cd ~ to return to the home directory.

Description of basic commands for viewing current user information

- The id command shows user and group identity information for the current user or a specified user.
- It displays the user ID (UID), the primary group ID (GID), and a list of supplementary group memberships
- Useful for understanding permissions and access control in a Linux environment.
- The getent command retrieves entries from administrative databases, such as users, groups, or hosts.
- It can access various database types configured in /etc/nsswitch.conf.
- Example: getent passwd displays user account information, similar to cat /etc/passwd.

- The whoami command displays the username of the current user executing the command.
- It tells you exactly who you are logged in as at that moment.
- In contrast, the who command shows a list of all users currently logged into the system, along with their terminal, login time, and originating IP address or hostname.
- The who command displays a list of users currently logged into the system.
- It provides details such as Username, Terminal name, Login time, and Source (IP address or hostname) from where the user is connected
- Useful for monitoring active users and system usage.
- The w command displays information about currently logged-in users and their activities.
- It provides Username, Terminal, Login time, Idle time, and the command currently being executed by each user
- It offers a more detailed view compared to the who command, allowing administrators to see what users are doing.

Description of processes management for Linux

- The ps command displays a snapshot of current processes running on the system.
- It provides information such as Process ID (PID), terminal associated with the process, CPU and memory usage, state of the process, and the command that started the process.
- Common options include ps aux for a detailed view of all processes or ps -ef for a full-format listing.
- The top command provides a dynamic, real-time view of processes running on the system, updating every few seconds.
- It displays CPU and memory usage, system load, running processes, and their statuses.
- Unlike ps, which gives a static snapshot, top allows users to monitor system performance and resource usage interactively.
- htop is an enhanced version of top, offering a more user-friendly interface with color-coded output.
- It allows users to scroll through the process list, search for processes, and sort by various columns interactively.
- Users can also send signals (like kill) to processes directly from the interface, making it easier to manage processes.
- The pgrep command searches for processes based on name and other attributes and returns their Process IDs (PIDs).

- It is useful for finding PIDs without listing all processes manually, using patterns or regular expressions.
- Common options include pgrep -u username to find processes owned by a specific user.
- The kill command is used to send signals to processes, typically to terminate them.
- It requires a PID as an argument, and the default signal sent is SIGTERM (terminate).
- Common signals include SIGKILL (kill immediately, cannot be ignored), SIGSTOP (pause the process), and SIGHUP (reload the process configuration).

Description of package management

- Know the name of the software you want to install.
- Use the appropriate package manager for your Linux distribution (e.g., apt for Debian/Ubuntu, yum for CentOS/RHEL, dnf for Fedora).
- It's often a good practice to update the package database before installation, typically done with commands like sudo apt update or sudo yum check-update.
- Use the installation command specific to the package manager (e.g., sudo apt install package_name or sudo yum install package_name).
- Check if the software is installed correctly by running it or checking its version.
- Determine the exact name of the software package to be uninstalled.
- Utilize the corresponding uninstallation command for your package manager (e.g., sudo apt remove package_name for apt, or sudo yum remove package_name for yum).
- Some package managers allow the option to remove configuration files as well (e.g., sudo apt purge package_name).
- Check if the package has been successfully uninstalled by verifying it with the package manager or checking the software's availability.
- Use commands like dpkg -l package_name for Debian-based systems or rpm -q package_name for RPM-based systems to check if a specific package is installed.
- Use dpkg -l for Debian-based systems or rpm -qa for RPM-based systems to list all installed packages.
- Many package managers also have GUI tools (like Synaptic for apt) that allow users to view installed packages visually.

Description of Current network information

- Ifconfig is a command-line tool used in Unix-based systems (like Linux) to configure network interfaces and view network settings such as IP address, subnet mask, and MAC address.

- Ping is a utility used to test the reachability of a network host. It sends ICMP (Internet Control Message Protocol) echo requests to the target host and waits for replies. This helps verify whether the target host is reachable.
- Traceroute is a network diagnostic tool that tracks the path packets take from your system to a destination, showing each router or hop along the way. It is useful for identifying where delays or issues occur on the network.
- Netstat is a command-line tool that displays network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. It is useful for monitoring network activity and troubleshooting.
- Nslookup is a command-line tool used to query DNS (Domain Name System) to obtain the domain name or IP address mapping. It helps troubleshoot DNS-related issues.

Description of System configuration

- The uname -a command displays comprehensive system information, including the kernel name, the hostname, the kernel version and release, the machine hardware name, the processor type, and the operating system.
- This information is useful for system administrators to quickly understand the system configuration and troubleshoot issues.
- The hostnamectl command is used to query and change the system hostname and related settings.
- It displays information such as Static hostname, transient hostname, icon name, chassis type, and operating system and kernel version.
- It allows administrators to change the hostname and manage other settings related to the system's identity.
- The df -h command displays disk space usage for all mounted file systems.
- The -h option makes the output human-readable, showing sizes in KB, MB, or GB instead of in bytes.
- It provides essential information for monitoring disk usage and managing filesystem space.
- The free -h command shows memory usage statistics, including total, used, free, shared, and cached memory.
- The -h option formats the output in a human-readable way, making it easier to interpret (e.g., using MB or GB).
- It is important for monitoring system memory usage and performance.
- The system_profiler command is specific to macOS and provides detailed information about the hardware and software of the system.
- It gathers data on hardware specifications (CPU, memory, storage), software versions (macOS version, installed applications), and setwork configurations.

- It is useful for diagnostics, inventory management, and system auditing.
- The sysctl -a command displays all kernel parameters and their current values.
- It is used to configure kernel parameters at runtime, affecting system behavior and performance.
- This command helps system administrators understand and tune system settings for better performance and security.

Description of task automation

- The at command is used for one-time task scheduling. It allows users to execute commands or scripts at a specific time in the future. Once the task is executed, it will not repeat, unlike cron or anacron.
- Anacron is designed for scheduling periodic tasks on systems that are not guaranteed to be running all the time, such as laptops or desktops that may be turned off. Unlike cron, anacron will run missed jobs once the system is powered back on.
- Cron is one of the most widely used tools for scheduling repetitive tasks. It allows users to schedule commands to run at specified intervals such as minutes, hours, days, weeks, or months. Unlike anacron, cron assumes the system will always be running, and missed jobs are not executed later.

You can Apply basic Linux commands to:

- Provides detailed system information, including the kernel version, architecture, and OS.
- Displays and allows changes to the hostname, and provides additional system information.
- Use these commands to verify system configurations and ensure you're working on the correct machine.
- Checks disk usage for each mounted file system in a human-readable format. Helps you identify if the system is running out of disk space.
- Shows memory (RAM) and swap usage. Use this to monitor available and used memory.
- Provides a real-time overview of system processes, CPU, and memory usage. Use it to monitor system performance and identify resource-intensive processes.
- Tests network connectivity by sending ICMP packets to a remote host. Useful for verifying if a network host or server is reachable.
- Shows the path data takes to reach a destination, helpful in diagnosing network issues.
- Use network commands to troubleshoot connectivity and latency issues on the network.

- Lists all system and kernel parameters, which are essential for system tuning and optimization.
- Displays block devices (disks) and their mount points. Use it to identify available storage devices.
- Mount or unmount file systems when needed.
- Regularly check and manage file systems to prevent data loss or corruption.
- Use these commands to list files, copy/move files, and remove files/directories.
- Change file permissions and ownership for better security.
- Handle files carefully, especially when deleting or changing permissions.
- Always refer to the manual (e.g., man df) to understand command options and usage.
- Learn the key options (e.g., -h for human-readable output, -r for recursive operations) for better control over command execution.
- Practice caution when using commands that modify the system (rm, chmod, chown, etc.) to avoid accidental system changes.
- Automate repetitive tasks by creating simple shell scripts.
- Use scheduling tools like cron to run regular maintenance tasks like backups or system updates.
- Monitor system resources (CPU, memory, disk, and network) regularly to prevent slowdowns.
- Clean up unnecessary files (e.g., old logs, temporary files) to free up disk space.
- Ensure updates and patches are applied to maintain security and stability.



Application of learning 2.1.

You have been assigned the responsibility of managing a web server for a small company. This server hosts the company's website and requires regular maintenance, monitoring, and troubleshooting to ensure optimal performance and minimize downtime.

Your role includes applying updates, securing the server, monitoring traffic, resolving technical issues, and ensuring the website remains accessible and functional at all times.

Indicative content 2.2: Creation of Linux files and Directories

Duration: 10 hrs

Theoretical Activity 2.2.1: Description of Linux files and directories

Tasks:

1: Answer the following questions

i. Describe the Linux file system hierarchy, including the significance of the root directory and common directories such as /home, /etc, and /var.

ii. What are the different categories or types of file systems used in Linux, and how do they differ in terms of functionality and performance?

iii. Explain the process of creating and managing directories and files in Linux, including commands for creating, deleting, and modifying files and directories.

2: Present the findings to trainer, your colleagues or to the whole class

3: Ask questions where necessary.

4: Read the key readings 2.2.1.

Key readings 2.2.1:Description of Linux files and directories

1. Directory Structure

- ✓ **Root Directory (/)**: The top-level directory of the Linux file system hierarchy. All other directories are subdirectories of /.
  - **Example**: /home, /etc, /var.
- ✓ **Home Directory (/home)**: Contains personal directories for each user. For example, /home/john would be the home directory for user john.
- ✓ **Root User's Home Directory (/root)**: The home directory for the root user (superuser). It is separate from /home for security reasons.
- ✓ **System Directories**
  - **/bin**: Contains essential user binaries (commands) that are needed for basic system operation and repair. **Examples**: ls, cp, mv.
  - **/sbin**: Contains system binaries used for system administration. These are typically used by the root user. **Examples**: shutdown, ifconfig.
  - **/etc**: Contains system-wide configuration files and shell scripts. **Examples**: /etc/passwd, /etc/fstab.

- **/var**: Contains variable data like logs, mail spools, and temporary files that are expected to grow in size. **Examples**: /var/log, /var/spool.
- **/tmp**: Holds temporary files that are created by various programs. These files are usually deleted upon reboot or after a certain period.
- **/dev**: Contains device files that represent hardware components and drivers. These are not regular files but rather interfaces to hardware. **Examples**: /dev/sda (a disk), /dev/tty (a terminal).
- **/proc**: A virtual file system that provides information about system processes and hardware. **Examples**: /proc/cpuinfo (CPU information), /proc/meminfo (memory usage).
- **/sys**: A virtual file system that exposes kernel and device information.
    - **Usage**: Provides a way to interact with the kernel and devices, often used by system tools and utilities.
- **/usr**: Contains user utilities and applications. It is the primary directory for user software and libraries. **Examples**: /usr/bin, /usr/lib.
- **/opt**: Used for installing third-party applications and software packages.
- **/mnt and /media**: Used for mounting file systems, such as external drives or network file systems.

2. File Permissions and Ownership

✓ **File Permissions**

- **Read (r)**: Allows viewing the contents of the file.
- **Write (w)**: Allows modifying the file.
- **Execute (x)**: Allows executing the file if it is a script or program.

✓ **Permission Notation**

- **Symbolic Notation**: rwxr-xr,
- **Explanation**: rwx (owner), r-x (group), r-- (others).
- **Octal Notation: 755**
- **Explanation**: 7 (owner: read, write, execute), 5 (group: read, execute), 5 (others: read, execute).

✓ **Changing Permissions**

- **chmod Command**: Used to change file permissions.
- **Example**: chmod 755 file.txt sets the permissions to rwxr-xr-x.

✓ **Changing Ownership**

- **chown Command**: Used to change the owner and group of a file.
- **Example**: chown user: group file.txt changes the owner to user and the group to group.

3. Special Files and Links

✓ **Symbolic Links**: A symbolic link is a file that points to another file or directory.

- **Creation**: ln -s target linkname

- **Example**: ln -s /etc/hosts myhosts creates a symlink myhosts pointing to /etc/hosts.

✓ **Hard Links**: A hard link is another name for an existing file on the file system. It refers to the same inode.

- **Creation**: ln target linkname
- **Example**: ln file1 file2 creates a hard link file2 to file1.

4. Managing Files and Directories

✓ **Creating and Deleting Directories**

- **Create Directory**: mkdir directory_name
- **Remove Directory**: rmdir directory_name (empty directory) or rm -r directory_name (directory with contents)

✓ **Creating and Deleting Files**

- **Create File**: touch file_name
- **Delete File**: rm file_name

✓ **Listing Files**

- **List Files**: ls (lists files in the current directory)
- **Options**: ls -l (detailed listing), ls -a (includes hidden files)

5. Searching and Finding Files

✓ **Finding Files**

- **find Command**: Searches for files and directories in a directory hierarchy.
- **Example**: find /home -name "*.txt" searches for .txt files in /home.

✓ **Locating Files**

- **locate Command**: Finds files quickly using a database.
- **Example**: locate filename finds files matching filename.



Practical Activity 2.2.2: Creation of Linux files and directories



Task:

1: Individual, perform the following activity:

You are setting up a new project for your team. You need to create a structured directory layout for organizing project files, including documentation, source code, configuration files, and logs. You will also create some initial files and set appropriate permissions.

a) Create a main directory for the project named NewProject.
b) Within the NewProject directory, create subdirectories for docs, src, config, and logs.

c) In the docs directory, create a README file. In the config directory, create an initial configuration file named app.conf. In the logs directory, create a placeholder log file named app.log.

2: Read the key reading 2.2.2.

3: create a structured directory layout

4: Present your work to the trainer and your classmate

5: Perform the task provided in application of learning 2.2.



Key readings 2.2.2: Creation of Linux files and directories

1. Creating Files

There are several ways to create files in Linux. Below are some of the most common methods:

- **Using the touch command**: This command is used to create an empty file.
  This creates an empty file named filename.txt in the current directory.
- **Using the cat command**: You can create a file and add content at the same time using the cat command.
  After running this command, you can type the content of the file and then press CTRL+D to save and exit.
- **Using a text editor**: You can also create and edit files using a text editor like nano
  After editing, save the file with CTRL+O (in nano) and exit with CTRL+X.

2. Creating Directories

The mkdir command is used to create directories (folders).

- **Create a single directory**:
- **Create a directory and its parent directories**: If you want to create a directory inside another directory that doesn't exist yet, use the -p option.
  This command will create both parent_directory and child_directory in one go.

3. File and Directory Permissions

Linux uses a permission model that controls who can read, write, or execute files and directories.

- **Check file or directory permissions**: Use the ls -l command to view the permissions.
- **Change permissions**: Use the chmod command to modify the permissions.
  In this example, 755 sets the file to be readable and executable by everyone, but only writable by the owner.

4. File and Directory Ownership

- **Check ownership**: Use ls -l to see the owner and group of a file or directory.
- **Change ownership**: The chown command is used to change ownership of a file or **5. File and Directory Operations**

- **Copying files**: Use the cp command to copy files.
- **Moving/Renaming files**: Use the mv command to move or rename files and directories.
- **Deleting files and directories**:
    - To delete a file: rm filename.txt
    - To delete a directory: rm -r directory_name

6. Managing Hidden Files and Directories

In Linux, files and directories that start with a dot (.) are hidden. You can view them using:

ls -a

To create a hidden file or directory, just prefix the name with a dot:

touch .hidden_file

mkdir .hidden_directory

7. Symbolic and Hard Links

- **Symbolic link (symlink)**: A symbolic link points to a file or directory, like a shortcut.
  ln -s /path/to/file symlink_name
- **Hard link**: This links directly to the inode of the file.
  ln /path/to/file hardlink_name

8. File Compression

You can compress files using commands like gzip, bzip2, or zip.

- Compress a file: gzip filename.txt
- Decompress a file: gunzip filename.txt.gz



Points to Remember

Description of Linux files and directories

- The top level of the file system hierarchy; all other directories branch from this point.
- **/home:** Contains user-specific directories where personal files and settings are stored.
- **/etc:** Stores system configuration files that control the operation of the system.
- **/var:** Contains variable data files, such as logs, mail spools, and temporary files.
- **/usr:** Contains user-related programs and data, including application binaries and libraries.
- **/bin:** Holds essential command binaries available in single-user mode and required for system boot.
- **/lib:** Contains essential shared libraries and kernel modules.

- File Hierarchy Standard (FHS) is a set of guidelines that defines the directory structure and directory contents in Unix-like operating systems.
- Ext (Extended File System) such as ext2 for the original extended file system, no journaling, ext3 with journaling capabilities for improved reliability, and ext4 to improved version of ext3, offering larger file sizes and faster performance.
- ReiserFS is a journaling file system known for its performance with small files.
- XFS is a high-performance journaling file system designed for scalability, particularly for large files.
- Btrfs (B-tree File System) is a modern file system that supports snapshots, pooling, and built-in RAID functionality.
- FAT32 and NTFS are commonly used for compatibility with Windows systems; often used on removable media.
- Network File Systems (NFS) allows file access over a network, enabling multiple machines to share files.
- Use the mkdir command (e.g., mkdir my_directory), and se commands like touch to create an empty file (e.g., touch myfile.txt) or echo to create a file with content (e.g., echo "Hello World" > myfile.txt).
- Use ls to view files in a directory, rm to remove files (e.g., rm myfile.txt) and rmdir to remove empty directories (e.g., rmdir my_directory), mv to move or rename files and directories (e.g., mv oldname.txt newname.txt), cp to copy files (e.g., cp source.txt destination.txt), and use commands like cat, less, or more to display file content (e.g., cat myfile.txt).

Creating Linux files and directories

- Create a root directory for the project to keep all files organized.
- Create subdirectories for key components such as docs, src, config, and logs for better organization.
- Create initial configuration file and create a placeholder log file to capture application logs.
- Assign appropriate permissions to files and directories based on who should access them (owner, group, others).
- Changing Ownership to ensure that the correct user and group own the directories and files for security purposes.
- Verify Directory and File Structure creation using ls or tree (if installed).
- Modifying and Editing Files to use text editors like nano, vim, or vi to edit files, such as configuration or README files.
- Secure Sensitive Files like configuration or credentials, ensure restrictive permissions (600 for owner-only access) and regularly check file permissions using ls -l to avoid accidental exposure.

- Maintain the structure as the project grows by organizing new files into the relevant subdirectories (e.g., additional docs in docs, scripts in src).
- Ensure that critical files like configuration files and logs are backed up regularly, especially if logs are essential for audit purposes.



Application of learning 2.2.

You are tasked with setting up a new web development project for your team on a Linux server. The project involves organizing files for the website's source code, configuration files for the server, documentation, and logs for troubleshooting. You need to ensure the correct file structure is created, necessary files are initialized, and permissions are set appropriately for team collaboration.

Indicative content 2.3: Management of User and Groups

Duration: 10 hrs

Theoretical Activity 2.3.1: Description of user management

Tasks:

1: Answer the following questions:

i. What is a login name in Linux, and how is it used to identify a user in the system?

ii. What is an encrypted password in Linux, and why is it important for system security?

iii. What are UID (User ID) and GID (Group ID) in Linux, and how do they function in managing user permissions and access?

iv. What is a user's default shell in Linux, and how does it affect the user experience and command execution?

2: Present the findings to trainer, your colleagues or to the whole class

3: Ask questions where necessary.

4: Read the key readings 2.3.1.a

Key readings 2.3.1: Description of user management

1. **User Attributes:** When creating a user in Linux, several attributes are associated with the user account
2. **Login Name**: This is the unique identifier used to log in to the system. It is a string of characters (e.g., john_doe).
3. **Encrypted Password**: The password associated with a user is encrypted and stored in /etc/shadow to ensure security. The password is never stored in plain text.
4. **UID (User ID)**: Each user is assigned a unique numeric identifier known as the **User ID** (UID), which is used by the system to identify the user.
5. **Default Range**: User UIDs usually start at 1000 for regular users (0 for root).
6. **Command to Specify UID**: useradd -u 1001 john_doe
7. **GID (Group ID)**: Each user belongs to a group, and the **Group ID** (GID) represents the primary group of the user.
8. **Default Behavior**: A user typically has their own primary group with the same name as the login name.
9. **Command to Specify GID**: useradd -g 1001 john_doe

10. **User's Default Shell**: The shell that the user interacts with after logging in (e.g., bash, sh, zsh).
11. **Command to Specify Shell**: useradd -s /bin/bash john_doe



Practical Activity 2.3.2: Creating user attributes



Task:

1: Individual, perform the following activity:
You are asked to go to computer lab to create a new user account for a developer on a Linux server. Additionally, you need to ensure proper user attributes are set to align with company policies.
2: Read the key reading 2.3.2
3: create account for developer
4: Present your work to the trainer and your classmate
3: Perform the task provided in application of learning 2.3



Key readings 2.3.2:Creating user attributes

1. Creating a New User

The primary command for creating new users in Linux is useradd. You can also use adduser (which is a more user-friendly frontend to useradd in some distributions).

Basic Syntax:

sudo useradd [options] username

2. Key Attributes to Define for a User

When creating a user, several attributes can be set:

2.1 Username

The username is the identifier for the user in the system.

- Example: Create a new user john.
  sudo useradd john

2.2 User ID (UID)

The UID is a unique numerical identifier for the user. By default, the system assigns the next available UID.

- To assign a specific UID: sudo useradd -u 1050 john

2.3 Primary Group

When a user is created, a primary group is also created by default. You can also assign the user to an existing group.

- To specify a primary group: sudo useradd -g groupname john

2.4 Secondary Groups

A user can belong to multiple groups, which help manage access to files and resources.

- To add the user to secondary groups: sudo useradd -G group1,group2 john

2.5 Home Directory

By default, the user's home directory is created in /home/username. However, you can specify a custom directory.

- To specify a home directory: sudo useradd -d /path/to/home/directory john
- To ensure the home directory is created automatically: sudo useradd -m john

2.6 Shell

Each user is assigned a default shell, usually /bin/bash or /bin/sh. You can specify a different shell for the user.

- To assign a shell: sudo useradd -s /bin/bash john

2.7 Password

After creating a user, you need to set a password for the account. The passwd command is used for this.

- Set a password for the user: sudo passwd john

2.8 User Expiry Date

You can specify an expiry date for the user account. After the expiry date, the user will no longer be able to log in.

- To set an expiry date (in YYYY-MM-DD format): sudo useradd -e 2024-12-31 john

2.9 Account Inactivity

The inactivity period is the number of days after a password expires that the account will be disabled.

- Set inactivity period (in days): sudo useradd -f 30 john

3. Modifying User Attributes

If you need to modify an existing user's attributes, you can use the usermod command.

Syntax: sudo usermod [options] username

Example Modifications:

- Change the user's home directory: sudo usermod -d /new/home john
- Add the user to a new secondary group: sudo usermod -aG newgroup john
- Change the user's shell: sudo usermod -s /bin/zsh john

4. Deleting a User

To delete a user, the userdel command is used.

- To remove a user without deleting their home directory: sudo userdel john
- To remove the user and their home directory: sudo userdel -r john

5. Viewing User Attributes

- **Check User Details**: You can view user attributes by inspecting the /etc/passwd file.

  cat /etc/passwd

Each user has an entry that includes their username, UID, GID (group ID), home directory, and shell.

- **Check Group Membership**: You can check which groups a user belongs to using the groups command: groups john
- **Check Expiry Date**: You can view the account's expiry date using the chage command: sudo chage -l john

6. User Configuration Files

Some key files that define user attributes and settings include:

- **/etc/passwd**: Contains user account information like username, UID, GID, home directory, and shell.
- **/etc/shadow**: Contains encrypted password information and details about account expiration.
- **/etc/group**: Defines the groups and lists the members of each group.

7. Additional Attributes and Settings

You can also configure advanced user attributes related to system resource limits, such as CPU usage, memory limits, etc., using configuration files like /etc/security/limits.conf.



Points to Remember

While creating a new user account in a Linux server cchoose a unique user ID and descriptive username for easy identification

- Ensure the UID is unique and does not conflict with existing UIDs on the system.
- Set the default shell based on the developer's needs (e.g., /bin/bash, /bin/zsh).
- Specify a dedicated home directory for the user to store their files, if needed.
- Assign a secure password for the new user. Ensure it complies with your organization's password policies.
- Use passwd to set or change the user's password.
- Assign the user to a primary group, often a group with the same name as the user or a specific group for development.
- Add the user to additional groups as needed (e.g., developers, admin).
- Ensure the user has the correct permissions for their files and directories.
- Manage group ownership and permissions for shared resources.
- Set the user’s shell to one that is compatible with their development environment.
- Set an account expiration date if the user’s access needs to be limited to a specific period.
- Check the user account details using commands like id, getent passwd, and groups.

- Verify that the home directory is created and has the correct ownership and permissions.
- Understand how to lock and unlock user accounts as needed.
- Keep track of user activities and changes to the account for security auditing.
- Document the user's account details, including username, UID, GID, and groups.
- Inform the developer of their new account details and any relevant instructions for accessing the server.



Theoretical Activity 2.3.3: Description of group management



Tasks:

1: Answer the following questions:

i. What is a group name in Linux, and how is it used in relation to user management?
ii. What is a GID (Group ID) in Linux, and why is it important for managing permissions and group associations?
iii. How can you list the members of a specific group in Linux, and what command would you use to do so?
iv. What is the purpose of a group password in Linux, and how does it differ from individual user passwords?
v. How does group ownership affect file permissions in Linux, and what command can you use to change the group ownership of a file?
vi. Explain how group permissions work in Linux and how they can be set for files and directories.
vii. What is a group expiration date in Linux, and how can it be set or modified for a specific group?

2: Present the findings to trainer, your colleagues or to the whole class

3: Ask questions where necessary.

4: Read the key readings 2.3.2.a



Key readings 2.3.3:Description of group management

1. **Group Attributes:** Groups are collections of users that can share common permissions. Groups are defined by various attributes:
   - ✓ **Group Name**: A unique name assigned to identify a group (e.g., developers).
   - ✓ **GID (Group ID)**: The **Group ID** is a numeric identifier assigned to the group.
2. **Group Members**: Users can be members of one or more groups. Each user's primary group is specified, and they can belong to additional secondary groups.
   - ✓ **Command to Add Users to a Group**:

- Primary Group: useradd -g developers john_doe
- Secondary Group: usermod -aG dev_team john_doe

3. **Group Password**: Some groups can have a password that allows users to temporarily join the group using the newgrp command.
4. **Group Ownership**: Files and directories can be assigned ownership by a group. Any member of the group has access to files based on the group's permissions.
5. **Group Permissions**: Files and directories can have permissions assigned to the group that owns them (e.g., read, write, execute).
   - ✓ **Command to Set Group Permissions**:
     - Grant read/write to group: chmod g+rw file.txt
     - Remove execute for group: chmod g-x script.sh
6. **Group Expiration Date**: Groups can have an expiration date that determines when the group will be disabled.
   - ✓ **Command to Set Group Expiration Date**: gpasswd -e YYYY-MM-DD developers



Practical Activity 2.3.4: Creating group attributes



Task:

1: Individual, perform the following activity:

You are tasked with you school to create a new group for a team of developers on a Linux server. This group will have access to shared resources like source code directories and project files. You will also need to define the necessary group attributes, such as group name, GID, members, and permissions to ensure proper collaboration and security.

2: Read the key reading 2.3.4

3: create new group for a team of developers

4: Present your work to the trainer and your classmate

5: Perform the task provided in application of learning 2.3



Key readings 2.3.4: Creating group attributes

1. **Group Name**: Choose a meaningful and descriptive name for the group that reflects its purpose (e.g., dev_team, admins).
2. **GID (Group ID)**: Ensure that the Group ID is unique and does not conflict with other groups. You can either assign a specific GID or let the system automatically allocate one. Use the groupadd -g GID groupname command if you need to

specify a GID.

3. **Add Members to the Group**
   - ✓ **Primary Group**: Each user has one primary group. When a user creates files, the group ownership defaults to their primary group.
   - ✓ **Secondary Groups**: Users can belong to multiple secondary groups. Add users to the group with the usermod -aG groupname username command.
     - Make sure that adding a user to a group doesn't remove them from existing groups by using the -aG option.
4. **Assign Group Ownership**: Ensure that directories or files that the group should manage have their group ownership set correctly.
5. **File Permissions**: Define read, write, and execute permissions for the group. Group permissions control what actions users in the group can perform on files and directories.
6. **Setgid on Directories**: If a directory is used by multiple users, setting the setgid permission ensures that new files and subdirectories created within inherit the group ownership.
7. **Group Password**: In rare cases, a group password can be set using the gpasswd groupname command. This allows users to switch to the group using the newgrp command and the group password.
8. **List Groups**: Use the getent group groupname or grep groupname /etc/group command to verify that the group has been created with the correct attributes.
9. **Check Group Members**: Use the groups username or getent group groupname command to see which users belong to a specific group.
10. **Control Resource Access**: Groups are essential for managing access to system resources, shared files, and directories. Assign groups based on roles or projects to simplify permission management.
11. **Shared Project Directories**: Ensure that all users in the group have the correct level of access to shared project directories (e.g., read, write, execute).
12. **Document Group Creation**: Keep records of the group attributes such as group name, GID, members, and any special permissions for future reference and auditing purposes.



Points to Remember

- While creating group attributes choose a meaningful and descriptive name for the group that reflects its purpose (e.g., dev_team, admins).
- Ensure that the Group ID is unique and does not conflict with other groups. You can either assign a specific GID or let the system automatically allocate one. Use the groupadd -g GID groupname command if you need to specify a GID.

- Each user has one primary group. When a user creates files, the group ownership defaults to their primary group.
- Users can belong to multiple secondary groups. Add users to the group with the usermod -aG groupname username command.
- Make sure that adding a user to a group doesn't remove them from existing groups by using the -aG option.
- Ensure that directories or files that the group should manage have their group ownership set correctly.
- Define read, write, and execute permissions for the group. Group permissions control what actions users in the group can perform on files and directories.
- If a directory is used by multiple users, setting the setgid permission ensures that new files and subdirectories created within inherit the group ownership.
- In rare cases, a group password can be set using the gpasswd groupname command. This allows users to switch to the group using the newgrp command and the group password.
- Use the getent group groupname or grep groupname /etc/group command to verify that the group has been created with the correct attributes.
- Use the groups username or getent group groupname command to see which users belong to a specific group.
- Groups are essential for managing access to system resources, shared files, and directories. Assign groups based on roles or projects to simplify permission management.
- Ensure that all users in the group have the correct level of access to shared project directories (e.g., read, write, execute).
- Keep records of the group attributes such as group name, GID, members, and any special permissions for future reference and auditing purposes.



Theoretical Activity 2.3.5: Description of user and group management



Tasks:

1: Answer the following questions:

i. What is the purpose of user and group management in Linux?
ii. How does Linux distinguish between users and groups in terms of permissions and access control?
iii. What is the difference between a primary group and a secondary group for a user?
iv. Where are user and group information stored in Linux?

v. What is the purpose of the /etc/passwd and /etc/shadow files in user management?

vi. How do you delete a user from the system, and what happens to their files?

2: Present the findings to trainer, your colleagues or to the whole class

3: Ask questions where necessary.

4: Read the key readings 2.3.3.a



Key readings 2.3.5:Description of user and group management

1.. Managing Users and Groups: Managing users and groups is essential for maintaining system security, ensuring proper access, and organizing users.

- ✓ **Creating Users**
  - **Command**: useradd
  - Creates a user with UID 1002, adds them to the developers group, and sets their default shell to bash.
- ✓ **Modifying Users**
  - **Command**: usermod
  - Modify existing user attributes, such as adding them to a secondary group.
- ✓ **Deleting Users**
  - **Command**: userdel
- ✓ **Creating Groups**
  - **Command**: groupadd
- ✓ **Modifying Groups**
  - **Command**: groupmod
- ✓ **Deleting Groups**
  - **Command**: groupdel
- ✓ **Assigning Permissions**
  - **Command**: chmod

2. Manage file and directory permissions for users and groups.

- ✓ **Viewing User and Group Information**
  - **Command**: id john_doe (Shows UID, GID, and groups for the user).
  - **Command**: groups john_doe (Lists all groups a user belongs to).



Practical Activity 2.3.6: Manage users and groups



Task:

1: Individual, perform the following activity:

You are an IT consultant responsible for managing users and groups in a small legal office with 10 employees. The office has three departments: Legal (4 users), Admin (3 users), and IT (2 users). Each department should have access to its own files, but not to files of other departments. Additionally, all users should have access to a shared directory where they can collaborate on non-sensitive documents. You are tasked with creating user accounts, assigning them to appropriate groups, and managing file permissions.

2: Read the key reading 2.3.6

3: creating user accounts and managing file permissions.

4: Present your work to the trainer and your classmate

5: Perform the task provided in application of learning 2.3



Key readings 2.3.6: Manage users and groups

1. **Creating User Accounts:**
   - Use useradd to create new users and specify important attributes like home directory (-m), default shell (-s), and user password (passwd).
   - Ensure that each user is given the appropriate login environment and permissions.
2. **Creating Groups:**
   - Use groupadd to create groups based on departmental or role-based needs.
   - Assign users to groups using usermod -aG to manage access controls more efficiently.
3. **Assigning Users to Groups:** Ensure that each user is added to the correct group for their department, ensuring access to shared resources and limiting unauthorized access to sensitive data.
4. **Setting File and Directory Permissions:**
   - Use chmod and chown to control access to files and directories.
   - Set appropriate permissions based on department requirements:
   - 770 for directories that need to be private within a department.
   - 777 for directories that should be accessible to all users (like shared collaboration spaces).
5. **Managing Shared Resources:**
   - Create shared directories with open permissions (chmod 777) for collaboration.
   - Ensure sensitive files are stored in department-specific directories with restricted access.
6. **Ongoing User Management:**
   - Regularly update group memberships as users join or leave the

organization.

- Change user passwords periodically for enhanced security.
- Use gpasswd -d to remove users from specific groups when needed.

7. **Verifying Access Control:**
    - Test permissions for users to ensure they can only access authorized directories.
    - Use commands like cd to verify that users are blocked from unauthorized areas and have proper access to shared resources.



Points to Remember

Description of user management

- Login Name is unique identifier assigned to each user in the Linux system, often referred to as the username.
- Identification used for user authentication and to control access to resources.
- Format consists of alphanumeric characters, underscores, and cannot exceed a certain length (usually 32 characters).
- File Storage for Login names are stored in the /etc/passwd file alongside user information.
- **Encrypted Password** are stored in an encrypted format rather than as plain text to enhance security.
- Uses cryptographic hash functions (e.g., SHA-512) to convert passwords into a fixed-length string, making it difficult for attackers to retrieve the original password.
- Encrypted passwords are usually stored in the /etc/shadow file, which is only accessible by the root user.
- Protects user accounts from unauthorized access, ensuring that even if the password file is compromised, the actual passwords remain secure.
- **UID (User ID) is a** unique numerical identifier assigned to each user in the system. It helps the operating system track user permissions and ownership of files.
- **GID (Group ID) is** unique numerical identifier assigned to each user group. It determines the access rights for all users within that group.
- Both UIDs and GIDs are used to enforce file permissions, control access to resources, and facilitate multi-user environments.
- The root user typically has a UID of 0, while regular users start from UID 1000.
- The command-line interface that is loaded for a user when they log in. Common shells include /bin/bash, /bin/sh, and /bin/zsh.

- The default shell determines the command syntax, available features (like scripting capabilities), and the environment (e.g., prompts, aliases).
- Each shell has associated configuration files (e.g., .bashrc, .bash_profile) that allow users to customize their environment.
- Users can change their default shell using the chsh command or by modifying the /etc/passwd file.

While creating a new user account in a Linux server choose a unique user ID and descriptive username for easy identification.

- Ensure the UID is unique and does not conflict with existing UIDs on the system.
- Set the default shell based on the developer's needs (e.g., /bin/bash, /bin/zsh).
- Specify a dedicated home directory for the user to store their files, if needed.
- Assign a secure password for the new user. Ensure it complies with your organization's password policies.
- Use passwd to set or change the user's password.
- Assign the user to a primary group, often a group with the same name as the user or a specific group for development.
- Add the user to additional groups as needed (e.g., developers, admin).
- Ensure the user has the correct permissions for their files and directories.
- Manage group ownership and permissions for shared resources.
- Set the user's shell to one that is compatible with their development environment.
- Set an account expiration date if the user's access needs to be limited to a specific period.
- Check the user account details using commands like id, getent passwd, and groups.
- Verify that the home directory is created and has the correct ownership and permissions.
- Understand how to lock and unlock user accounts as needed.
- Keep track of user activities and changes to the account for security auditing.
- Document the user's account details, including username, UID, GID, and groups.
- Inform the developer of their new account details and any relevant instructions for accessing the server.

Description of group management

- A group name is an identifier for a collection of users.
- It is used to manage permissions and access controls for files and resources.
- Group names can be used in commands to apply the same permissions or settings to multiple users at once.

- GID (Group ID) is a unique numerical identifier assigned to each group in the system.
- It is used by the operating system to manage group memberships and permissions.
- Each group has a corresponding GID that differentiates it from other groups.
- Group members are the users who belong to a specific group.
- You can list members of a group by examining the /etc/group file or using commands like getent group <groupname>.
- Being a member of a group grants users the permissions associated with that group.
- A group password allows users to join a group if they know the password, enhancing access control.
- This feature is rarely used in modern systems, as managing access through user and group permissions is typically sufficient.
- Group passwords are set in the /etc/group file but may not be relevant for all distributions.
- Group ownership refers to the association of a file or directory with a specific group.
- The group ownership can affect the access permissions of the file for users belonging to that group.
- You can change the group ownership of a file using the chgrp command.
- Group permissions determine what actions members of the group can perform on a file or directory (read, write, execute).
- Permissions are set using the chmod command, and the group permissions can be viewed using ls -l.
- Group permissions can help manage access to shared resources among users.
- A group expiration date can be set to limit the time a group can exist or to restrict access to the group after a certain date.
- The expiration date can be managed using commands like chage or by editing the /etc/gshadow file.
- Setting expiration dates is useful for temporary groups or project-based access.

While creating group attributes consider:

- choose a meaningful and descriptive name for the group that reflects its purpose (e.g., dev_team, admins).
- Ensure that the Group ID is unique and does not conflict with other groups. You can either assign a specific GID or let the system automatically allocate one. Use the groupadd -g GID groupname command if you need to specify a GID.
- Each user has one primary group. When a user creates files, the group ownership defaults to their primary group.

- Users can belong to multiple secondary groups. Add users to the group with the usermod -aG groupname username command.
- Make sure that adding a user to a group doesn't remove them from existing groups by using the -aG option.
- Ensure that directories or files that the group should manage have their group ownership set correctly.
- Define read, write, and execute permissions for the group. Group permissions control what actions users in the group can perform on files and directories.
- If a directory is used by multiple users, setting the setgid permission ensures that new files and subdirectories created within inherit the group ownership.

Description of user and group management

- Managing users and groups is essential for maintaining system security, ensuring proper access, and organizing users.
- Creates a user with UID 1002, adds them to the developers group, and sets their default shell to bash.
- Modify existing user attributes, such as adding them to a secondary group.
- Manage file and directory permissions for users and groups.

While managing users and groups:

- Use useradd to create new users and specify important attributes like home directory (-m), default shell (-s), and user password (passwd).
- Ensure that each user is given the appropriate login environment and permissions.
- Use groupadd to create groups based on departmental or role-based needs.
- Assign users to groups using usermod -aG to manage access controls more efficiently.
- Ensure that each user is added to the correct group for their department, ensuring access to shared resources and limiting unauthorized access to sensitive data.
- Use chmod and chown to control access to files and directories.
- Set appropriate permissions based on department requirements with 770 for directories that need to be private within a department and 777 for directories that should be accessible to all users (like shared collaboration spaces).
- Create shared directories with open permissions (chmod 777) for collaboration.
- Ensure sensitive files are stored in department-specific directories with restricted access.
- Regularly update group memberships as users join or leave the organization.
- Change user passwords periodically for enhanced security.
- Use gpasswd -d to remove users from specific groups when needed.

- Test permissions for users to ensure they can only access authorized directories.
- Use commands like cd to verify that users are blocked from unauthorized areas and have proper access to shared resources.



Application of learning 2.3.

A team of developers is working on a shared project. All team members should have read/write access to the project folder, but other users should only have read access. Your tasks are:

a) Create a new group for the development team.
b) Add the developers to the group.
c) Change the ownership and permissions of the project folder so that the devteam group can modify files.
d) Ensure other users have only read access

Indicative content 2.4: Application of files and directories permissions

Duration: 15 hrs

Theoretical Activity 2.4.1: Description of files and directories permissions

Tasks:

1: Answer the following questions:

i. What does the read permission allow a user to do with a file or directory in a Linux system?
ii. How does the write permission affect a user's ability to interact with files and directories in Linux?
iii. What is the significance of the execute permission for files and directories in a Linux environment?
iv. Explain how symbolic notation is used to represent file permissions in Linux, including the meaning of r, w, and x.
v. How is octal notation used to represent file permissions in Linux, and what does the octal number 755 signify for a file or directory?

2: Present the findings to trainer, your colleagues or to the whole class
3: Ask questions where necessary.
4: Read the key readings 2.4.1

Key readings 2.4.1:Description of files and directories permissions

Types of Permissions:

- **Read (r)**: Allows a file to be read or viewed.
- **Write (w)**: Allows a file to be modified.
- **Execute (x)**: Allows a file to be executed (run as a program/script) or a directory to be accessed (e.g., changing into it with cd).

2. Permission Levels:

- **User/Owner** (u): The individual who owns the file or directory.
- **Group** (g): A set of users who share access to the file or directory.
- **Others** (o): All other users on the system who are not the owner or part of the group.

3. Viewing Permissions:

- Use the ls -l command to view the permissions of files and directories. It shows a string like rwxr-xr-- representing permissions for **owner**, **group**, and **others** respectively.

4. Changing Permissions (chmod):

- Permissions can be altered using chmod either:
  - **Symbolically** (e.g., chmod u+r to add read permission for the user).
  - **Numerically** (e.g., chmod 755, where each digit represents the permissions for **user**, **group**, and **others** in the form of a sum: r=4, w=2, x=1).

5. Changing Ownership (chown):

- Ownership of a file or directory can be changed using the chown command. This includes:
  - **Changing the user** (e.g., chown user filename).
  - **Changing the group** (e.g., chown user:group filename).

6. Special Permissions:

- **Setuid (s)**: Applied to executables, allows users to execute the file with the permissions of the file's owner.
- **Setgid (s)**: Applied to files or directories, causes new files to inherit the group of the directory.
- **Sticky Bit (t)**: Used on directories to ensure only the file owner can delete or modify files, even if other users have write permissions.

7. Directory Permissions:

- **Read (r)**: Allows users to list the contents of a directory.
- **Write (w)**: Allows users to create, delete, or rename files within the directory.
- **Execute (x)**: Allows users to enter or access the directory, enabling cd to the directory.

8. Recursive Permission Changes:

- To change permissions for all files and directories within a directory, use the -R (recursive) option with chmod or chown.

9. Sticky Bit Use Case:

- Commonly applied to shared directories like /tmp. It ensures that users can only delete or modify their own files within the directory.

10. Default Permissions (umask):

- The umask command sets the default permissions for newly created files and directories. It subtracts permissions from the maximum possible (usually 666 for files and 777 for directories).

11. Setting Group Permissions with g+s:

- Applying the **setgid** (g+s) on a directory ensures all files created in the directory inherit the group of the directory rather than the user's primary group.

12. Limiting Access:

- To ensure **only specific users** or groups can modify files, but everyone else has only **read** access:



Practical Activity 2.4.2: Applying files and directories permissions



Task:

1: Individual, perform the following activity:

You are the system administrator for a small web development team of five developers working on a shared project. Your task is to configure permissions for a shared project directory so that:

a) The developers can collaborate by editing files within the shared directory.
b) Other users on the server should only have read access to the directory.
c) Developers should be able to create and edit files without needing to manually adjust permissions each time. 2: Read the key reading 2.4.2

3: Configure Permissions on file and directories

4: Present your work to the trainer and your classmate

4: Perform the task provided in application of learning 2.4



Key readings 2.4.2: Applying files and directories permissions

1. Understanding Linux Permissions:

- **File Permissions:** Each file in Linux has three types of permissions:
  - **Read (r)**: Allows viewing the content of the file.
  - **Write (w)**: Allows modifying the content of the file.
  - **Execute (x)**: Allows running the file as a program (for executable files).
- **Directory Permissions:** For directories, the permissions have specific meanings:
  - **Read (r)**: Allows listing the contents of the directory.
  - **Write (w)**: Allows creating, deleting, or renaming files within the directory.
  - **Execute (x)**: Allows accessing the directory (navigating into it).

2. Checking Current Permissions:

- Use the ls -l command to view the permissions of files and directories. The output will look like this: drwxr-xr-- 2 user group 4096 Sep 29 10:00 mydirectory
  -rw-r--r-- 1 user group 123 Sep 29 10:00 myfile.txt
- The first character indicates whether it's a directory (d) or a file (-). The next nine characters are the permissions for user, group, and others.

3. Creating Users and Groups:

- **Create Groups:** Use the groupadd command to create new groups as needed.

For example: sudo groupadd devteam

- **Add Users to Groups:** Use usermod to add users to a group:
  sudo usermod -aG devteam username

4. Creating Directories and Files:

- Create directories for projects or shared resources: sudo mkdir /path/to/project_directory
- Create files within those directories: sudo touch /path/to/project_directory/file.txt

5. Changing Ownership:

- **Change the Owner:** Use chown to change the owner of a file or directory: sudo chown username /path/to/file
- **Change Group Ownership:** Use chown with the group option: sudo chown :groupname /path/to/directory

6. Setting Permissions with chmod:

- **Symbolic Mode:** Use chmod with letters to change permissions: sudo chmod u+rwx,g+rx,o+r /path/to/directory # User gets read/write/execute, group gets read/execute, others get read
- **Numeric Mode:** Use numbers to set permissions (where r=4, w=2, x=1):

sudo chmod 775 /path/to/directory # 7=4+2+1 for user, 7=4+2+1 for group, 5=4+1 for others

7. Setting Special Permissions:

- **Setgid (g+s):** Ensures files created in a directory inherit the group of the directory: sudo chmod g+s /path/to/directory
- **Setuid (s):** Allows users to execute a file with the permissions of the file's owner: sudo chmod u+s /path/to/executable
- **Sticky Bit (t):** Typically used on directories to allow only the owner to delete or rename files: sudo chmod +t /path/to/directory

8. Verifying Permissions:

- After applying permissions, verify them using:
  - ls -ld /path/to/directory
  - ls -l /path/to/directory

9. Testing Access:

- Log in as different users or switch users using the su command to test whether permissions are correctly applied: su - username
- Attempt to read, write, or execute files as different users to ensure permissions are functioning as intended.

10. Monitoring and Auditing:

- Set up logging to monitor file access and changes, which can help in auditing and security:
  - sudo apt install auditd

- sudo auditctl -w /path/to/directory -p rwxa -k my_audit_key

Steps:

1. **Create the Group:**
   sudo groupadd devteam
2. **Add Developers:**
   sudo usermod -aG devteam dev1
   sudo usermod -aG devteam dev2
3. **Create the Shared Directory:**
   bash
   Copy code
   sudo mkdir /projects/devteam
4. **Change Ownership and Set Permissions:**
   sudo chown : devteam /projects/devteam
   sudo chmod 775 /projects/devteam
   sudo chmod g+s /projects/devteam
5. **Verify Setup:** ls -ld /projects/devteam
6. **Test Access:** create a file: touch /projects/devteam/testfile.txt



Points to Remember

Description of files and directories permission

- The read permission allows a user to view the contents of a file or directory, a user with read permission can open and read the file's contents, and a user with read permission on a directory can list the files and subdirectories within it.
- The write permission allows a user to modify a file or directory, a user with write permission can edit, delete, or create new content in the file, and a user with write permission can add, remove, or rename files and subdirectories within the directory.
- The execute permission allows a user to run a file as a program or script, a user with execute permission can execute the file, provided it is a script or a binary executable, and a user with execute permission on a directory can access the directory's contents and perform actions on the files within it (but not necessarily list them unless they also have read permission).
- Symbolic notation uses letters to represent permissions: r (read), w (write), and x (execute) and permissions are typically represented in a string of 10 characters, such as -rwxr-xr--.
- Octal notation is a numerical representation of file permissions, using numbers to indicate permission levels, and each permission type corresponds to a specific value Read = 4, Write = 2, and Execute = 1

Applying files and directories permissions

- Ensure that all developers are members of the devteam group to facilitate collaboration.
- The shared project directory is set up at the specified path (/path/to/shared/project). Confirm that all team members are aware of this location.
- The directory permissions are set to 775, allowing owner and group to read, write, and execute permissions and others to read-only permissions.
- The setgid bit is applied to the project directory, ensuring that any new files or directories created within inherit the group ownership of devteam.
- Newly created files in the shared directory will automatically have the group set to devteam, which means developers won't need to manually adjust permissions for each new file.
- Other users on the server will only have read access to the shared project directory and its files, maintaining security and preventing unauthorized modifications.
- Regularly communicate with the team about the directory usage and any changes in group membership or permissions.
- Periodically check the permissions and ownership of the project directory to ensure they remain correctly configured, especially after any system updates or changes.
- Establish a backup strategy for the project directory to safeguard against data loss. This may include regular snapshots or backups to a separate storage location.

- Document the changes made, including the creation of the devteam group, the directory path, and the applied permissions for future reference and onboarding new team members.



Application of learning 2.4.

You are managing a system for a financial institution that requires strict access controls to protect sensitive data, such as customer records and financial reports. Only authorized personnel should have the ability to read and modify these files, while others should have no access at all. Your task is to Secure File Storage for Sensitive Data.



Learning Outcome 2 end assessment

Written assessment

Choose the right answer in the questions below:

1) **What command is used to list files in a directory?**
   a) ls
   b) list
   c) dir
   d) show
2) **Which option with the ls command displays all files, including hidden files?**
   a) -l
   b) -a
   c) -h
   d) -R
3) **What command is used to create a new directory?**
   a) newdir
   b) mkdir
   c) createdir
   d) mkfolder
4) **Which command removes an empty directory?**
   a) rmdir
   b) rm
   c) del
   d) remove
5) **To delete a file named example.txt, which command would you use?**
   a) rm example.txt
   b) del example.txt
   c) delete example.txt
   d) remove example.txt
6) **What command would you use to copy a file named file1.txt to file2.txt?**
   a) cp file1.txt file2.txt
   b) copy file1.txt file2.txt
   c) mv file1.txt file2.txt
   d) cp file1 to file2
7) **Which command moves a file named document.pdf to a directory named backup?**
   a) mv document.pdf backup/
   b) move document.pdf backup/
   c) cp document.pdf backup/
   d) transfer document.pdf backup/
8) **What command would you use to rename a file in Linux?**

a) mv
b) rename
c) renamefile
d) change

9) **Which command is used to display the contents of a text file?**
a) show
b) cat
c) read
d) display

10) **What does the command touch filename do?**
a) Deletes the file
b) Creates an empty file or updates the timestamp
c) Edits the file
d) Displays the file

11) **Which command is used to change file permissions?**
a) chmod
b) chown
c) changeperm
d) setperm

12) **What does the -R option do when used with rm?**
a) Removes files only
b) Removes directories recursively
c) Renames files
d) Creates directories

13) **How do you change to the parent directory using the cd command?**
a) cd /
b) cd ..
c) cd -
d) cd parent

14) **Which command lists the current directory you are in?**
a) pwd
b) ls
c) dir
d) currentdir

15) **What does the command ls -l do?**
a) Lists files with detailed information
b) Lists hidden files
c) Lists files in reverse order
d) Lists only directories

16) **Which option would you use with chmod to add execute permission for the user?**
a) +x

b) +u
c) +r
d) +w

17) **What is the default permission for newly created files?**
a) 755
b) 644
c) 700
d) 777

18) **Which command creates a hard link to a file?**
a) ln -s
b) link
c) ln
d) hardlink

19) **What does ls -lh do?**
a) Lists files in a hidden format
b) Lists files with human-readable sizes
c) Lists files with permissions
d) Lists only large files

20) **What does the command chmod 777 filename do?**
a) Grants all permissions to everyone
b) Denies all permissions to everyone
c) Grants read and write permissions only
d) Grants read permission only

21) **How can you view the contents of a directory in a tree-like format?**
a) tree
b) ls -tree
c) dir
d) view

22) **What command would you use to search for a specific file named myfile.txt?**
a) find myfile.txt
b) locate myfile.txt
c) search myfile.txt
d) grep myfile.txt

23) **Which command is used to change the owner of a file?**
a) chperm
b) chown
c) chmod
d) ownerchange

24) **What does the command cp -r source/ destination/ do?**
a) Copies files only
b) Copies directories and their contents recursively

c) Moves files

d) Renames files

25) **What is the effect of running rm -f filename?**

a) Prompts for confirmation before deleting

b) Deletes the file without prompting for confirmation

c) Forces deletion of read-only files

d) Recursively deletes directories

26) **How do you view a file in the terminal one page at a time?**

a) cat

b) more

c) view

d) less

27) **What command would you use to create a symbolic link to a file?**

a) ln

b) ln -s

c) symlink

d) link

28) **Which command can be used to view the disk usage of files and directories?**

a) du

b) df

c) usage

d) ls -du

29) **What is the purpose of the command file filename?**

a) Displays the contents of the file

b) Shows the type of file

c) Edits the file

d) Deletes the file

30) Match the Linux command in **Column A** with its **correct description** in **Column B**:

Column A (Linux Commands)	**Column B (Descriptions)**	**Answers**
1. ls	A. Change the current directory	**….**
2. cd	B. List files and directories in the current directory	**….**
3. mkdir	C. Remove a file or directory	**….**
4. touch	D. Create a new file	**….**
5. rm	E. Move or rename a file	**….**
6. mv	F. Create a new directory	**….**

7. cp	G. Copy files or directories	….
8. pwd	H. Print the full path of the current directory	….
9. rmdir	I. Remove an empty directory	….
10. chmod	J. Change the permissions of a file or directory	….

Practical assessment

You are the system administrator for a research lab that conducts various scientific experiments. The lab consists of several teams, each working on different projects that generate numerous files, including datasets, reports, and analysis scripts. Efficient management of these files and directories is crucial for collaboration, data integrity, and security. Your task is to Set up an organized file and directory structure, apply Basic Linux commands, Create users and groups, Set directory permissions and Implement data security measures



References

Barrett, D. J. (2016). *Linux pocket guide* (3rd ed.). O'Reilly Media.

Blum, R., & Bresnahan, C. (2021). *Linux command line and shell scripting bible* (4th ed.). Wiley.

Cobbaut, P. (2019). *Linux fundamentals*. Linux Training Academy.

Kerrisk, M. (2010). *The Linux programming interface: A Linux and UNIX system programming handbook*. No Starch Press.

Negus, C. (2020). *Linux Bible* (10th ed.). Wiley.

Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *Linux administration handbook* (6th ed.). Prentice Hall.

Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *UNIX and Linux system administration handbook* (5th ed.). Prentice Hall.

Red Hat. (n.d.). *Manage permissions*. Retrieved from www.redhat.com/en/blog/manage-permissions

Shotts, W. E., Jr. (2012). *The Linux command line: A complete introduction*. No Starch Press.

TechRepublic. (n.d.). *How to create users and groups in Linux from the command line*. Retrieved from www.techrepublic.com/article/how-to-create-users-and-groups-in-linux-from-the-command-line/

Ward, B. (2021). *How Linux works: What every superuser should know* (3rd ed.). No Starch Press.

DigitalOcean. (n.d.). *Linux commands*. Retrieved from www.digitalocean.com/community/tutorials/linux-commands


_all:
@umask $(REQ_UMASK) && \
$(MAKE) -C $(CANONICAL_CURDIR) --no-print-directory
else # umask / $(CURDIR) /$(@)
#This is our default rule, so must come first
all:
.PHONY: all
Set and export the version string
export BRW_VERSION := 2019.05
# #Actual time the release is cut (for reproducible builds)
BB2_VERSION_EPOCH = 1559462000
#Save running make version since it's clobbered by the make package
RUNNING_MAKE_VERSION := $(MAKE_VERSION)
#Check for minimal make version(note: this check will break at make 10.X)
MIN_MAKE_VERSION = 3.81
ifneq ($(firstword$(sort$(RUNNING_MAKE_VERION)$(MIN_MAKE_VERION))),$(MIN_MAKE_
VERSION))

Indicative contents
3.1 Configuration of DNS service
3.2 Configuration of DHCP service
3.3 Configuration of File Server service
3.4 Configuration of Web Server
3.5 Configuration of Mail Server service

Key Competencies for Learning Outcome 3: Manage Linux services

Knowledge	Skills	Attitudes
● Description of DNS service ● Description of DHCP service ● Description of File Server service ● Description of Web Server ● Description of Mail Server service	● Configuring DNS service ● Configuring DHCP service ● Configuring File Server service ● Configuring Web Server ● Configuring Mail Server service	● Being innovative ● Having creativity ● Having a teamwork spirit ● Being a problem solver ● Having patience ● Having critical thinking ● Being passionate



Duration: 35 hrs

Learning outcome 3 objectives:



By the end of the learning outcome, the trainees will be able to:

1. Configure DNS service correctly based on functionality
2. Configure DHCP service appropriately based on Purpose and Functionality
3. Configure File Server service appropriately based on user requirements
4. Configure Web Server appropriately based on functionality
5. Configure Mail Server service correctly based on Purpose and Functionality



Resources

Equipment	Tools	Materials
● Computer ● Projector	● Ubuntu server ● Hypervisors (VMware or VirtualBox)	● Internet

Indicative content 3.1: Configuration of DNS Service

Duration: 7 hrs

Theoretical Activity 3.1.1: Description of DNS services

Tasks:

1: Answer following questions:

i. How do you install a DNS server package using the package manager specific to your Linux distribution (e.g., apt, yum, or dnf)?
ii. How do you configure the necessary DNS server files, such as named.conf, to define DNS zones and server settings?
iii. How do you configure common DNS records (e.g., A, MX, PTR) within the DNS zone files?
iv. What tools and methods can you use to test the DNS configuration and ensure proper resolution (e.g., dig, nslookup, checking logs)?

2: Present the findings to trainer, your colleagues or to the whole class

3: Address any questions or concerns

4: Ask them to read the Key readings 3.1.1 in their manuals

Key readings 3.1.1.: Description of DNS services

1. Ensure System is Up-to-Date

Before installing any packages, run an update on your system to ensure you have the latest package information.

2. To install a DNS server package using the package manager specific to your Linux distribution (e.g., apt, yum, or dnf) Use the appropriate package manager**:**

- ✓ For Debian-based systems (Ubuntu, etc.): sudo apt install bind9
- ✓ For Red Hat-based systems (CentOS, Fedora): sudo yum install bind or sudo dnf install bind

3. **Verify installation**: Check with systemctl status named (or bind9) to ensure the DNS service is running.

- ✓ **Install dependencies**: Some DNS services may require additional packages like bind-utils or dnsutils.
- **Main configuration file**: named. conf (often located in /etc/bind/ or /etc/named/)
- **Define zones**: Add zone configurations for forward and reverse lookup zones,

- **Security settings**: Optionally configure access control lists (ACLs) for restricting queries, forwarding, or recursion.
- **Log file setup**: Ensure logging is configured to monitor DNS queries and errors.

4. **To** configure common DNS records within the DNS zone files
   - **Zone file location**: Typically located in /var/named/ or /etc/bind/.
   - **A record**: Maps a domain to an IP address.
   - **MX record**: Defines mail server for the domain.
   - **PTR record**: Used for reverse DNS lookup (in reverse zone file).
   - **SOA record**: Defines the Start of Authority, which should be properly configured for each zone.

5. **Testing tools**:

   - Use dig to query specific DNS records: dig example.com A or dig -x 192.168.1.10 for reverse lookup.
   - Use nslookup to test domain resolution: nslookup example.com.
   - **Check DNS service status**: Run systemctl status named (or bind9) to check if the DNS service is active.
   - **Log file checks**: Look into DNS logs (/var/log/messages or /var/log/syslog) for any error messages or query logs.
   - **Syntax checks**: Use named-checkconf for configuration file syntax and named-checkzone to validate zone file correctness.



Practical Activity 3.1.2: Installing of DNS Server



Task:

1. Trainer begins by demonstrating the activity and explaining the tasks that trainees will perform.

As an IT professional, you have been assigned to install, configure, and test a DNS server on a Linux machine for a small company. The server will be used to resolve internal domain names and manage the company's local network name resolution.

2. Read the Key readings 3.2.2

3. install, configure file and perform DNS configuration test.

4. Ask them to read the Key readings 3.2.2 in their manuals.



Key readings 3.1.2 Installing of DNS Server

- ✓ The most common DNS server package is BIND (Berkeley Internet Name Domain). Make sure to install it with the appropriate package manager for your distribution
- ✓ Always update your system before installing the DNS server package to avoid compatibility issues.
- ✓ Ensure all required dependencies for the DNS package are installed. Most package managers will handle this automatically, but it's important to verify if any libraries are missing, especially for custom setups.
- ✓ Ensure that DNS-related ports (UDP and TCP on port 53) are open on your firewall
- ✓ After installation, make sure the DNS server configuration files (e.g., /etc/bind/named.conf) have the proper permissions. These files should typically be owned by the bind or named user, depending on the Linux distribution.
- ✓ After installation, you need to configure the server. The key files include:
  - **Main Configuration File**: /etc/bind/named.conf.local or /etc/named.conf
  - **Zone Files**: These files contain DNS records (A, CNAME, MX, etc.) for your domains.
  - **Options File**: Configures general server settings like recursion, forwarding, etc.
- ✓ Enable and start the DNS service to make sure it runs on boot and immediately starts after installation.
  - DNS (Domain Name System) resolves human-readable domain names (e.g., example.com) into IP addresses (e.g., 192.168.1.1).
  - DNS server (authoritative or recursive), DNS client, zone files, and resource records (e.g., A, MX, CNAME).
  - Install the DNS server software package (e.g., BIND on Linux).
    - Define forward and reverse zones in the main config file.
    - Forward zone: Resolves domain names to IP addresses.
    - Reverse zone: Resolves IP addresses to domain names.



Points to Remember

Description of DNS services

- Install the DNS server software package (e.g., BIND on Linux).
- Define forward and reverse zones in the main config file.
- Forward zone: Resolves domain names to IP addresses.
- Reverse zone: Resolves IP addresses to domain names.
- Allow DNS traffic through the firewall on port 53 for both TCP and UDP:
- Start or restart the DNS service after configuration changes
- Regularly monitor DNS logs (/var/log/syslog or /var/named/data/named.run) for troubleshooting.
- Use tools like systemctl status bind9 to check DNS service status.
- Before installing any packages, run an update on your system to ensure you have the latest package information.
- sudo apt update # For Debian/Ubuntu or sudo yum update # For CentOS/RHEL
- The most common DNS server package is BIND (Berkeley Internet Name Domain). Make sure to install it with the appropriate package manager for your distribution:
- After installation, check that the DNS server software was installed correctly by using the package manager's verification tool or checking the service status.
- Familiarize yourself with key BIND configuration files
- Zones define how the DNS server handles requests for specific domains. When configuring zones
- DNS configuration and zone files should have proper permissions to avoid unauthorized access or accidental modification. They should typically be owned by the bind or named user and group, depending on your Linux distribution.
- Even a small syntax error in configuration files can prevent the DNS service from starting. Double-check zone files, especially SOA (Start of Authority) records, as they are sensitive to formatting.
- Ensure you are using the correct record types for your needs like Map domain names to IP addresses, create alias names for domains, specify mail servers for a domain., Identify authoritative name servers for the domain and Used for reverse DNS lookups (IP to domain name mapping).
- Set appropriate TTL values for each DNS record. TTL controls how long a DNS resolver caches the information.
- Mistakes in the structure or format of these records can cause issues with DNS resolution.

Installing of DNS Server Services

- DNS (Domain Name System) resolves human-readable domain names (e.g., example.com) into IP addresses (e.g., 192.168.1.1).
- DNS server (authoritative or recursive), DNS client, zone files, and resource records (e.g., A, MX, CNAME).
- Install the DNS server software package (e.g., BIND on Linux).
- Define forward and reverse zones in the main config file.
- Forward zone: Resolves domain names to IP addresses.
- Reverse zone: Resolves IP addresses to domain names.
- Allow DNS traffic through the firewall on port 53 for both TCP and UDP:
- Start or restart the DNS service after configuration changes
- Regularly monitor DNS logs (/var/log/syslog or /var/named/data/named.run) for troubleshooting.
- Use tools like systemctl status bind9 to check DNS service status.
- Before installing any packages, run an update on your system to ensure you have the latest package information.
- sudo apt update # For Debian/Ubuntu or sudo yum update # For CentOS/RHEL
- The most common DNS server package is BIND (Berkeley Internet Name Domain). Make sure to install it with the appropriate package manager for your distribution:
- After installation, check that the DNS server software was installed correctly by using the package manager's verification tool or checking the service status.
- Familiarize yourself with key BIND configuration files
- Zones define how the DNS server handles requests for specific domains. When configuring zones
- DNS configuration and zone files should have proper permissions to avoid unauthorized access or accidental modification. They should typically be owned by the bind or named user and group, depending on your Linux distribution.
- Even a small syntax error in configuration files can prevent the DNS service from starting. Double-check zone files, especially SOA (Start of Authority) records, as they are sensitive to formatting.
- Ensure you are using the correct record types for your needs like Map domain names to IP addresses, create alias names for domains, specify mail servers for a domain., Identify authoritative name servers for the domain and Used for reverse DNS lookups (IP to domain name mapping).
- Set appropriate TTL values for each DNS record. TTL controls how long a DNS resolver caches the information.

- Mistakes in the structure or format of these records can cause issues with DNS resolution.



Application of learning 3.1.

You have been tasked with setting up a DNS server for your organization's local domain, **mv_tvet.ac.rw**, with the server IP address 192.168.1.20. You are required to install, configure the DNS server, create necessary DNS records, and perform a configuration test to ensure the system is working correctly.

Indicative content 3.2: Configuration of DHCP service

Duration: 7 hrs

Theoretical Activity 3.2.1: Description of DHCP services

Tasks:

1: Answer the following questions:

i. What are the four steps involved in the DHCP handshake process?
ii. How do you install a DHCP server package on a Linux system?
iii. What is the main configuration file for a DHCP server on Linux?
iv. What parameters must be set when configuring a DHCP server?

2: Ask trainees to present their findings to trainer, workshop assistant, classmate or whole class.

3: Address any questions or concerns.

4: Ask them to read the Key readings 3.2.1 in their manuals

Key readings 3.2.1.: Description of DHCP services

1. **Description of DHCP Handshake Operations**
   - **DHCP Discovery**: Client broadcasts a request to find a DHCP server.
   - **DHCP Offer**: DHCP server responds with an IP address offer and network settings (subnet mask, gateway, DNS, etc.).
   - **DHCP Request**: Client selects an offer and sends a request to the server, asking to use the provided IP.
   - **DHCP Acknowledgment**: DHCP server confirms the IP assignment and sends a final acknowledgment, completing the handshake.
2. **Install DHCP Server Package**
   - **Linux Systems**: Use isc-dhcp-server for Linux-based systems like Ubuntu.
     - Install with: sudo apt-get install isc-dhcp-server
   - **Windows Systems**: The DHCP server role is installed via the Server Manager or PowerShell.
   - **Ensure the server has a static IP**: Dynamic IP on the DHCP server itself may cause issues during network configuration.
3. **DHCP Configuration Files**
   - **Location**:
     - ✓ **Linux**: /etc/dhcp/dhcpd.conf is the primary configuration file.
     - ✓ **Windows**: Managed through the DHCP Manager GUI or PowerShell.

- **Critical Settings**: Define IP range (scope), lease times, DNS servers, default gateways, and domain names.
- **Backup Configuration**: Always keep a backup of the configuration file before making changes to avoid downtime if issues arise.

4. **Set the DHCP Server Parameters**
   - **IP Range**: Set the range of IP addresses to be assigned to clients.
     - Example: range 192.168.1.100 192.168.1.200;
   - **Lease Time**: Define default and maximum lease times to control how long IP addresses are assigned.
     - Example: default-lease-time 600; max-lease-time 7200;
   - **Network Details**: Include the subnet mask, default gateway, and DNS server options.
     - Example: option routers 192.168.1.1; option domain-name-servers 192.168.1.20;
   - **Domain Name**: Set the domain name to be assigned to clients.
     - Example: option domain-name "mv_tvet.ac.rw";
   - **Static IP Reservations**: Reserve specific IPs for important devices (servers, printers) by mapping MAC addresses to IP addresses.



Practical Activity 3.2.2: Configuring DHCP services



Task:

1: Perform the below task.

Your school need to install and configure a DHCP (Dynamic Host Configuration Protocol) server to automatically assign IP addresses and network configuration settings to devices on the school's network. You are required to install the DHCP server software, configure it, and set specific DHCP server parameters to ensure efficient IP address management.

2: Present the findings to trainer, your colleagues or to the whole class

3: Address any questions or concerns

4: Ask them to read the Key readings 3.2.2 in their manuals



Key readings 3.2.2: Configuring DHCP services

1. **Description of DHCP Handshake Operations**: The DHCP handshake involves a sequence of messages between a DHCP client and server:
    - **Discovery**: The client broadcasts a DHCPDISCOVER message to find available DHCP servers.
    - **Offer:** The server responds with a DHCPOFFER message, offering an IP address and other network configurations.
    - **Request:** The client responds with a DHCPREQUEST message, requesting the offered IP address.
    - **Acknowledgment:** The server sends a DHCPACK message, confirming the IP address allocation and providing the client with network settings.
    - **Install DHCP Server Package:** On a Linux system, you can typically install the DHCP server package using a package manager.
2. **DHCP Configuration Files**: The main configuration file for the DHCP server is usually located at /etc/dhcp/dhcpd.conf. This file contains settings for network pools, leases, and options.
3. **Set the DHCP Server Parameters**: Configure the DHCP server by editing the configuration file (/etc/dhcp/dhcpd.conf). You need to set parameters such as:
    - **Subnet**: Define the subnet and IP address range.
    - **Range**: Specify the range of IP addresses to be assigned to clients.
    - **Options**: Set options like default gateway, DNS servers, and lease time.



Points to Remember

Description of DHCP services

- The DHCP handshake process involves four steps: Discover, Offer, Request, and Acknowledgement.
- To install a DHCP server on Linux, use package managers like apt for Debian-based systems or yum/dnf for Red Hat-based systems.
- The main configuration file for the DHCP server on Linux is located at /etc/dhcp/dhcpd.conf.

- Key parameters to configure include the subnet, range, router, DNS server, and lease time.
- DHCP Discover is broadcast by the client to find available servers, while the DHCP Offer comes from the server with an IP proposal.
- Setting the broadcast address helps ensure proper communication within the subnet.
- Set the domain name to be assigned to clients.
- The DHCP server is properly installed and running.
- The subnet, IP range, router (gateway), DNS server, and lease time parameters are correctly set in the configuration file (/etc/dhcp/dhcpd.conf).
- The server is bound to the correct network interface.
- Test that clients can successfully obtain IP addresses and related settings through the DHCP process.
- Ensure the firewall rules allow DHCP traffic (UDP ports 67 and 68).
- Regularly monitor logs for any errors or issues with IP leasing or conflicts.

While you are Configuring DHCP services

- To install a DHCP server on Linux, use package managers like apt for Debian-based systems or yum/dnf for Red Hat-based systems.
- The main configuration file for the DHCP server on Linux is located at /etc/dhcp/dhcpd.conf.
- Key parameters to configure include the subnet, range, router, DNS server, and lease time.
- DHCP Discover is broadcast by the client to find available servers, while the DHCP Offer comes from the server with an IP proposal.
- Setting the broadcast address helps ensure proper communication within the subnet.
- Set the domain name to be assigned to clients.



Application of learning 3.2.

You have been tasked with setting up and configuring a DHCP server on a Linux machine to automatically assign IP addresses and network configuration to devices at **M&V TVET School**. The network needs to serve staff and student computers, providing automatic IP addressing.

Indicative content 3.3: Configuration of File Server Service

Duration: 7 hrs

Theoretical Activity 3.3.1: Description of file Server service

Tasks:

1: Answer the following questions:

i. What is SMB/CIFS, and how is it typically used in network environments?
ii. Explain the role of NFS in file sharing on Unix/Linux networks?
iii. What is FTP, and how does it differ from other file sharing protocols like SMB and NFS?

2: Present the findings to trainer, your colleagues or to the whole class

3: Address any questions or concerns

4. Read key reading 3.3.1 in their manuals.

Key readings 3.3.1: Description of file Server service

1. **Definition of SMB/CIFS**
   SMB/CIFS is a network file sharing protocol primarily used for sharing files, printers, and serial ports between systems, especially in Windows environments.
2. **Key Features**:
   - **Cross-Platform Support**: Although originally designed for Windows, SMB is now supported on multiple platforms, including Linux and macOS.
   - **Authentication**: Supports user authentication to control access to shared resources.
   - **File and Record Locking**: Prevents multiple users from editing the same file simultaneously, ensuring data integrity.
   - **Versioning**: SMB has several versions, such as SMBv1, SMBv2, and SMBv3, with newer versions offering better security and performance.
   - **Encryption**: SMBv3 introduces encryption to protect data in transit, reducing the risk of eavesdropping.
   - **Windows Integration**: Tight integration with Active Directory (AD) for managing user permissions and access controls.
3. **Considerations**:
   - Ensure that **SMBv1** is disabled as it has known security vulnerabilities.
   - SMB is **chattier** (higher network overhead) compared to other protocols.

- Firewall settings should be configured to allow SMB ports (445 for direct SMB and 139 for NetBIOS over TCP).

4. **NFS (Network File System):** NFS is a file-sharing protocol widely used in UNIX/Linux environments to share directories and files across a network.
5. **Key Features**:

   - **Stateless Protocol**: NFS is designed to be stateless, meaning the server doesn't maintain session information for clients. This makes it easier to recover from client or server failures.
   - **File Locking**: NFS provides optional support for file locking, ensuring that concurrent file access doesn't lead to data corruption.
   - **Permissions and Access Control**: NFS relies on the Unix permissions system (UIDs and GIDs) to manage access control.
   - **Versions**: NFS has several versions (NFSv3, NFSv4), with NFSv4 offering improvements in security (support for Kerberos authentication) and performance.
   - **Integration with Linux**: It is highly integrated into Linux and UNIX operating systems and works well in environments where centralized file storage is needed.
6. **Considerations**:

   - **Firewall Configuration**: Ensure that the required NFS ports (111, 2049) are open.
   - **Security**: Use NFSv4 for better security (Kerberos-based authentication) and encrypted file transfers.
   - **UID/GID Mapping**: Ensure proper user ID mapping between clients and servers to avoid permission issues.
   - NFS might not perform well in environments where there is high network latency due to its stateless nature.

7. **FTP (File Transfer Protocol):** FTP is a standard protocol used for transferring files between a client and server over a TCP-based network, such as the internet.
8. **Key Features**:

   - **Basic Authentication**: FTP uses simple username and password authentication, though these credentials are transmitted in plaintext.
   - **Active vs. Passive Mode**: FTP supports two communication modes—active and passive. In active mode, the client opens a port for data transfer, while in passive mode, the server opens the port.
     - **Anonymous FTP**: Allows users to download public files from servers without providing login credentials.
     - **File Transfer**: FTP is ideal for large file transfers and supports both binary and ASCII modes for file transmission.

- **Directory Navigation**: FTP allows users to navigate the server's directory structure and upload/download files.

9. **Considerations**:

   - **Security**: FTP is inherently insecure as it transmits data, including passwords, in plaintext. For secure file transfers, consider using **SFTP (Secure FTP)** or **FTPS (FTP over SSL)**.
   - **Firewall Configuration**: Ensure appropriate ports (21 for command, 20 for data) are open, and configure firewalls accordingly for passive/active modes.
   - **Encryption**: Use FTPS or SFTP to encrypt data in transit for sensitive information.

10. FTP has high **latency sensitivity**, meaning network delays can negatively affect performance, especially in high-latency environments.



Practical Activity 3.3.2: Configuring file server service



Task:

1: Individually, Perform the below task:

You are tasked with installing and configuring file-sharing protocols SMB/CIFS, NFS, and FTP on a Linux server.

2: Read key reading 3.3.2.

3: install and configure file-sharing protocols SMB/CIFS, NFS, and FTP

4: Present the findings to trainer and whole class

5: Perform the task provided in application of learning 3.3.



Points to Remember

Description of file Server service

- SMB (Server Message Block) is a protocol for sharing files, printers, and other resources, primarily in Windows environments but supported across platforms like Linux through Samba. CIFS is a dialect of SMB used for file sharing over the internet.
- SMB/CIFS provides features like user authentication, access control, and remote file access. It operates over TCP/IP (port 445) and supports both TCP and UDP communication.

- NFS (Network File System) is a distributed file system primarily used in Unix/Linux environments for file sharing. It is stateless, uses port 2049, and is suitable for data centers and environments requiring centralized data access.
- FTP (File Transfer Protocol) is focused on file transfers, ideal for uploading/downloading files between hosts. It operates over port 21 for commands and uses additional ports for data transfer.
- Differences: SMB/CIFS and NFS focus on file sharing and resource management, whereas FTP is designed for file transfers. SMB/CIFS is common in Windows, NFS in Linux/Unix, and FTP is platform-independent but widely used in web hosting.

  transfers.

While Configuring file server service

- Install Samba (the open-source implementation of SMB/CIFS) using the package manager (apt install samba for Ubuntu/Debian or yum install samba for CentOS/RHEL).
- For Windows: SMB is built into Windows and can be enabled/disabled via "Turn Windows Features On or Off."
- Edit the smb.conf file in /etc/samba/ to define shared directories, access controls, and authentication settings.
- Ensure the Samba service is enabled to start automatically after a system reboot.
- Open the necessary firewall ports (TCP 445 for SMB, TCP 139 for NetBIOS over TCP).
- Test the configuration by connecting to the server using another machine with the SMB/CIFS protocol.
- Install the NFS server package (apt install nfs-kernel-server on Ubuntu/Debian or yum install nfs-utils on CentOS/RHEL).
- Define shared directories in the /etc/exports file. Specify which clients are allowed access and their permission levels (e.g., read-only or read-write).
- Open the necessary firewall ports (TCP/UDP 2049 for NFS).
- Test the NFS server by mounting the shared directories on a client machine and verifying file access
- Install an FTP server package (e.g., vsftpd or proftpd) using the package manager (apt install vsftpd or yum install vsftpd).
- Edit the FTP configuration file (e.g., /etc/vsftpd.conf) to set parameters like anonymous access, local user permissions, and file directories.
- Ensure that the FTP service starts automatically after system boot and is running properly.
- Open FTP ports (21 for command, 20 for data). Configure firewall rules for active or passive FTP mode, depending on the use case.

- Use FTPS (FTP over SSL) for encrypted file transfers to enhance security.
- Test the FTP server by connecting to it from a client using an FTP client or web browser and transferring files.
- Define the root directory for FTP users, where they will land upon connection. In vsftpd.conf, this is typically set using the local_root directive.
- Disable anonymous access (anonymous_enable=NO) for better security.
- Set up user authentication by allowing local users (local_enable=YES), ensuring that only authenticated users can access the FTP server.
- For networks with firewalls, Passive Mode might be preferable to avoid firewall issues. Configure passive mode settings in the configuration file.
- Define share permissions in the smb.conf file for shared folders. Control access by specifying users or groups that are allowed to read/write.
- Use Linux's file permissions (chmod and chown) to manage access at the filesystem level.
- Samba's built-in authentication can be used to control access for specific users or groups, using smbpasswd to manage passwords.
- Set permissions in the /etc/exports file to define which clients can access the NFS shares and whether they have read-only or read-write access.
- Use Unix/Linux permissions (chmod, chown) to control file-level access once NFS clients are connected.
- Use root squash (an option in /etc/exports) to prevent root users on client machines from having root privileges on the NFS server.
- Set user access controls in the FTP server configuration (vsftpd.conf). Control whether anonymous users or local users can access the server.
- Use local user directories and set read/write permissions to control what users can do with files.
- Use chroot jail to limit users' file access to their home directories (chroot_local_user=YES).
- Regularly back up shared files using tools like rsync, tar, or backup software such as Bacula or Amanda.
- Create a cron job to automate the backup of critical directories.
- Ensure backup files are stored both locally and on remote locations (e.g., cloud storage or offsite servers).
- Test recovery by restoring files from backups periodically to ensure that the backups are functional.



Application of learning 3.3.

You are tasked with setting up an SMB/CIFS file server in a Linux server environment for file sharing across the network. The file server will host shared folders for different departments: HR, Finance, and IT. Each department should have access only to its own folder, and permissions should be set accordingly

Indicative content 3.4: Configuration of Web Server

Duration: 7 hrs

Theoretical Activity 3.4.1: Description of web server

Tasks:

1: Answer the following questions:

i. What is a web server?
ii. How does a web server differ from an application server?
iii. What protocols do web servers use to communicate with clients?
iv. Configuration and Management
v. What are the common software options for web servers?
vi. How do you configure a virtual host in a web server?
vii. What is the purpose of the Document Root directive in a web server configuration?

2: Present the findings to trainer, your colleagues or to the whole class

3: Ask questions where necessary.

4: Read the key readings 3.4.1.

Key readings 3.4.1.: Description of web server

1. **Definition of a Web Server**

A web server is software or hardware that stores, processes, and delivers web pages to clients (browsers) over the internet using HTTP/HTTPS protocols. It serves static or dynamic content (HTML, CSS, JavaScript, etc.) requested by users.

2. **Types of Web Servers for Linux**:
   - **Apache HTTP Server**: The most popular open-source web server known for its flexibility and wide use. It supports dynamic content via modules.
   - **Nginx**: A lightweight, high-performance web server with a strong focus on concurrency and low resource usage. It excels at handling large numbers of simultaneous connections.
   - **Other Servers**: Lighttpd, Caddy, and OpenLiteSpeed also offer web serving capabilities for specific use cases.
3. **Configuration Files**: The configuration files for web servers define settings such as ports, document roots, server names, and security rules.
   - Apache: /etc/apache2/apache2.conf or /etc/httpd/httpd.conf
   - Nginx: /etc/nginx/nginx.conf or /etc/nginx/sites-available/default

- Ensure that settings like the root directory and virtual hosts are configured correctly.

4. **Installation of Apache HTTP Server**:
   - Use the package manager to install:
   - Ubuntu/Debian: sudo apt install apache2
   - CentOS/RHEL: sudo yum install httpd
   - Start the service and enable it on boot: sudo systemctl start apache2 (or httpd for CentOS).
5. **Installation of Nginx:**
   - Similarly, install Nginx via the package manager:
   - Ubuntu/Debian: sudo apt install nginx
   - CentOS/RHEL: sudo yum install nginx
   - Start and enable it on boot: sudo systemctl start nginx.
6. **Firewall Configuration:**
   - Open ports for HTTP (80) and HTTPS (443) to allow web traffic:
   - UFW (Ubuntu): sudo ufw allow 'Apache Full' or sudo ufw allow 'Nginx Full'
   - FirewallD (CentOS): sudo firewall-cmd --add-service=http --permanent and sudo firewall-cmd --add-service=https --permanent.
7. **Verify the Installation:**
   - Use a browser to visit http://your_server_ip and ensure the default page is displayed.
   - Verify the web server status:
   - Apache: sudo systemctl status apache2
   - Nginx: sudo systemctl status nginx
8. **Permissions:**
   - Ensure proper permissions are set for the web server's root directory (/var/www/html) to prevent unauthorized access:
   - Ownership: sudo chown -R www-data:www-data /var/www/html
   - Permissions: sudo chmod -R 755 /var/www/html
   - Only allow the web server user (e.g., www-data) write access to sensitive files.
9. **Encryption**:
   - Protect web server communication by using SSL/TLS encryption to secure data transferred between the server and clients.
   - Use strong ciphers and SSL/TLS protocols to minimize security vulnerabilities.
10. **SSL Certificate**:

- SSL Certificates are essential for securing HTTPS connections. Use **self-signed certificates** for internal testing or **Let's Encrypt** for free public certificates.



Practical Activity 3.4.2: Configuring Web Server



Task:

1: Individually, perform the following activity:
The Speratha Co. Ltd wants to install a web server and perform security. You have beentasked with install a web server (Apache or Nginx), configure it, and secure it using permissions, encryption, and SSL certificates. As a network administrator, you are required to install and configure it.

2: Read the key reading 3.4.2

3: Configure web server

4: Perform the task provided in application of learning 3.4



Key readings 3.4.2

Configuring Web Server

1. Install Web Server Software

- Choose and install software like **Apache** or **Nginx** on your server. **Image Idea**: A server dashboard with installation steps for Apache or Nginx.

2. Configure Server Settings

- Access the web server configuration file (e.g., /etc/apache2/apache2.conf for Apache).
- Set important parameters like the **port** (80 for HTTP, 443 for HTTPS). **Image Idea**: Editing configuration files with parameters like port and server name.

3. Set Up a Virtual Host

- Define a **virtual host** for each website/domain to host multiple sites on the same server.
- Specify domain name, document root, and log files. **Image Idea**: Virtual host configuration blocks with domain names and directory paths.

4. Document Root

- The **Document Root** specifies the directory from which the server serves website files (e.g., /var/www/html). **Image Idea**: Folder structure showing the document

root and website files.

5. Configure SSL (Optional)

- To enable **HTTPS**, configure **SSL certificates** and set the port to **443**.
- Use tools like **Let's Encrypt** for free SSL certificates. **Image Idea**: SSL certificate setup with padlock icons and "HTTPS" in the browser.

6. Restart Web Server

- After configuration, restart the web server to apply changes using commands like sudo systemctl restart apache2 or nginx. **Image Idea**: Command-line terminal with the restart command and confirmation.

7. Testing and Access

- Access the server through a browser by entering the domain or IP address. Ensure files are served properly. **Image Idea**: A browser displaying the homepage served by the configured web server.



Points to Remember

Description of web server

- A web server is software that serves web content (like HTML pages, images, and files) to clients (browsers) over the internet.
- A web server handles HTTP requests, while an application server runs business logic and processes dynamic content like APIs and databases.
- Web servers typically use the HTTP and HTTPS protocols to communicate with clients.
- Configuration and management involve setting up web server files, managing traffic, security settings, and optimizing performance.
- Common web server software includes Apache, Nginx, IIS, and LiteSpeed.
- To configure a virtual host, you specify the domain name, IP address, and document root in the web server's configuration file.
- The Document Root directive defines the folder where the web server looks for files to serve for a specific website or application.

Configuring Web Server

- A web server is software or hardware that stores, processes, and delivers web pages to clients (browsers) over the internet using HTTP/HTTPS protocols. It serves static or dynamic content (HTML, CSS, JavaScript, etc.) requested by users.

- There are different types of Web Servers for Linux such as Apache HTTP Server, Nginx, Lighttpd, Caddy, and OpenLiteSpeed also offer web serving capabilities for specific use cases.
- The configuration files for web servers define settings such as ports, document roots, server names, and security rules.
- Use the package manager to install Ubuntu/Debian: sudo apt install apache2, CentOS/RHEL: sudo yum install httpd
- Start the service and enable it on boot: sudo systemctl start apache2 (or httpd for CentOS).
- Open ports for HTTP (80) and HTTPS (443) to allow web traffic like UFW (Ubuntu) sudo ufw allow 'Apache Full' or sudo ufw allow 'Nginx Full' and FirewallD (CentOS): sudo firewall-cmd --add-service=http --permanent and sudo firewall-cmd --add-service=https --permanent.
- Use a browser to visit http://your_server_ip and ensure the default page is displayed.
- Ensure proper permissions are set for the web server's root directory (/var/www/html) to prevent unauthorized access
- Protect web server communication by using SSL/TLS encryption to secure data transferred between the server and clients.
- Use strong ciphers and SSL/TLS protocols to minimize security vulnerabilities.
- SSL Certificates are essential for securing HTTPS connections. Use self-signed certificates for internal testing or Let's Encrypt for free public certificates.
- Configure the web server to use SSL by enabling the SSL module and specifying the certificate paths in the configuration file.
- Verify the SSL certificate by accessing the web server via https://your_server_ip.



Application of learning 3.4.

The SP Co. Ltd wants to install a web server and perform security. You have been tasked with install a web server (Apache or Nginx), configure it, and secure it using permissions, encryption, and SSL certificates. As a network administrator, you are required to install and configure it.

Duration: 7 hrs

Theoretical Activity 3.5.1: Description of Mail server services

Tasks:

1: Answer the following questions:

i. What are the key components of an email server, and what roles do they play in the email delivery process?

ii. What log files are important for monitoring the email server, and what key metrics should you track for performance and troubleshooting?

iii. What methods can you use to test the functionality of the email server, and how can you optimize its performance for better efficiency?

iv. What strategies should be employed to ensure regular backups of the email server, and what steps are necessary to recover from a disaster?

2: Present your findings to the trainer or the whole class

3: Address any questions or concerns.

5: Read the Key readings 3.5.1 in their manuals

Key readings 3.5.1: Description of Mail server services

- **SMTP (Simple Mail Transfer Protocol)**: Protocol used for sending emails from client to server and between servers.
- **IMAP (Internet Message Access Protocol)**: Allows users to access and manage their emails directly on the server, supporting multiple device synchronization.
- **POP3 (Post Office Protocol)**: Downloads emails from the server to the local client, often removing them from the server.
- **Mail Transfer Agent (MTA)**: Software responsible for routing and delivering email (e.g., Postfix).
- **Mail Delivery Agent (MDA)**: Delivers email to user mailboxes (e.g., Dovecot).
- **Mail User Agent (MUA)**: Client software used by users to send and receive emails (e.g., Outlook, Thunderbird).
- **Set Up of SMTP Server (Postfix)**
  - ✓ **Installation**: Use package manager commands (apt or yum) to install Postfix.
  - ✓ **Configuration**: Set essential parameters in /etc/postfix/main.cf, including

myhostname, mydomain, and mydestination.

- ✓ **Service Management**: Start and enable Postfix using systemctl.

- **Set Up Mailboxes and Users**
  - ✓ **User Creation**: Add users using adduser to create corresponding mailboxes.
  - ✓ **Mailbox Location**: Configure mailbox storage in Postfix settings to define where emails are stored.
  - ✓ **Testing**: Send test emails to ensure proper mailbox functionality.
- **Monitor and Logging**
  - ✓ **Log Files**: Regularly check /var/log/mail.log or /var/log/maillog for email traffic and errors.
  - ✓ **Monitoring Tools**: Use tools like Logwatch for email activity reporting and spam detection.
- **Testing and Optimization**
  - ✓ **Email Functionality Testing**: Use command-line tools (e.g., mail, telnet) to test sending and receiving emails.
  - ✓ **Optimization Settings**: Adjust Postfix parameters for performance, such as default_process_limit and maximal_queue_lifetime.
- **Implement Backup and Disaster Recovery Procedures**
  - ✓ **Regular Backups**: Schedule backups of email data and configuration files using tools like rsync or tar.
  - ✓ **Disaster Recovery**: Document and regularly test the restoration process to ensure data integrity and availability.



Practical Activity 3.5.2: Configuring mail server services



Task:

1: Individually, perform the following activity:

You are required to go to computer lab to configure a mail server for your school to facilitate both trainers and trainees for communicating between them.

2: Read the key reading 3.5.2.

3: configure mail server

4: Perform the task provided in application of learning 3.5



Key readings 3.5.2:Configuring mail server services

1. Choose the Right Mail Server Software

- **Postfix**, **Send mail**, and **Exim** are popular SMTP server options.
- For incoming mail, consider **Dovecot** or **Courier** for IMAP/POP3 services.

- Use **DNS** records (MX records) to route email correctly.

2. Installation and Configuration

- **Postfix Setup:** Install Postfix for outgoing emails (SMTP) and configure it to handle email for your domain.
  - ✓ Configure /etc/postfix/main.cf for general settings like domain, hostname, and networks.
  - ✓ Set up **virtual mailboxes** if supporting multiple domains.
- **Dovecot Setup:** Install Dovecot for IMAP/POP3 to allow users to retrieve emails.
  - ✓ Configure /etc/dovecot/dovecot.conf to define mail location and user settings.

3. DNS Configuration

- Set up **MX Records** in your DNS to point to the mail server's IP address.
- Configure **SPF (Sender Policy Framework)**, **DKIM (DomainKeys Identified Mail)**, and **DMARC (Domain-based Message Authentication, Reporting & Conformance)** records to improve email deliverability and security.
  - ✓ SPF defines which mail servers are allowed to send emails for your domain.
  - ✓ DKIM helps ensure that email content hasn't been tampered with.
  - ✓ DMARC provides a way to tell receiving mail servers how to handle messages that fail SPF or DKIM checks.

4. Security Considerations

- **TLS/SSL Encryption:** Ensure all email transmissions are encrypted using TLS (Transport Layer Security). This can be done by configuring Postfix to use SSL/TLS for SMTP and Dovecot for IMAP/POP3.
  - ✓ Certificates can be generated using **Let's Encrypt** or any other CA (Certificate Authority).
- **SASL Authentication:** Enable SASL (Simple Authentication and Security Layer) to require user authentication for sending emails, preventing open relay abuse.
- **Firewall Settings:** Ensure that ports like **25 (SMTP)**, **587 (SMTP submission)**, **993 (IMAP)**, and **995 (POP3)** are open only for relevant services.

5. User and Mailbox Management

- Create mail users and set up mailboxes. Users can be system accounts or virtual users, depending on your setup.
- For virtual users, configure Postfix to use a virtual mailbox system, often stored in /var/mail/ or /home/mailboxes/.

6. Spam Filtering and Antivirus

- Set up **SpamAssassin** or **Amavis** to filter spam and junk mail.
- Install an antivirus tool like **ClamAV** to scan emails for malware and other malicious attachments.

7. Monitoring and Logging

- Regularly monitor mail logs located in /var/log/maillog or /var/log/mail.log for

any issues with email delivery or performance.
- Set up monitoring tools to track email queue length and system performance.

8. Testing the Mail Server

- Use tools like **telnet** or **swaks** to test sending and receiving emails from the server.
- Check that emails are delivered correctly and can be retrieved by users.
- Verify that encryption is correctly applied during email transmission.

9. Backup and Recovery

- Ensure regular backups of the mail server configuration files, user mailboxes, and important logs.
- Implement a disaster recovery plan that outlines how to restore mail services in case of failure.

10. User Documentation and Training

- Provide users with the correct settings to configure their email clients (e.g., IMAP/POP3 settings, SMTP server, encryption).
- Educate users about strong password usage and how to recognize phishing emails.



Points to Remember

Description of Mail Server service

- The key components of an email server include SMTP for sending emails, IMAP/POP3 for receiving, and a mail transfer agent (MTA) like Postfix to manage the delivery process.
- To install and configure Postfix as an SMTP server on Linux, use package managers like apt or yum and edit the configuration file (/etc/postfix/main.cf) to set up domains and relay policies.
- Creating user accounts involves using Linux commands to add users and configuring their mailboxes in the mail server using tools like Dovecot for IMAP/POP3 access.
- Important log files for monitoring the email server include /var/log/mail.log, where you track delivery failures, bounce rates, and server load.
- To test email server functionality, you can use commands like telnet or mail clients, and optimize performance by managing queues and enabling anti-spam measures.
- Regular backups of the email server can be automated using tools like rsync or backup scripts, and disaster recovery involves restoring configurations, mailboxes, and ensuring DNS settings are correct.

While configuring mail server:

- choose mail server to install and use DNS records (MX records) to route email correctly.
- Install Postfix for outgoing emails (SMTP) and configure it to handle email for your domain.
- Install Dovecot for IMAP/POP3 to allow users to retrieve emails.
- Configure /etc/dovecot/dovecot.conf to define mail location and user settings.
- Set up MX Records in your DNS to point to the mail server's IP address.
- Configure SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail), and DMARC (Domain-based Message Authentication, Reporting & Conformance) records to improve email deliverability and security.
- Ensure all email transmissions are encrypted using TLS (Transport Layer Security). This can be done by configuring Postfix to use SSL/TLS for SMTP and Dovecot for IMAP/POP3.
- Certificates can be generated using Let's Encrypt or any other CA (Certificate Authority).
- Enable SASL (Simple Authentication and Security Layer) to require user authentication for sending emails, preventing open relay abuse.
- Ensure that ports like 25 (SMTP), 587 (SMTP submission), 993 (IMAP), and 995 (POP3) are open only for relevant services.
- Create mail users and set up mailboxes. Users can be system accounts or virtual users, depending on your setup.
- For virtual users, configure Postfix to use a virtual mailbox system, often stored in /var/mail/ or /home/mailboxes/.
- Set up SpamAssassin or Amavis to filter spam and junk mail.
- Install an antivirus tool like ClamAV to scan emails for malware and other malicious attachments.
- Regularly monitor mail logs located in /var/log/maillog or /var/log/mail.log for any issues with email delivery or performance.
- Set up monitoring tools to track email queue length and system performance.
- Use tools like telnet or swaks to test sending and receiving emails from the server.
- Check that emails are delivered correctly and can be retrieved by users.
- Verify that encryption is correctly applied during email transmission.



Application of learning 3.5.

JALI Innovations Company is a growing technology company that requires a robust email communication system to facilitate internal and external communication among

its employees and clients. The company has decided to set up an email server using Postfix to manage its email services effectively. You are tasked by The IT department to configure the mail server, ensure security, and implement a backup recovery plan.



Learning outcome 3 end assessment

Written assessment

1. Which of the following command is used to start a service in Linux?

a) systemctl stop
b) service start
c) systemctl start
d) service status

2. Which configuration file contains the list of services to be managed by systeme?

a) /etc/services
b) /etc/systemd/system.conf
c) /etc/sysconfig/services
d) /etc/init.d/services.conf

3. What is the purpose of the systemctl enable command?

a) To immediately start a service
b) To stop a running service
c) To allow a service to start automatically at boot time
d) To restart a service after it fails

4. Which of the following commands will show the status of a service in Linux?

a) systemctl status
b) service --status-all
c) systemctl show
d) service info

5. If a Linux service is running but not responding correctly, which command can you use to restart it?

a) systemctl reload
b) systemctl restart
c) systemctl disable
d) systemctl stop

6. What is the function of the journal ctl command in managing services?

a) To display real-time service status
b) To modify system service files
c) To view logs for system services managed by systemd
d) To restart all services

7. Which command can be used to stop a service from running?

a) systemctl kill
b) service quit

c) systemctl stop
d) stop -service

8. How do you disable a service from starting at boot using systemctl?

a) systemctl shutdown
b) systemctl disable
c) systemctl block
d) systemctl pause

9. To view active and inactive services managed by systemd, which command should you use?

a) systemctl list-units --type=service
b) service --list-active
c) systemctl list-services
d) ls /etc/systemd/services

10. Which command can be used to reload a service configuration without stopping it?

a) systemctl restart
b) systemctl reload
c) systemctl enable
d) systemctl reset

11. When using the cron service, which file is edited to schedule automated tasks?

a) /etc/crontab
b) /etc/cron.conf
c) /etc/systemd/cron.d
d) /var/log/cronjobs

12. Which service is typically responsible for managing scheduled tasks in Linux?

a) atd
b) cron
c) httpd
d) ftp

13. In order to list all failed services managed by systemd, you can use which command?

a) systemctl list-units --state=failed
b) systemctl failed-services
c) service --list-failed
d) systemctl show --failed

14. Which of the following commands will display the logs related to a specific service (e.g., httpd)?

a) journalctl -u httpd
b) service httpd logs
c) systemctl log httpd
d) httpd showlog

15. If a service is disabled, what must be done to make it active and start at boot again?

a) Run systemctl start
b) Run systemctl reload
c) Run systemctl enable
d) Run systemctl shutdown

16. Match with related answer to the following table:

Answer	**COLUMN A**	**COLUMN B**
1…….	1. systemctl start	A. Stop a service start
2…….	2. systemctl enable	B. Check the status of a service
3…….	3. systemctl stop	C. Disable a service from starting at boot
4…….	4. systemctl status	D. Restart a service
5…….	5. systemctl restart	E. Start a service
6…….	6. /etc/systemd/system/	F. Configuration file location for services
7…….	7. journalctl	G. View service logs
8…….	8. crontab	H. Service manager for scheduling jobs
9…….	9. systemctl disable	I. Enable a service to start at boot
10……..	10. ps aux	J. Command to list running processes

Practical assessment

You are the system administrator for a growing organization that requires robust network infrastructure to support its expanding operations. The organization needs to set up and configure several critical services on a Linux server to ensure efficient communication, data management, and resource accessibility. These services include DNS, DHCP, a file server, a web server, and a mail server. Each service must be configured with security, reliability, and scalability in mind. Your task involves setting up, configuring, and testing these services while ensuring they meet the organization's operational requirements.



References

DigitalOcean. (n.d.). *How to configure a DNS server on Ubuntu 20.04*. Retrieved from www.digitalocean.com/community/tutorials/how-to-configure-a-dns-server-on-ubuntu-20-04

DigitalOcean. (n.d.). *How to set up Postfix as a Sendmail replacement*. Retrieved from www.digitalocean.com/community/tutorials/how-to-set-up-postfix-as-a-sendmail-replacement

Moffett, K. D., & Salus, P. H. (2008). *Postfix: The definitive guide*. O'Reilly Media.

Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *Linux administration handbook* (6th ed.). Prentice Hall.

Network World. (n.d.). *Understanding DNS and its functions*. Retrieved from www.networkworld.com/article/3239637/understanding-dns-and-its-functions.html

Sharma, C. (2005). *The book of Postfix: State-of-the-art message transport*. No Starch Press.

Shotts, W. E., Jr. (2012). *The Linux command line: A complete introduction*. No Starch Press.

Vohra, R. (2022). *DNS for dummies*. Wiley.

Ward, B. (2021). *How Linux works: What every superuser should know* (3rd ed.). No Starch Press.

RTB | RWANDA TVET BOARD

October 2024