DEPARTMENT:ICT

TRADE:SOFTWARE DEVELOPMENT

LEVEL:5

MODULE TYPE:SPECIFIC

MODULE NAME:DEVOPS APPLICATION

LEARNING UNIT 1: **PERFORM SERVER CONFIGURATION**

Definitions of key Terms

- ✓ Server:
  A server is a computer or system that provides resources, data, services, or programs to other computers, known as clients, over a network.

- ✓ Linux:
  Just like Windows, iOS, and Mac OS, Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn't function.

- ✓ Development Operations(DevOps) :
  DevOps combines development (Dev) and operations (Ops) to increase the efficiency, speed, and security of software development and delivery compared to traditional processes. A more nimble software development lifecycle results in a competitive advantage for businesses and their customers.

- ✓ DevSecOps:
  DevSecOps which stands for development, security, and operations, is a framework that integrates security into all phases of the software development lifecycle. Organizations adopt this approach to reduce the risk of releasing code with security vulnerabilities.

- ✓ Container :
  Containers are a form of operating system virtualization. A single container might be used to run anything from a small microservice or software process to a larger application. Inside a container are all the necessary executables, binary code, libraries, and configuration files.

- ✓ Node
  A node is essentially a physical or virtual machine that hosts multiple pods, providing the necessary resources for running the applications. Nodes play a crucial role in enabling DevOps practices within Kubernetes.

- ✓ Infrastructure as code (IaC):
  Infrastructure as code (IaC) is used for infrastructure automation to create environments. The most common use of IaC is in software development to build, test, and deploy applications. Traditionally, system administrators used a combination of scripts and manual processes to set up infrastructure environments.

- ✓ Infrastructure as a Service:
  IaaS, or Infrastructure as a Service, is a cloud computing model that provides on-demand access to computing resources such as servers, storage, networking, and virtualization.

  IaaS is attractive because acquiring computing resources to run applications or store data the traditional way requires time and capital. Organizations must purchase equipment through procurement processes that can take months. They must invest in physical spaces, typically specialized rooms with power and cooling. And after deploying the systems, they need IT professionals to manage and maintain them

- ✓ CI/CD:

  which stands for continuous integration and continuous delivery/deployment, aims to streamline and accelerate the software development lifecycle.

  Continuous integration (CI) refers to the practice of automatically and frequently integrating code changes into a shared source code repository. Continuous delivery and/or deployment (CD) is a 2 part process that refers to the integration, testing, and delivery of code changes. Continuous delivery stops short of automatic production deployment, while continuous deployment automatically releases the updates into the production environment

Identification of Linux distributions

A Linux distribution, commonly known as a Linux distro, refers to a complete operating system that is based on the Linux kernel and bundled with various software packages. In English, a Linux distro can be explained as follows:

A Linux distribution is a complete operating system that includes the Linux kernel, system libraries, utilities, application software, and a package management system. It is created by assembling various software components from different sources and packaging them together to provide a cohesive and user-friendly computing environment.

Some popular Linux distributions include:

- **Ubuntu**: Known for its user-friendliness and extensive community support.

- **Debian:** A stable distribution that's often used as a base for other distros.

- **Fedora**: Features the latest software and technologies, often serving as a testing ground for Red Hat.

- **Arch Linux:** A rolling release distribution known for its simplicity and customization.

- **CentOS/RHEL**: Focused on enterprise users with a stable and secure environment.

- **openSUSE:** Offers flexibility and choice with both rolling and fixed releases.

Installation of Linux operating system

Installing a Linux operating system can vary slightly depending on the distribution you choose, but here's a general guide that covers the basic steps for a popular distribution like Ubuntu.

- ✓ **Prerequisites**

1. Choose a Distribution: Common options include Ubuntu, Fedora, Mint, and Debian.

2. Download ISO File: Go to the distribution's website and download the ISO file.

3. Create Bootable Media: Use tools like Rufus (Windows), balenaEtcher (Mac/Linux), or the `dd` command (Linux) to create a bootable USB drive.

Steps to Install Linux

1. Back Up Your Data: Before making any changes to your system, back up important files.

2. Boot from USB:

- Insert the USB drive into your computer.

- Restart your computer and enter the BIOS/UEFI settings (often by pressing `F2`, `F10`, `DEL`, or `ESC` during boot).

- Set the boot order to prioritize the USB drive.

3. **Start Installation:**

- Save the BIOS settings and exit. The system should boot from the USB drive.

- You'll see a welcome screen. Select "Try Ubuntu" to test it out, or "Install Ubuntu" to start the installation process.

4. **Select Language:** Choose your preferred language and click "Continue."

5.Prepare Installation:

- Updates and Other Software: You can choose to install updates and third-party software during the installation.

- Installation Type: Decide whether to install alongside an existing OS, erase the disk, or do a custom partition.

6. Partitioning (if applicable):

- If you choose "Something else," you can manually create partitions. A typical setup includes:

  - Root (/): Minimum of 20 GB.

  - Swap: Usually equal to RAM size or a bit more (if you plan to use hibernation).

  - Home (/home): For personal files, space depending on your needs.

7. Set Your Time Zone: Choose your location to set the correct time zone.

8. Create User Account:

- Enter your name, computer name, username, and password.

9. Installation: Click "Install Now." Review the changes, and confirm.

10. Finish Installation: Once installation completes, you'll be prompted to restart the computer. Remove the USB drive when prompted.

11.Post-Installation:

- Boot into your new Linux system.

- Update the system using the terminal:

  - ✓ sudo apt update
  - ✓ sudo apt upgrade

12. Install Additional Software: Use the Software Center or the terminal to install any additional applications you need.

Applying Linux basics commands

summarizing basic Linux commands across various categories, along with their usage and examples:

Category	Command	Description	Example
System Information	uname	Displays system information	uname -a
	top	Displays real-time system processes and resource usage	top
	htop	Enhanced version of top	htop (may require installation)
	free	Shows memory usage	Free -h
	df	Displays disk space usage	df -h
	Lsb_release	Displays Linux distribution information	Lsb_release -a
File and Directory Management	ls	Lists files in a directory	Ls -l
	cd	Changes the current directory	Cd /path/to/directory
	mkdir	Creates a new directory	mkdir new_folder
	rmdir	Removes an empty directory	Rmdir empty_folder
	cp	Copies files or directories	Cp source.text destination.txt
	rm	Removes files or directories	Rm file.text

	mv	Moves or renames files or directories	Mv oldname.txt newname.txt
Text Processing	cat	Concatenates and displays file content	Cat file.txt
	less	Views file content page by page	Less file.txt
	grep	Searches for specific patterns within files	Grep “search_term” file.txt
	awk	Powerful text processing tool	Awk ‘{print $1}’ file.txt
	sed	Stream editor for filtering and transforming text	Sed ‘s/old/new/g’ file.txt
Process Management	ps	Displays current running processes	Ps aux
	kill	Terminates a process by PID	Kill 1234
	pkill	Terminates processes by name	Pkill process_name
	jobs	Lists active jobs in the current shell session	jobs
	bg	Sends a job to the background	Bg %1
	fg	Brings a job to the foreground	Fg %1
Package Management	Apt update	Updates the package list	Sudo apt update
	Apt upgrade	Upgrades installed packages	Sudo apt upgrade
	Apt install	Installs a new package	Sudo apt install package_name
	Apt remove	Removes a package	Sudo apt remove package_name
User and Group Management	Adduser	Creates a new user	sudo adduser new_username
	usermod	Modifies an existing user account	sudo usermod -aG group_name username
	deluser	Deletes a user account	sudo deluser username
	groupadd	Creates a new group	sudo groupadd new_group

	passwd	Changes a user's password	sudo passwd username
System Control	shutdown	Shuts down the system	sudo shutdown now
	reboot	Reboots the system	sudo reboot
	systemctl	Manages system services (for systems using systemd)	sudo systemctl start service_name
	service	Manages services on older systems	sudo service service_name restart

Management of server services:

What is Server Management? Server management involves all the monitoring and maintenance required for servers to operate reliably and at optimal performance levels. This includes managing hardware, software, security, and backups to keep the IT environment operational and efficient.

Description of server services :

Service	Description
Web	Web servers host websites and serve web pages to clients over HTTP/HTTPS. Common software includes Apache, Nginx, and Microsoft IIS. They handle requests from browsers and deliver content (HTML, CSS, JS, images). Features may include SSL/TLS support, virtual hosting, and URL rewriting.
Mail	Mail servers manage the sending, receiving, and storage of email. They use protocols like SMTP (for sending), IMAP, and POP3 (for receiving). Examples include Postfix, Exim, and Dovecot. Mail servers often support spam filtering, mailing lists, and user authentication.
File	File servers provide centralized storage for files, allowing users to access and share files over a network. Common protocols include SMB (used in Windows environments), NFS (for Unix/Linux), and FTP/SFTP for file transfers. File servers enable file versioning, permissions management, and redundancy.

SSH	Secure Shell (SSH) is a protocol used for secure remote administration of servers. It encrypts data transmitted between the client and server, providing secure command-line access. SSH is widely used for secure file transfers (SCP, SFTP) and tunneling.
Network	Network services manage and route data traffic across networks. This includes DHCP (dynamic IP address assignment), DNS (domain name resolution), and VPN (secure remote access). Networking services ensure connectivity, security, and proper routing of data packets.
DNS	Domain Name System (DNS) translates human-readable domain names into IP addresses. It allows users to access websites using easy-to-remember names instead of numerical IP addresses. DNS servers maintain records (A, CNAME, MX, etc.) to route requests accurately.
PROXY	Proxy servers act as intermediaries between clients and other servers. They can provide anonymity, content filtering, caching, and load balancing. Proxies can enhance security by hiding client IP addresses and can be used for monitoring internet traffic and enforcing corporate policies.
Monitoring and Logging	These services collect and analyze data about server performance and user activity. Monitoring tools (like Nagios, Zabbix) track metrics (CPU, memory, disk usage) and alert administrators to issues. Logging services (like syslog, ELK stack) capture and store logs for analysis, auditing, and troubleshooting.
Backup	Backup services ensure data integrity by creating copies of data to prevent loss from failures, corruption, or disasters. This can involve local storage, cloud solutions, or off-site backups. Common tools include rsync, Bacula, and various cloud-based solutions. Backups can be scheduled for regular intervals and may support incremental or full backups.

KEY POINTS TO CONCIDER:

- **Web Servers**: Enable online content delivery and website hosting.
- **Mail Servers**: Facilitate email communication and storage.
- **File Servers**: Centralize file storage for easy access and sharing.
- **SSH**: Provides secure remote access for administration.
- **Network Services**: Manage data routing and connectivity.
- **DNS**: Converts domain names to IP addresses for website access.
- **Proxy Servers**: Enhance security and control over internet traffic.
- **Monitoring and Logging**: Help maintain server health and troubleshoot issues.
- **Backup Services**: Protect data through regular backups and recovery options.

Configure server services:

Server configuration is the process of setting up the hardware and software components of a server to meet the specific needs and preferences of an organization or a user. It involves choosing the right server type, operating system, network settings, security measures, and performance optimization strategies.

configuration of various server services along with installation steps, main configuration files, and restart commands:

service	Installation Command	Main Configuration File	Restart Command
Web Server (Apache)	sudo apt update && sudo apt install apache2	/etc/apache2/apache2.conf	sudo systemctl reload apache2
Mail Server (Postfix)	sudo apt update && sudo apt install postfix	/etc/postfix/main.cf	sudo systemctl restart postfix
File Server (Samba)	sudo apt update && sudo apt install samba	/etc/samba/smb.conf	sudo systemctl restart smbd
SSH Server	sudo apt update && sudo apt install openssh-server	/etc/ssh/sshd_config	sudo systemctl restart ssh
DNS Server (BIND)	sudo apt update && sudo apt install bind9	/etc/bind/named.conf.options	sudo systemctl restart bind9
Proxy Server (Squid)	sudo apt update && sudo apt install squid	/etc/squid/squid.conf	sudo systemctl restart squid
Monitoring (Nagios)	Installation varies (see specific guide)	Varies based on configuration	sudo systemctl restart nagios (if applicable)

Best Practices

- **Backup Configuration Files**: Always back up existing configuration files before making changes.
- **Use Firewalls**: Ensure proper firewall rules to protect services.
- **Regular Updates**: Keep server software up to date to mitigate vulnerabilities.
- **Secure Access**: Use strong passwords and consider key-based authentication for SSH.
- **Monitoring**: Implement monitoring solutions to track performance and uptime.
- **Documentation**: Maintain records of configurations and changes for future reference.