****

**DevOps APPLICATION**

**Apply DevOps Techniques**

**RQF Level: 5 Learning Hours 60**

**Credits: 6**

**Sector: ICT and MULTIMEDIA**

**Trade: SOFTWARE DEVELOPMENT**

**Module Type: Specific**

**Curriculum: ICTSWD5001 TVET CERTIFICATE V IN**

**SOFTWARE DEVELOPMENT**

**Copyright:** **@Rwanda TVET Board, 2024**

**Issue Date: February 2024**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Purpose statement** | This module describes the skills, knowledge and attitude required to apply DevOps techniques. It is prepared for students pursuing TVET Level 5 in Software Development. At the end of this module the student will be able to Perform server configuration, Deploy the system and Implement monitoring strategies. | | | | | |
| **Learning assumed to be in place** | Develop front-end using React Js, Develop Mobile Application using Flutter | | | | | |
| **Delivery modality** | **Training delivery** | | **100%** | **Assessment** | | **Total 100%** |
| Theoretical content | | 30% | Formative assessment | 30% | 50% |
| Practical work: | | 70% | 70% |
| Group project and presentation | 20% |
| Individual project /Work | 50% |
|  | | Summative Assessment | | | 50% |

**Elements of Competence and Performance Criteria**

|  |  |
| --- | --- |
| **Elements of competence** | **Performance criteria** |
| 1. **Perform server configuration** | 1.1. Environment is properly prepared based on platform. |
| 1.2. Linux basics are properly applied based on Linux distribution. |
| 1.3. Server services are properly managed based on selected platform. |
| 1. **Deploy the system** | 2.1. Deployment environment is properly prepared based on the system to be deployed. |
| 2.2 Continuous delivery is properly used according to the identified area of automation. |
| 2.3 Container is properly configured based on scalability of applications. |
| 2.4 Migration is properly performed based on the system environment. |
| 1. **Implement monitoring strategies** | 3.1. Monitoring tools are proactively prepared according to their respective documentation. |
| 3.2. Performance metrics and feedback data are routinely analysed in accordance with system requirements. |
| 3.3 Monitoring report is properly documented based on analysed metrics. |

|  |  |  |
| --- | --- | --- |
| **Knowledge** | **Skills** | **Attitude** |
| * Identification of Linux distributions * Description of server services * Description of DevOPs principles * Identification of technologies used in system * Description significance of Data Analysis | * Installation of Linux operating system * Applying Linux basics commands * Management of server services * Configuration of server services * Installation of system dependencies * Configuration of CI pipeline * Configuration of CD pipeline * Implementation of Continuous Integration * Performing migration * Configuration of monitoring tools * Analyzing Data in DevOps * Utilizing Monitoring Tools | * Team work * Being critical thinker * Being Innovative * Being creative * Practical oriented * Have Detail oriented * Being honesty * Have Passion for Learning * Problem-Solving Mindset * Collaboration and Communication * Have Attention to Security * Have Ethical Coding |

**Course content**

|  |  |
| --- | --- |
| **Learning outcomes** | **At the end of the module the learner will be able to:**   1. Perform server configuration 2. Deploy the system 3. Implement monitoring strategies |

**Learning Outcome 1: Perform server configuration**

**1.1: Preparation of environment**

**1.1.1: Definitions of key Terms**

1. **A server** is a computer or system that provides resources, data, services, or programs to other computers, known as clients, over a network.
2. **Linux** is an open-source operating system based on Unix. It was first introduced in 1991 by Linus Torvalds.

* It is known for its stability, security, and versatility, making it a popular choice for a wide range of applications, from personal computers to servers and supercomputers.

1. **Development Operations (DevOps):** is a software engineering methodology which aims to integrate the work of development teams and operations teams by facilitating a culture of collaboration and share responsibility.

* It allows single team to handle the entire application lifecycle from development to testing, deployment, and operations.
* It helps you to reduce the disconnection between software developer, quality assurance (QA) engineers, and system administrators.
* It is nothing but a practice or methodology of making developers and operators folks work together.

1. **DevSecOps:** is a combination of Development, Security, and Operations,

**.** it is an approach that integrates security practices within the DevOps process.

**.** It emphasizes collaboration between development, operations, and security teams to ensure software applications are built securely, vulnerabilities are mitigated, and risks are minimized throughout the software development lifecycle.

DevSecOps promotes the **“shift left”** concept, which means incorporating security measures at the earliest stages of development, rather than addressing them as an afterthought.

By automating security processes and creating a culture of shared responsibility,

DevSecOps enhances a company’s overall security posture, reduces the time to market for applications, and fosters a proactive response to security threats.

1. **A container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.
2. **Node** is Runtime environment for executing Js code, Outside of Browser. it is built upon Chrome v8 engine using c++.
3. **Infrastructure as code (IaC)** is the ability to provision and support your computing infrastructure using code instead of manual processes and settings.

. It is used for infrastructure automation to create environments.

. The most common use of IaC is in software development to build, test, and deploy applications.

. It is a process for managing and operating data servers, storage systems, system configurations, and network infrastructure.

. Examples of IaC include Puppet, Terraform, and Chef.

1. **Infrastructure as a service (IaaS)** is a business model that delivers IT infrastructure like compute, storage, and network resources on a pay-as-you-go basis over the internet.
2. **CI/CD**: stand for **continuous integration** and **continuous delivery / continuous deployment**.

. is a methodology that allows developers to automate the building, testing, and deploying of their applications.

**Benefits of DevSecOps for organizations**

Organizations that adopt DevSecOps typically see benefits that include:

* Reduced breach risk
* Preventing secret leaks
* Improved compliance
* Greater confidence in supply chain security
* Tracking progress and effectiveness
* Accelerated delivery to customers

**DevSecOps vs DevOps?**

**DevOps** has transformed how many organizations build and ship software. One aspect of the software development lifecycle (SDLC) was left outside the DevOps model is security.

**DevSecOps** seeks to correct that by integrating security into the SDLC in the same way that DevOps prioritizes quality, speed, and deep collaboration across all stages of development.

**What are the benefits of infrastructure as code?**

* + - 1. Easily duplicate an environment
      2. Reduce configuration errors
      3. Iterate on best-practice environments

**What is the role of IaC in DevOps?**

DevOps is the process of improving collaboration between software development and IT operations teams.

A key goal of DevOps is to automate infrastructure tasks across the development process.

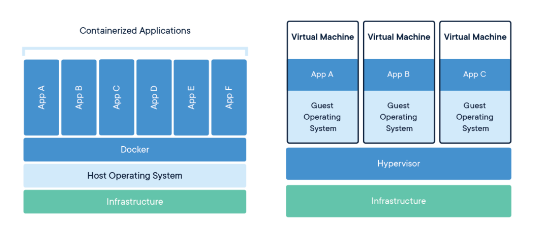
DevOps teams use infrastructure as code for many purposes:

* Quickly set up complete environments, from development to production
* Help ensure consistently reproducible configurations between environments
* Integrate seamlessly with cloud providers
* efficiently scale infrastructure resources up or down based on demand

**Comparing Containers and Virtual Machines**

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient.

|  |  |
| --- | --- |
| **CONTAINERS** | **VIRTUAL MACHINES** |
| Containers are an abstraction at the app layer that packages code and dependencies together. | Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers |
| Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. | The hypervisor allows multiple VMs to run on a single machine. |
| Containers take up less space than VMs (container images are typically tens of MBs in size) | Each VM includes a full copy of an operating system, the application, necessary binaries and libraries – taking up tens of GBs. |
| Container can handle more applications and require fewer VMs and Operating systems. | VMs can also be slow to boot. |



**1.1.2: Identification of Linux distributions**

**What is a Linux Kernel?**

The **kernel** is the central component of an operating system that manages the computer and its hardware operations. It handles memory operations and CPU time.

**What is a Linux distribution?**

A **Linux distribution** or **distro** is a version of the Linux operating system that includes the Linux kernel, system utilities, and other software.

**What does it mean that a distribution is derived?**

When you say that a distribution is "**derived**" from another, the newer distro is built upon the base or foundation of the original distro.

Today, there are thousands of Linux distributions to choose from, offering differing goals and criteria for selecting and supporting the software provided by their distribution.

Distributions vary from one to the other, but they generally have several common characteristics which are:

* A distribution consists of a Linux kernel.
* It supports user space programs.
* A distribution may be small and single-purpose or include thousands of open-source programs.
* Some means of installing and updating the distribution and its components should be provided.

Some popular Linux distributions are:

* **Ubuntu:** One of the most widely used and popular Linux distributions. It is user-friendly and recommended for beginners.
* **Linux Mint:** Based on Ubuntu, Linux Mint provides a user-friendly experience with a focus on multimedia support.
* **Arch Linux:** Popular among experienced users, Arch is a lightweight and flexible distribution aimed at users who prefer a DIY approach.
* **Manjaro:** Based on Arch Linux, Manjaro provides a user-friendly experience with pre-installed software and easy system management tools.
* **Kali Linux:** Kali Linux provides a comprehensive suite of security tools and is mostly focused on cybersecurity and hacking.
* **CentOS:** has a longer upgrade cycle, lasting approximately five years.
* **Fedora:** Fedora is also known for being a stable and dependable operating system. Fedora has a lot of graphical tools and useful software for office work, virus protection, system management, playing media, learning, and other things.
* **Debian:**Debian is a stable Linux distribution, but it also lets early adopters try out packages that aren't quite ready yet. People think it's a great distribution for running servers because its software is stable and it comes out less often.
* **Red Hat Enterprise Linux:** Red Hat Enterprise Linux (**RHEL**), which calls itself the **"world's leading enterprise Linux platform,"** is used for a lot of different things, such as making software and running hardware. RHEL is well-known for its stability and speed.

**What are the Basic Features of Linux?**

* **Portable** means that software can run on different types of hardware in the same way.
* **Open-Source** means that its source code is freely available, and it is a community-based development project.
* **Linux is a multiuser system**, which means that multiple users can access system resources such as memory/application programs at the same time.
* **Linux is a multiprogramming system**, which means that multiple applications can run at the same time.
* The Hierarchical File System (HLFS) provides a standard file structure for arranging system and user files.
* Linux Shell includes a special interpreter program that can be used to execute operating system commands.
* Linux provides user security through authentication features such as password protection, controlled access to specific files, and data encryption.

**1.1.3:  Install and access of Linux operating system**

**Install Linux as the primary OS**

Installing Linux as the primary OS is the most efficient way to use Linux, as you can use the full power of your machine.

In this section, you will learn how to install Ubuntu, which is one of the most popular Linux distributions.

* **Step 1 – Download the Ubuntu iso:** Go to the official website of Ubuntu <https://ubuntu.com/download/desktop> and download the iso file. Make sure to select a stable release that is labeled "LTS". LTS stands for Long Term Support which means you can get free security and maintenance updates for a long time (usually 5 years).
* **Step 2 – Create a bootable pendrive:** There are a number of softwares that can create a bootable pendrive. I recommend using Rufus, as it is quite easy to use. You can download it from rufus website which is <https://rufus.ie/en/#google_vignette> .
* **Step 3 – Boot from the pendrive**: Once your bootable pendrive is ready, insert it and boot from the pendrive. The boot menu depends on your laptop. You can google the boot menu for your laptop model.
* **Step 4 – Follow the prompts**. Once, the boot process starts, select try or install ubuntu.



The process will take some time. Once the GUI appears, you can select the language, and keyboard layout and continue. Enter your login and name. Remember the credentials as you will need them to log in to your system and access full privileges. Wait for the installation to complete.

* **Step 5 – Restart**: Click on restart now and remove the pen drive.
* **Step 6 – Login**: Login with the credentials you entered earlier.

And there you go! Now you can install apps and customize your desktop.



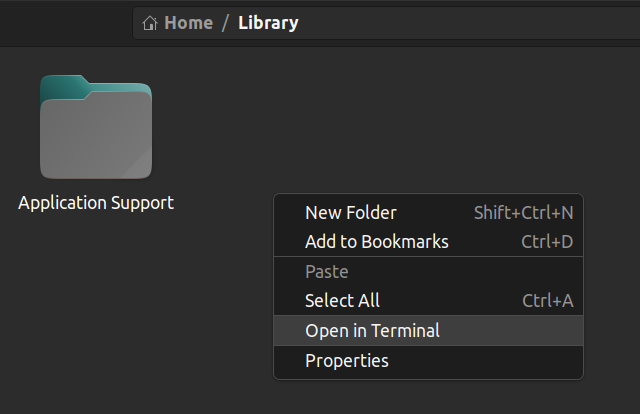
**Accessing the terminal**

The terminal is where you'll run all the commands and see the magic happen. You can search for the terminal by pressing the "windows" key and typing "terminal". You can pin the Terminal in the dock where other apps are located for easy access.



*💡 The shortcut for opening the terminal is ctrl+alt+t*

You can also open the terminal from inside a folder. Right click where you are and click on "Open in Terminal". This will open the terminal in the same path.



**How to use Linux on a Windows machine**

Sometimes you might need to run both Linux and Windows side by side. Luckily, there are some ways you can get the best of both worlds without getting different computers for each operating system.

**Option 1: "Dual-boot" Linux + Windows** With dual boot, you can install Linux alongside Windows on your computer, allowing you to choose which operating system to use at startup.

This requires partitioning your hard drive and installing Linux on a separate partition. With this approach, you can only use one operating system at a time.

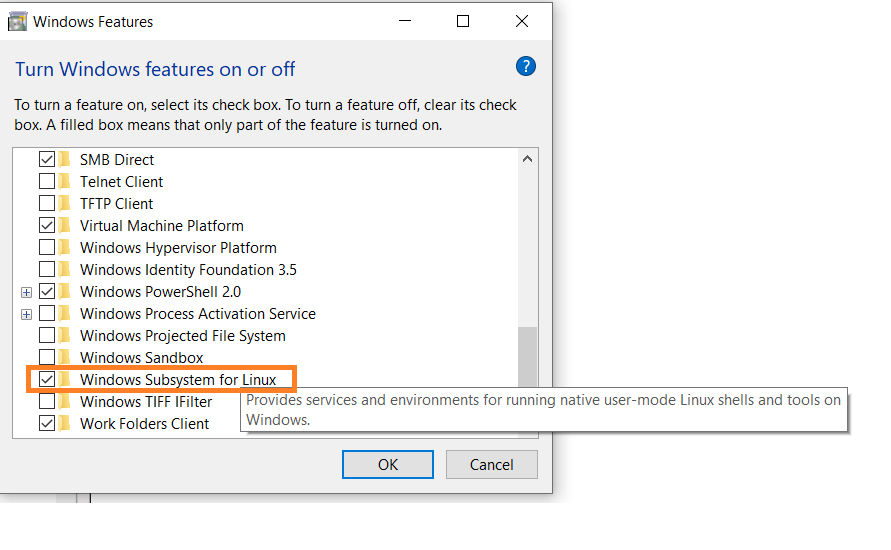
**Option 2:** Use Windows Subsystem for Linux (WSL) Windows Subsystem for Linux provides a compatibility layer that lets you run Linux binary executables natively on Windows.

Using WSL has some advantages. The setup for WSL is simple and not time-consuming. It is lightweight compared to VMs where you have to allocate resources from the host machine. You don't need to install any ISO or virtual disc image for Linux machines which tend to be heavy files. You can use Windows and Linux side by side.

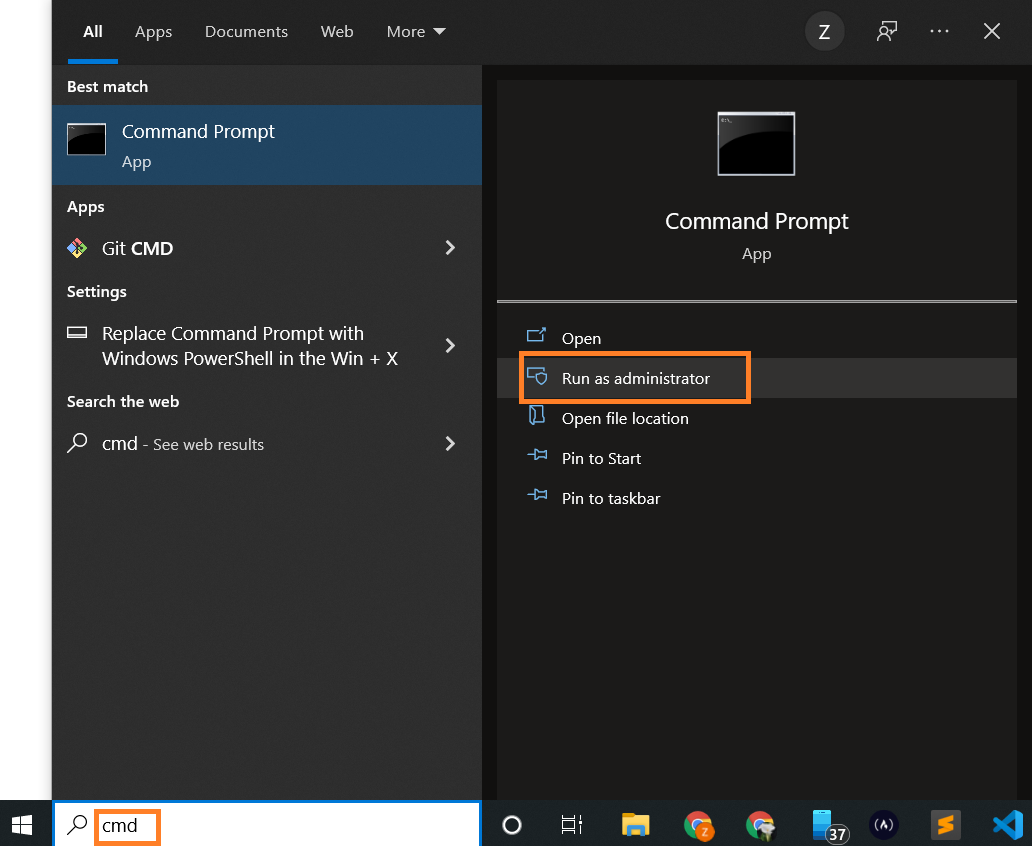
**How to install WSL2**

First, enable the Windows Subsystem for Linux option in settings.

* Go to Start. Search for "**Turn Windows features on or off.**"
* Check the option "**Windows Subsystem for Linux**" if it isn't already.



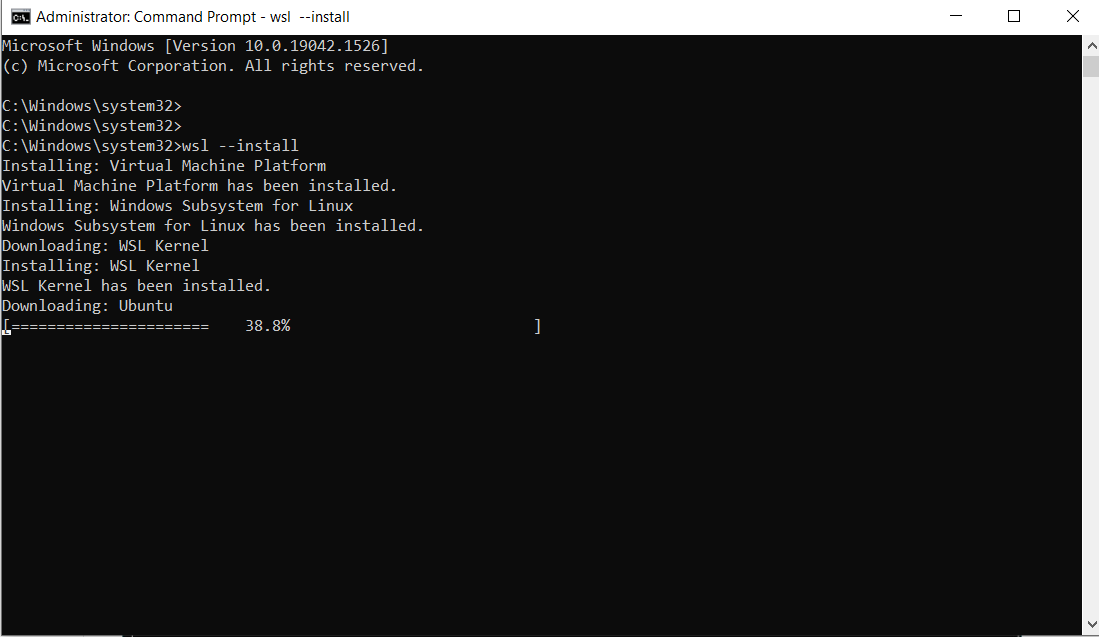
* Next, open your command prompt and provide the installation commands.
* Open Command Prompt as an administrator:



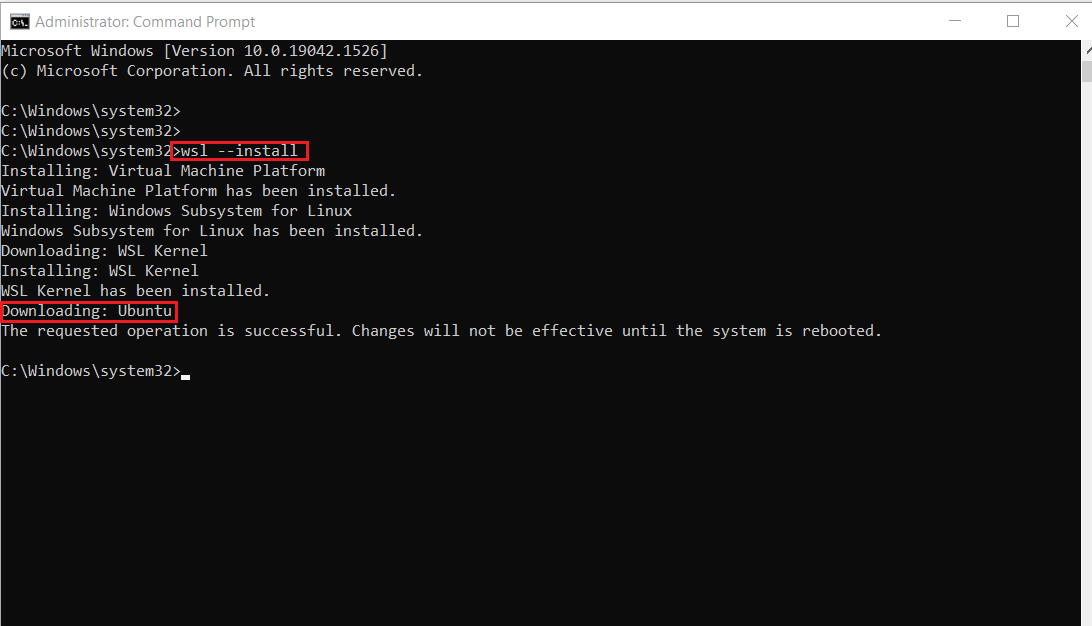
* Run the command below:

wsl --install

This is the output:

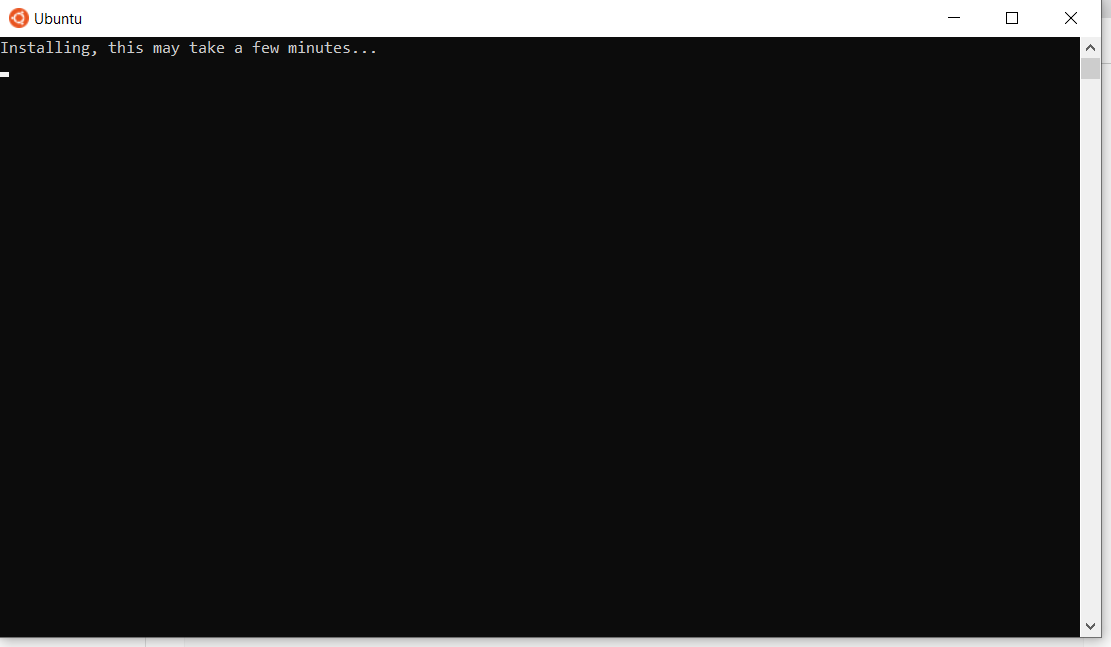


Note: By default, Ubuntu will be installed.

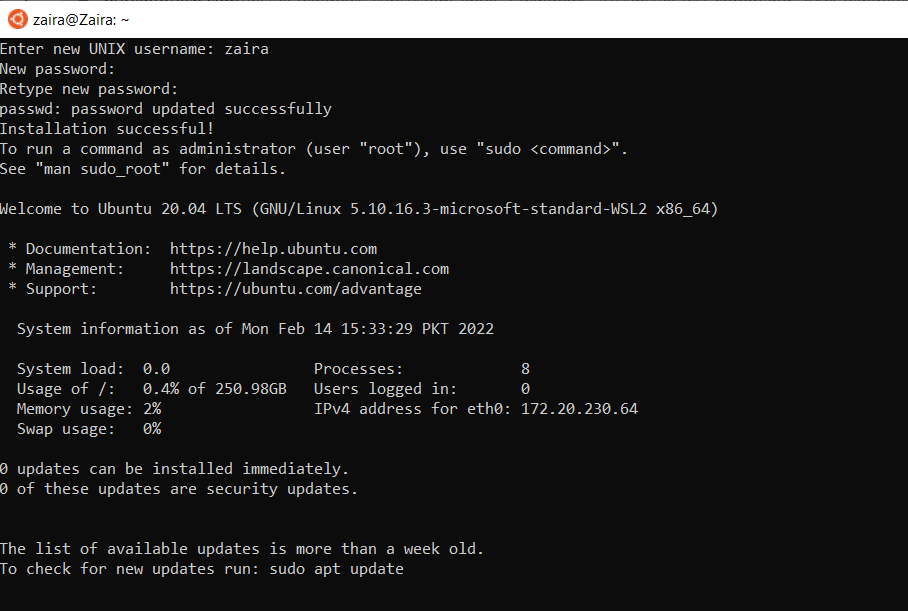


* Once installation is complete, you'll need to reboot your Windows machine. So, restart your Windows machine.

After restarting, you might see a window like this:

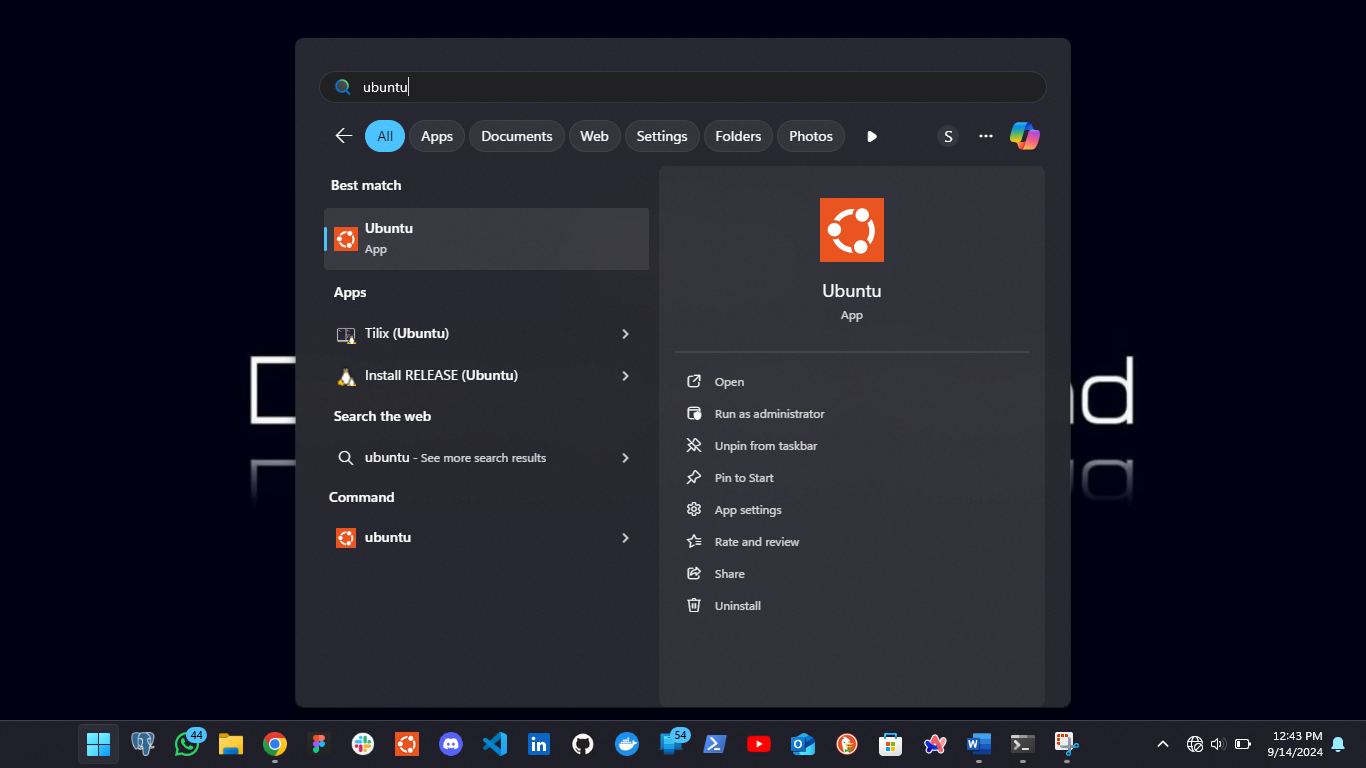


Once installation of Ubuntu is complete, you'll be prompted to enter your username and password.

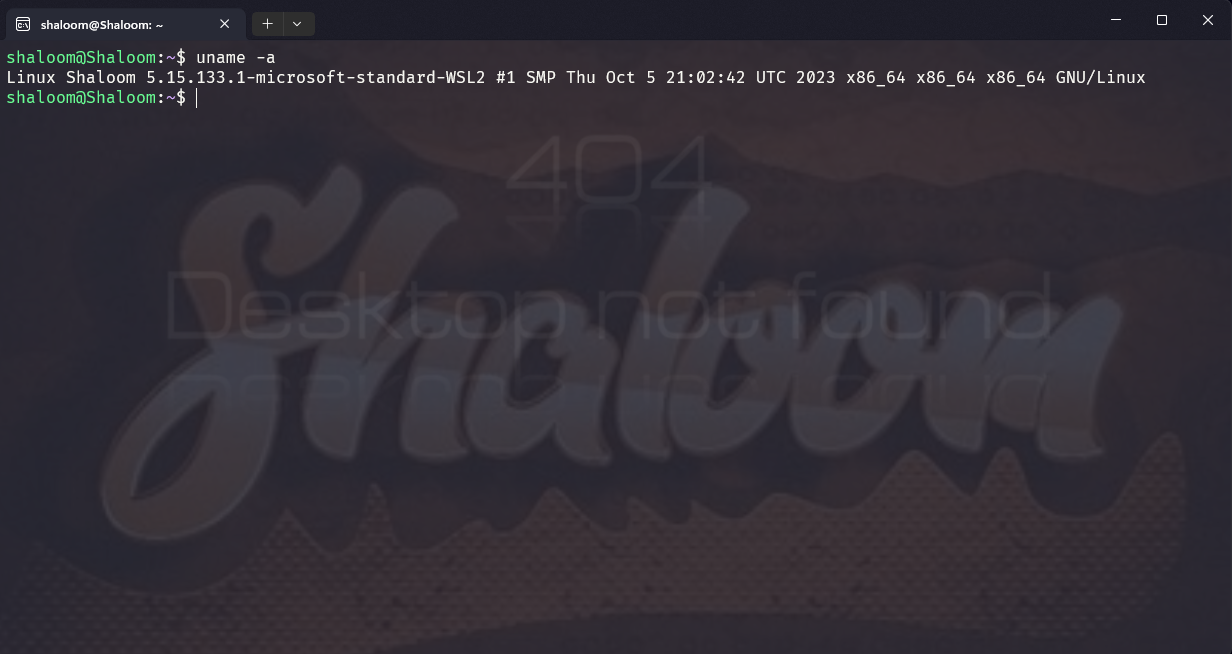


And, that's it! You are ready to use Ubuntu.

Launch Ubuntu by searching from the start menu.



And here we have your Ubuntu instance launched.



**Option 3: Use a Virtual Machine (VM)**

A virtual machine (VM) is a software emulation of a physical computer system. It allows you to run multiple operating systems and applications on a single physical machine simultaneously.

You can use virtualization software such as Oracle VirtualBox or VMware to create a virtual machine running Linux within your Windows environment. This allows you to run Linux as a guest operating system alongside Windows.

VM software provides options to allocate and manage hardware resources for each VM, including CPU cores, memory, disk space, and network bandwidth. You can adjust these allocations based on the requirements of the guest operating systems and applications.

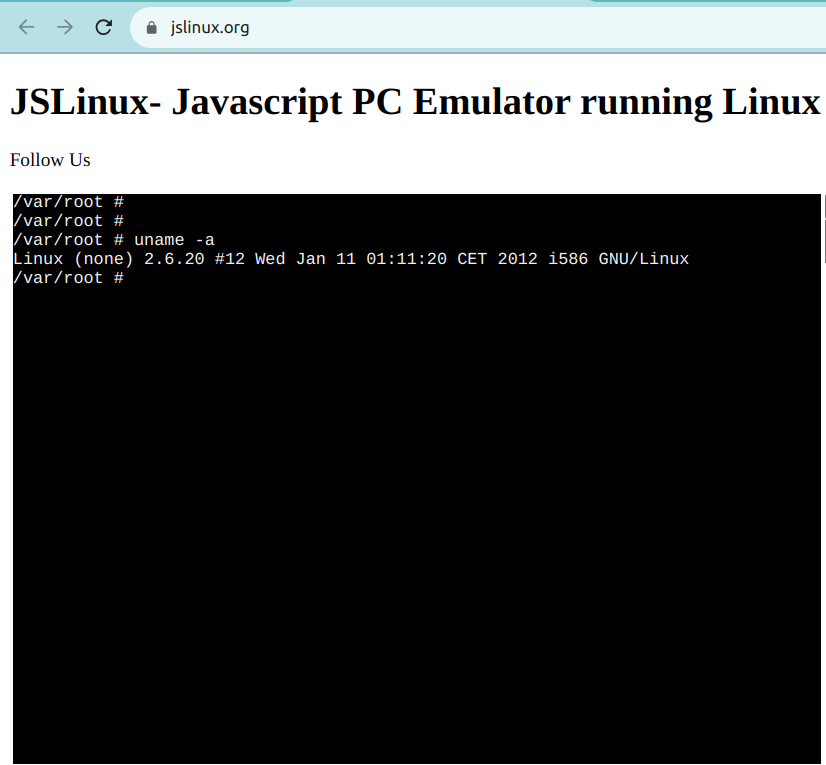
**Option 4: Use a Browser-based Solution**

Browser-based solutions are particularly useful for quick testing, learning, or accessing Linux environments from devices that don't have Linux installed.

**Web-based Linux terminals:**

Online Linux terminals allow you to access a Linux command-line interface directly from your browser. These terminals provide a web-based interface to a Linux shell, enabling you to execute commands and work with Linux utilities.

One such example is JSLinux. The screenshot below shows a ready-to-use Linux environment:



**Option 5: Use a Cloud-based Solution**

Instead of running Linux directly on your Windows machine, you can consider using cloud-based Linux environments or virtual private servers (VPS) to access and work with Linux remotely.

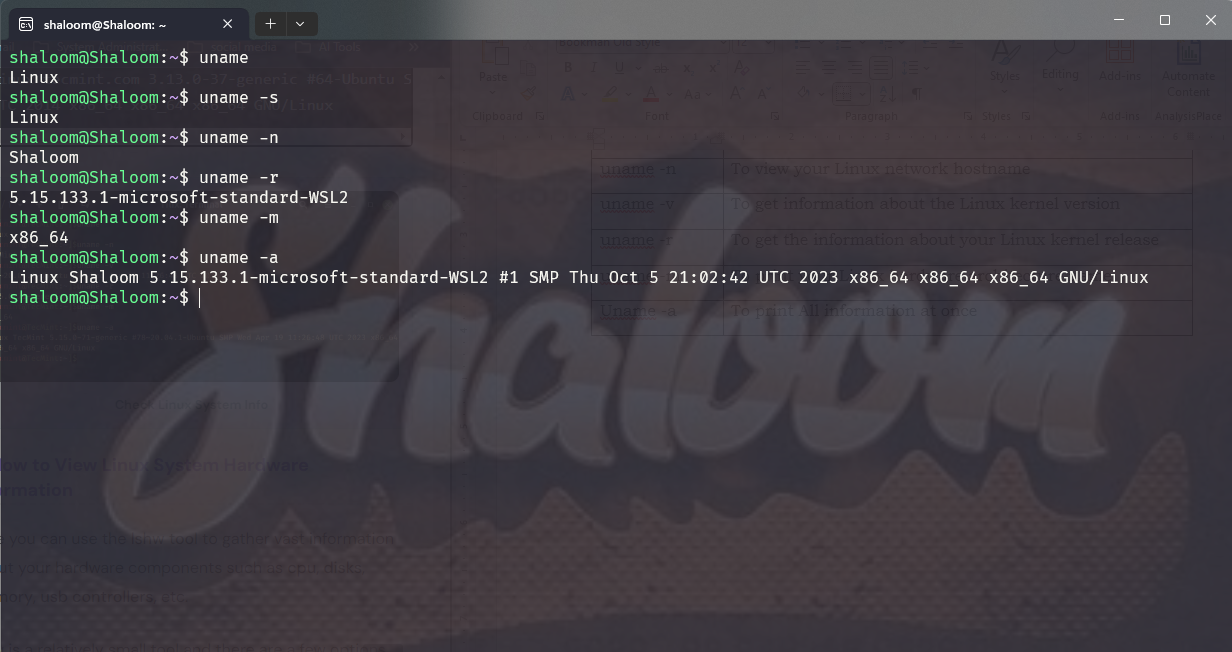
Services like **Amazon EC2**, **Microsoft Azure**, or Digital Ocean provide Linux instances that you can connect to from your windows computer. Note that some of these services offer free tiers, but they are not usually free in the long run.

**1.2: Applying Linux basics commands**

**Topic 1: System Information**

**1. How to View Linux System Information**

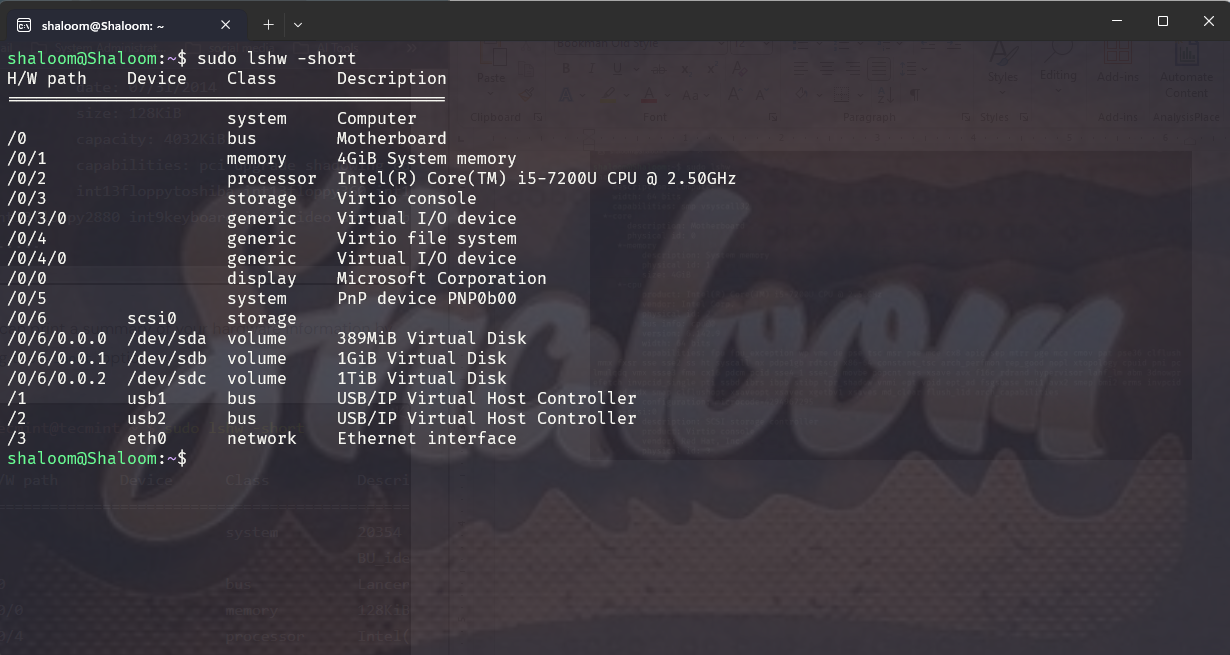
|  |  |
| --- | --- |
| **Command** | **Description** |
| uname | To print system information |
| uname -s | To print the kernel name of your system. |
| uname -n | To view your Linux network hostname |
| uname -v | To get information about the Linux kernel version |
| uname -r | To get the information about your Linux kernel release |
| uname -m | To print your Linux hardware architecture name |
| Uname -a | To print All information at once |



**2. How to View Linux System Hardware Information**

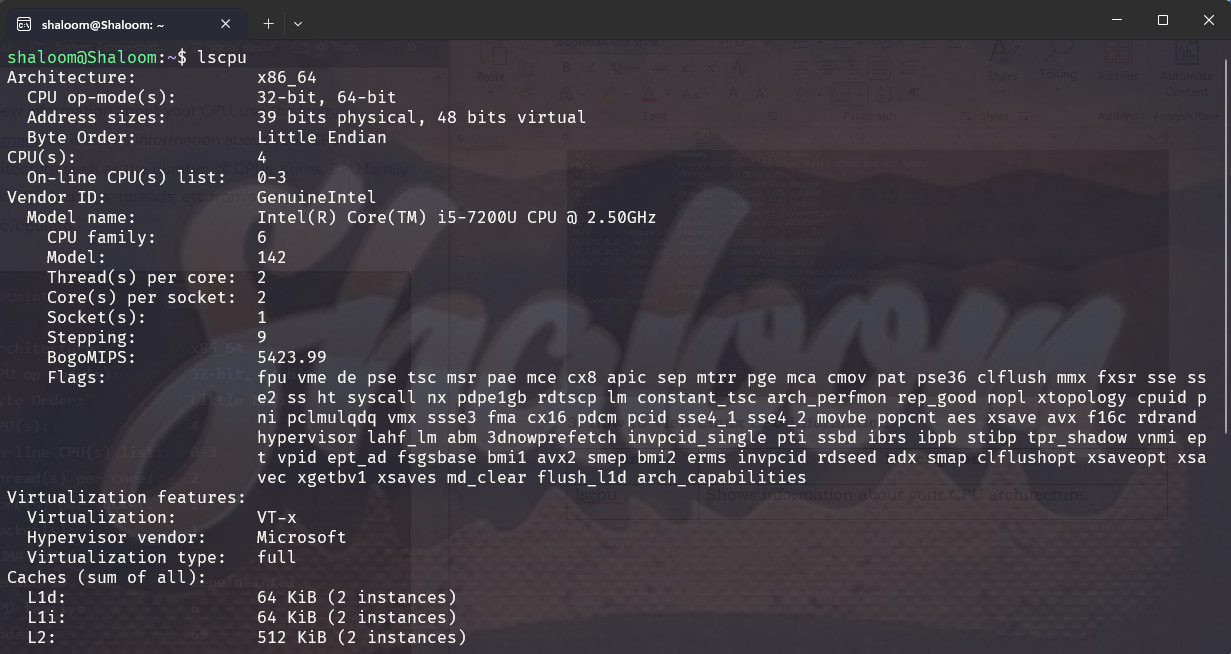
|  |  |
| --- | --- |
| **Command** | **Description** |
| lshw | To print information about your Linux system hardware such as cpu, disks, memory, usb controllers, etc. |
| lshw -short | To print a summary of your hardware information. |
| lshw -html > lshw.html | If you wish to generate output as an html file |





**3. How to View Linux CPU Information**

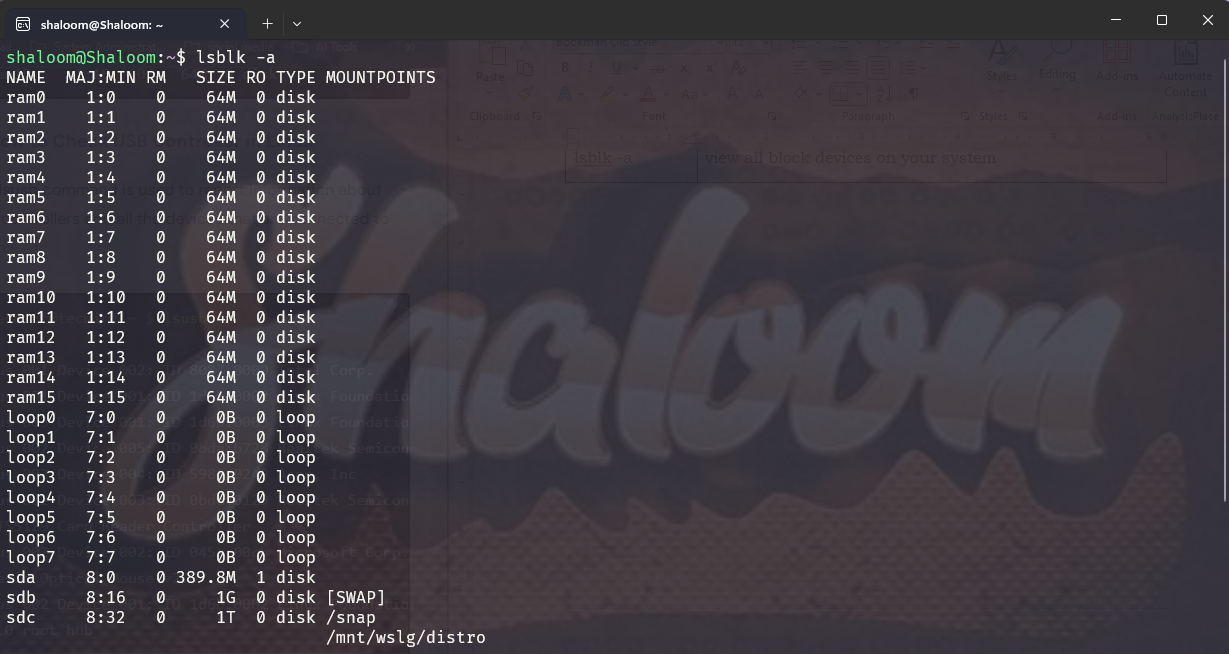
|  |  |
| --- | --- |
| **Command** | **Description** |
| lscpu | Shows information about your CPU architecture. |

****

**4. How to Collect Linux Block Device Information**

**Block devices** are storage devices such as hard disks, flash drives, etc.

|  |  |
| --- | --- |
| **Command** | **Description** |
| lsblk | used to report information about block devices |
| lsblk -a | view all block devices on your system |

****

**5. How to Check USB Controller in Linux**

|  |  |
| --- | --- |
| **Command** | **Description** |
| lsusb | Used to report information about USB controllers and all the devices that are connected to them. |
| lsusb -v | To generate detailed information about each USB device. |

**6. How to Check PCI Devices in Linux**

PCI devices may include USB ports, graphics cards, network adapters, etc.

|  |  |
| --- | --- |
| **Command** | **Description** |
| lspci | To generate information concerning all PCI controllers on your system plus the devices that are connected to them. |
| lspci -t | To produce output in a tree format. |
| lspci -v | To produce detailed information about each connected device. |

**Topic 2: File and Directory Management**

**1. File Management Commands**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| ls | Lists all files and directories in the current directory | ls |
| ls -la | Lists all files, including hidden files, with detailed information. | ls -la |
| touch | Creates an empty file. | touch filename.txt |
| cat | Displays the contents of a file. | cat filename.txt |
| cp | Copies files from one location to another | cp source.txt destination.txt |
| mv | Moves or renames file | mv oldname.txt newname.txt |
| rm | Removes/deletes files. | rm filename.txt |

**2. Directory Management Commands**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| pwd | Displays the current working directory | pwd |
| mkdir | Creates a new directory | mkdir new\_directory |
| rmdir | Removes an empty directory | rmdir directory\_name |
| rm -r | Removes a directory and its contents | rm -r directory\_name |
| cd | Changes the current directory. | cd /path/to/directory |
| cd .. | Moves up one directory level | cd .. |

**3. Permissions and Ownership**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| chmod | Changes the permissions of a file or directory | chmod 755 filename |
| chown | Changes the ownership of a file or directory | chown user:group filename |

**Topic 3: Text Processing**

**1. Viewing and Searching Text**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| cat | Displays the contents of a file. | cat filename.txt |
| more | Displays file content one screen at a time (useful for large files) | more file.txt |
| less | Similar to more, but allows backward/forward navigation in files | less file.txt |
| head | Shows the first few lines of a file (default is 10 lines) | head file.txt |
| tail | Shows the last few lines of a file (default is 10 lines). | tail file.txt |
| grep | Searches for a pattern in a file or output stream. | To find all lines containing the word "error" :  grep "error" logfile.txt |

**2. Text Manipulation**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| cat | Displays the contents of a file. | cat filename.txt |
| more | Displays file content one screen at a time (useful for large files) | more file.txt |
| less | Similar to more, but allows backward/forward navigation in files | less file.txt |
| head | Shows the first few lines of a file (default is 10 lines) | head file.txt |
| tail | Shows the last few lines of a file (default is 10 lines). | tail file.txt |
| grep | Searches for a pattern in a file or output stream. | To find all lines containing the word "error" :  grep "error" logfile.txt |
| sort | Sorts lines in a file. | sort file.txt |
| sort -r | To sort in reverse | sort -r file.txt |
| uniq | Removes duplicate lines in a sorted file or output. | uniq file.txt |
| tr | Translates or deletes characters in a file or stream. | tr 'a-z' 'A-Z' < file.txt  to converts lowercase letters to uppercase. |
| wc | Counts words, lines, characters, or bytes in a file. | wc file.txt |
| awk | A powerful text processing tool that works with columns and patterns. | awk '{print $1}' file.txt  To print the first column of each line. |
| sed | A stream editor that performs basic text transformations on input streams (e.g., search and replace). | sed 's/old/new/g' file.txt  This replaces all instances of "old" with "new" in the file. |
| | (Pipe) | Pipes the output of one command into another. | cat file.txt | grep "error"  Search for "error" in the output of a cat command. |
| > and >> (Redirection) | Redirect output to a file. | grep "error" logfile.txt > errors.txt # Overwrites  grep "error" logfile.txt >> errors.txt # Appends |

**Topic 4: Package Management**

Package management is essential for installing, updating, and removing software on a Linux system. Different Linux distributions use different package managers.

Here’s a guide to some of the most common package management tools:

**1.** **APT (Advanced Package Tool) for Debian/Ubuntu Based Systems**

|  |  |
| --- | --- |
| **Command** | **Description** |
| sudo apt update | Update the package list |
| sudo apt upgrade | Upgrade installed packages |
| sudo apt install package\_name | Install a package |
| sudo apt remove package\_name | Remove a package |
| sudo apt autoremove | Remove unnecessary packages (clean up) |
| apt search package\_name | Search for a package |
| apt show package\_name | View details about a package |

**2. YUM (Yellowdog Updater Modified) / DNF for CentOS/RHEL/Fedora Based Systems**

YUM is used on older versions of CentOS/RHEL, while DNF is its successor used in Fedora and recent RHEL/CentOS versions.

|  |  |
| --- | --- |
| **Command** | **Description** |
| * sudo yum update * sudo dnf update | Update the package list |
| * sudo yum install package\_name * sudo dnf install package\_name | Install a package |
| * sudo yum remove package\_name * sudo dnf remove package\_name | Remove a package |
| yum search package\_name | Search for a package |
| yum list installed | List installed packages |

**Topic 5: User and Group Management**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| adduser / useradd | Adds a new user to the system | sudo adduser username |
| * sudo useradd username |
| passwd | Sets or changes a user’s password. | sudo passwd username |
| usermod | Modifies user properties (like adding the user to a group). | sudo usermod -aG groupname username |
| deluser / userdel | Deletes a user from the system. | * sudo deluser username * sudo userdel username |
| who | Displays a list of currently logged-in users. | who |
| w | Shows detailed information about users logged in and what they are doing. | w |
| id | Displays user ID (UID), group ID (GID), and the groups the user belongs to. | id username |
| groupadd | Creates a new group. | sudo groupadd groupname |
| groupdel | Deletes a group. | sudo groupdel groupname |
| groupmod | Modifies a group (e.g., change the group name). | sudo groupmod -n newgroupname oldgroupname |
| gpasswd | Administers the /etc/group file and manages group memberships. |  |
| gpasswd -a | Add a user to a group: | sudo gpasswd -a username groupname |
| gpasswd -d | Remove a user to a group: | sudo gpasswd -d username groupname |
| cat /etc/passwd | Displays the list of all users on the system. | cat /etc/passwd |
| cat /etc/group | Displays the list of all groups on the system. | cat /etc/group |
| getent passwd | Lists users with details (more organized than /etc/passwd). | getent passwd |
| getent group | Lists groups with details. | getent group |
| su | Switches to another user (by default switches to root user if no username is provided). | su - username |
| sudo | Executes a command with superuser privileges. | sudo command |
| sudo su | Switches to the superuser (root) with the user’s own password. | sudo su |
| chown | Changes the ownership of a file or directory. | sudo chown user:group filename |
| chmod | Changes the permissions of a file or directory. | chmod 755 filename |
| chgrp | Changes the group ownership of a file or directory. | sudo chgrp groupname filename |

**NB:** The -aG flag adds the user to a group without removing them from other groups.

Examples of Common Tasks

**Create a new user:** sudo adduser newuser

**Change a user’s password:** sudo passwd newuser

**Add an existing user to a group:** sudo usermod -aG sudo newuser

**Delete a user and their home directory:** sudo userdel -r newuser

**Create a new group:**  sudo groupadd newgroup

**Add a user to a new group:** sudo gpasswd -a newuser newgroup

**Topic 6: System Control**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Example** |
| uname | Displays system information. | uname |
| uname -a | Display all system information | uname -a |
| hostname | Displays or sets the system's hostname. | hostname |
| uptime | Shows how long the system has been running, along with load averages. | uptime |
| top | Displays a real-time, dynamic view of running processes and system resource usage (CPU, memory, etc.). | top |
| htop | A more user-friendly, interactive version of top (requires installation). | htop |
| df | Displays disk space usage for file systems. | df -h |
| du | Shows disk usage for files and directories. | du -sh /path/to/directory |
| free | Displays memory (RAM) and swap usage. | free -h |
| vmstat | Displays virtual memory statistics. | vmstat |
| ps | Displays information about running processes for the current user | ps |
| ps -e | Displays information about all running processes | ps -e |
| ps aux | Displays detailed view of all processes | ps aux |
| kill | Terminates a process by PID (process ID). | kill PID |
| killall | Terminates all processes with a specific name. | killall processname |
| shutdown | Shuts down or reboots the system. |  |
| Schedule a shutdown after a certain number of minutes: | * sudo shutdown +10 * This will shut down the system after 10 minutes. |
| Shut down immediately | sudo shutdown now |
| Reboot the system | sudo shutdown -r now |
| Cancel a scheduled shutdown | sudo shutdown -c |
| reboot | Reboots the system immediately. | sudo reboot |
| halt | Stops all processes and halts the system. | sudo halt |
| poweroff | Turns off the system. | sudo poweroff |
| systemctl | Controls systemd services and daemons. |  |
| Start a service | sudo systemctl start servicename |
| Stop a service | sudo systemctl stop servicename |
| Restart a service | sudo systemctl restart servicename |
| Check the status of a service | systemctl status servicename |
| Enable a service to start at boot | sudo systemctl enable servicename |
| Disable a service from starting at boot | sudo systemctl disable servicename |
| View all active services | systemctl list-units --type=service |

**LO.1.3 Management of server services**

**1.3.1: Description of server services**

**Topic 1: Web Server**

A **web server** is a computer that hosts web pages, making them accessible online. When a user loads a site, the web server will retrieve the relevant files and send them to the browser so the user can interact with them.

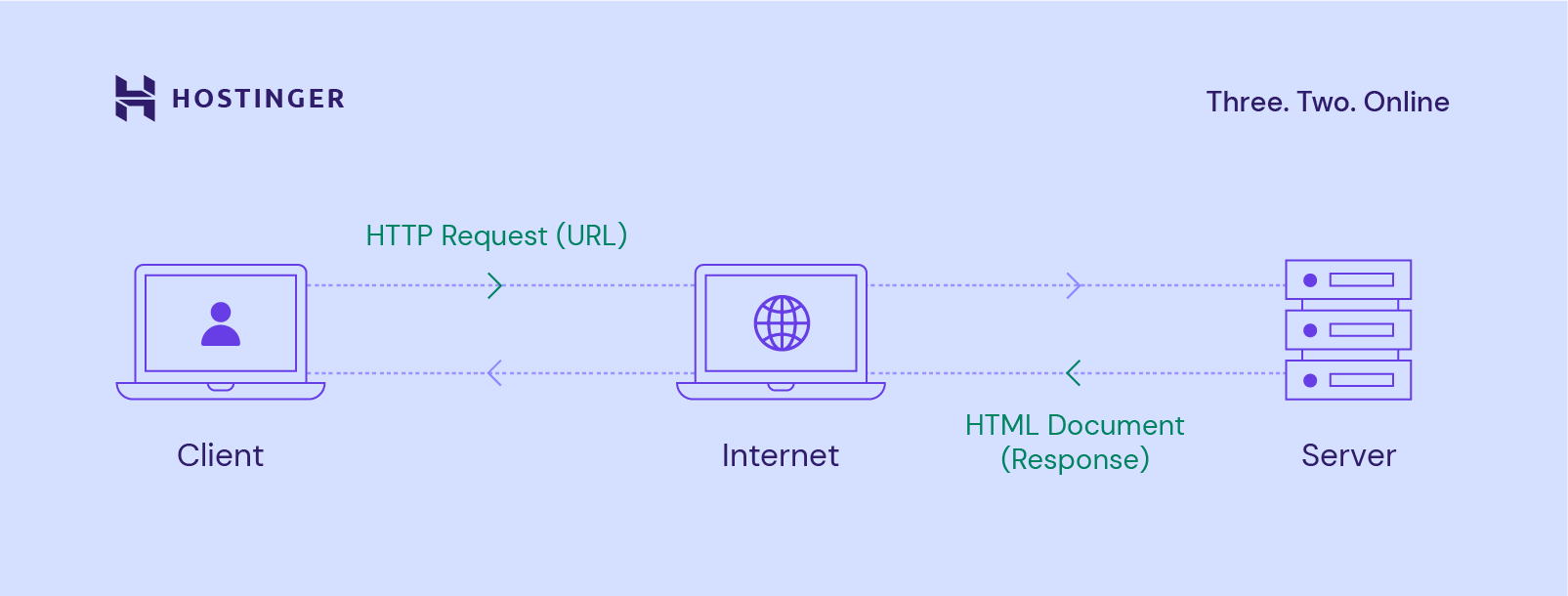
Its main role is to Hosts websites, handles HTTP requests, serves files, and can act as a reverse proxy.

Its Common Ports is HTTP (Port 80), HTTPS (Port 443)

Let’s break down the main components of a web server:

* **Software**. The web server software controls how users access hosted files. It comprises several components and houses at least an **HTTP server** to process and respond to incoming requests.
* **Hardware**. It stores web server software and its files, such as static HTML documents, JavaScript files, and CSS stylesheets. The web server hardware also connects to the internet, enabling data exchange with other physical devices.

**How Does a Web Server Work?**



Web and application servers follow a **client-server** model. In this structure, one program – the **client** – requests a resource or service from another program – the **server**.

Web servers use **Hypertext Transfer Protocol**(**HTTP**) when responding to user requests via the World Wide Web. **HTTP** is a protocol used to exchange information between computers.

Through the HTTP request process, servers can deliver the site’s **HTML document** to the user’s web browser, like Google Chrome.

Here’s an overview of the whole process to give you a better understanding:

1. When someone wants to load a website, the browser will look for the web server hosting the site’s files.
2. To achieve this, the web browser translates the site’s domain name into an IP address via the **Domain Name System (DNS)**. If the site is frequently visited, the web browser will search through its file cache.
3. After finding the corresponding web server, the browser sends an **HTTP request**to retrieve site content.
4. The web server receives and processes the HTTP request through its **HTTP server**. Once the HTTP server accepts the request, it will search through the database to obtain the relevant data.
5. Finally, the server returns the files to the web browser and delivers them to users.

When the HTTP server fails to find or process the requested files, it will send an **HTTP error status code** to the browser.

The most common error message is a **404 error**, which means the requested page is missing. Meanwhile, a **403 error** may appear if there are permission issues.

Furthermore, if a web server fails to receive a timely response from another server acting as a proxy or gateway, a **504 error** occurs.

**Static vs Dynamic Web Servers**

A graph showing how dynamic and static websites work


Web servers can generate both static and dynamic content depending on the installed software.

**Static**web servers comprise a computer and HTTP software. Meanwhile, a **dynamic**web server consists of a static web server plus extra software, commonly an application server and databases.

**Why Use a Web Server?**

Web servers can handle multiple tasks, such as sending and receiving emails, storing web applications, and processing FTP requests. However, the primary use of a web server is to host websites, making them functional and interactive for users worldwide.

benefits of having a trusted web host include:

* **High uptime**: Top hosting companies handle hardware maintenance and software updates to help maintain your website’s performance and security.
* **Secure servers**. By implementing effective security protocols, web hosts can reduce vulnerabilities and protect hosted websites against malware and cyber-attacks.
* **Various hosting options**. There are many web hosting packages with different features and benefits. You can select one that suits your needs and budget best.
* **Cost-effective**. You don’t have to maintain a dedicated web server, which can be very costly. Instead, choose a hosting plan that provides the necessary amount of server resources.
* **Flexibility**. Web hosts like Hostinger offer scalable plans, so you can manage multiple websites and upgrade the hosting resources as needed.

**Web Server Features**

* **File logging:** Log files track any events or activities a web server performs, such as requests, security, and error logs.
* **Authentication**: Many web servers offer this feature before permitting partial or complete access to a website’s resources. It often involves asking web users to provide a username and password.
* **Bandwidth limiting**: A web server’s bandwidth is the amount of data it can transfer or process at any given time. Bandwidth limiting controls the speed of responses to ensure that a network can deliver files smoothly.
* **Storage space**: It refers to the amount of disk space available to store files. The amount of storage space directly impacts the server’s ability to host and manage web pages and web applications.
* **Load balancing**: It is a technique employed by web servers to distribute incoming traffic across multiple servers. The purpose of load balancing is to minimize response times and prevent server overload.
* **Uptime guarantee:** Server uptime tracks how long a web server remains operational to process requests and deliver files. It directly impacts a hosted website’s availability.
* **Programming language support:** Also known as the server-side scripting language, it’s used to develop and run programs on the web server. A popular example is Python, which is supported by Hostinger’s virtual servers.

**Some of the most common web servers include:**

* **Apache HTTP Server**: A free and open-source web server used for many operating systems, including Windows, Linux, and macOS. **Apache** is one of the most popular choices among website owners, developers, and hosting providers, with amarket share of over 31%.
* **NGINX**. Initially designed only for HTTP web serving, this open-source software now also serves as a reverse proxy, HTTP load balancer, and email proxy. **NGINX** is known for its speed and ability to handle multiple connections, making it suitable for high-traffic websites.
* **Microsoft Internet Information Services (IIS)**.A closed web server software developed by Microsoft; IIS is widely used in Windows operating systems. It supports Active Server Pages (ASP), a server-side scripting technology developed by Microsoft for creating dynamic and interactive web applications.
* **Lighttpd**. A free and open-source web server software known for its fast data processing with less CPU power. Lighttpd is also popular for its small memory footprint, allowing the server to handle more requests while maintaining responsiveness and performance.

**Topic 2: Mail server**

A **mail server** (sometimes called an email server): is a software program that is responsible for sending emails from one email client to another.

**What are the types of mail servers?**

Mail servers can be divided into two categories: incoming mail servers and outgoing mail servers.

An incoming mail server stores mail and sends it to a user's inbox. Post Office Protocol 3 (POP3) and Internet Message Access Protocol (IMAP) are the two main types of incoming mail servers.

An outgoing mail server operates by having a user's machine communicate with Simple Mail Transfer Protocol (SMTP), which handles the email delivery process. SMTP servers work with other types of mail servers, namely POP3 or IMAP, to send emails from email clients.