February 6TH, 2023

Linux Administration

Prof. Dr. Soufiane Hourri

PLAN – LINUX ADMINISTRATION

- Introduction to Linux
- Linux Directory Structure and File Management
- Linux Shell and Shell Scripting
- Linux Package Management
- Comparison of Popular Distributions
- Introduction to Virtualization
- Setting up Fedora Server on a Virtual Machine
- Q&A
- Practice

Introduction to Linux

What is Linux ?

- Linux is a free and open-source operating system that was created by Linus Torvalds in 1991.
- Linux is based on the Unix operating system and has since become one of the most widely used operating systems in the world.
- Linux is known for its stability, security, and customization, and is used in a wide range of devices and applications, from personal computers and servers to mobile phones and embedded systems.
- Linux operates on the principles of open source software, meaning that the source code is freely available to the public, allowing anyone to use, modify, or distribute the software.

Brief history of evolution of Linux

- *Linus Torvalds*, a Finnish student, created the first version of the Linux kernel in 1991, which is the core component of an OS that manages communication between software and hardware.
- Linus Torvalds, while a student at the University of Helsinki, created Linux as a hobby project due to frustration with proprietary operating systems limitations, and released its first version under the GPL, allowing **free use**, **modification**, and **distribution of its source code**.
- Over the years, the Linux kernel has been developed and improved upon by a global community of developers and users, making it one of the **most reliable** and **secure** operating systems available today.
- Linux has evolved from a hobby project to a powerful, and widely used OS, powering servers, desktops, smartphones, and embedded systems.
- Linux has also become a popular platform for open-source software development, with a vast repository of free and open-source software available for users to install and use.

Key features of Linux operating systems

The Linux OS is known for its stability, security, and flexibility, and it offers a wide range of features that make it a popular choice for many different types of users. Here are some of the key features of Linux operating systems:

- **Open-source:** Linux is an open-source operating system, meaning that its source code is freely available for anyone to use, modify, and distribute. This has led to a large and thriving community of developers and users, who contribute to its development and improvement.
- **Customizable**: Linux is highly customizable, with a wide range of distributions available that cater to different user needs. From simple and user-friendly distributions like Ubuntu, to more specialized distribution like Kali Linux for security testing.
- **Command-line interface**: Linux provides a powerful command-line interface that allows users to perform complex tasks and automate processes more easily.

Key features of Linux operating systems

- **Package management:** Linux provides an easy-to-use package management system, which allows users to install, update, and remove software packages with just a few commands.
- **Security**: Linux is known for its security, with a strong focus on security features built into the operating system, as well as regular security updates.
- **Performance**: Linux is efficient and fast, and it can run on a wide range of hardware, from powerful servers to low-end embedded systems.
- **Community support**: Linux has a large and active community of users and developers, who provide support and assistance to each other, making it easy to get help when needed.

Linux Directory Structure and File Management

Understanding the file system hierarchy

In Linux, all files and directories are organized within a hierarchical file system, with a single root directory ("/") at the top, and all other files and directories organized under it. Here is a brief overview of the main elements of the Linux file system hierarchy:

- **/ (root directory):** This is the top-level directory in the file system hierarchy, and it contains all other files and directories in the system.
- **/bin:** This directory contains essential command-line utilities, such as ls, cp, mv, and rm, that are required for basic system operation.
- **/sbin:** This directory contains system administration utilities, such as shutdown and reboot, that are only accessible to the system administrator.
- **/usr:** This directory contains user-level software and documentation, such as games, text editors, and other user-level utilities.
- **/var:** This directory contains variable data, such as log files, that change over time.

Understanding the file system hierarchy

- **/etc:** This directory contains configuration files for the system and its applications.
- **/tmp:** This directory contains temporary files that can be deleted without affecting system operation.
- **/home:** This directory contains the home directories for individual users, where they can store their personal files and settings.

Common file and directory commands (e.g. ls, cp, mv, rm)

The Linux OS provides a wide range of commands for managing files and directories, here are some of the most commonly used ones:

- **ls:** This command is used to list the files and directories in a directory. For example, "ls /" will list the files and directories in the root directory.
- **cp:** This command is used to copy files and directories from one location to another. For example, "cp file1.txt file2.txt" will copy the file "file1.txt" to "file2.txt".
- **mv:** This command is used to move files and directories from one location to another, or to rename a file or directory. For example, "mv file1.txt file2.txt" will rename the file "file1.txt" to "file2.txt".
- **rm:** This command is used to delete files and directories. For example, "rm file1.txt" will delete the file "file1.txt".

Common file and directory commands (e.g. ls, cp, mv, rm)

- **mkdir:** This command is used to create a new directory. For example, "mkdir mydir" will create a new directory called "mydir".
- **rmdir:** This command is used to delete an empty directory. For example, "rmdir mydir" will delete the empty directory "mydir".
- **touch:** This command is used to create a new empty file. For example, "touch myfile.txt" will create a new empty file called "myfile.txt".

Linux Shell and Shell Scripting

Overview of the shell and its uses

The shell is the command-line interface in a Linux OS. It provides a way for users to interact with the underlying system by typing commands and receiving output. The shell provides many features and utilities for automating tasks, manipulating files and directories, and managing system resources.

Some of the main uses of the shell include:

- **Automating repetitive tasks**: The shell allows users to automate repetitive tasks by writing shell scripts, which are sequences of commands that can be executed automatically.
- **File and directory management**: The shell provides a wide range of commands for managing files and directories, including copying, moving, renaming, and deleting files.
- **Process management**: The shell allows users to start, stop, and monitor the status of processes running on the system.
- **System administration**: The shell provides a way for system administrators to manage and monitor the underlying system, including managing users, setting up networks, and installing software.
- **Program development**: The shell provides a powerful environment for developing and testing programs, including the ability to run and debug programs from the command line.

Basic shell commands and environment variables

Here are some basic shell commands and environment variables that are commonly used in the Linux shell:

- **cd:** This command is used to change the current working directory. For example, "cd /" will change the current directory to the root directory.
- **echo:** This command is used to display text on the screen. For example, "echo Hello World" will display the text "Hello World".
- **export:** This command is used to set environment variables. For example, "export MY_VAR=value" will set the environment variable "MY_VAR" to "value".
- **set:** This command is used to set shell variables. For example, "set MY_VAR=value" will set the shell variable "MY_VAR" to "value".
- **pwd:** This command is used to display the current working directory.
- **env:** This command is used to display the environment variables.
- **PATH:** This is an environment variable that specifies the directories in which the shell should search for executables.
- **HOME:** This is an environment variable that specifies the home directory of the current user.

Linux Package Management

Understanding package managers (e.g. apt, yum, pacman)

In Linux operating systems, package managers are used to install, manage, and remove software packages from the system. Here are some of the most commonly used package managers in Linux:

- **apt:** This is the package manager for Debian-based systems, such as Ubuntu. It is used to manage packages and dependencies and provides a user-friendly interface for installing, updating, and removing software packages.
- **yum:** This is the package manager for Red Hat-based systems, such as Fedora and CentOS. It provides a centralized repository of software packages and automates the process of installing, updating, and removing software packages.
- **pacman:** This is the package manager for Arch Linux. It is a simple, yet powerful package manager that provides a fast and efficient way to install, update, and remove software packages.

Installing, updating, and removing packages

Here is an overview of how to install, update, and remove packages using some of the most common package managers in Linux:

- apt (Debian-based systems):
  - Installing packages: "sudo apt-get install <package-name>"
  - Updating packages: "sudo apt-get update && sudo apt-get upgrade"
  - Removing packages: "sudo apt-get remove <package-name>"
- yum (Red Hat-based systems):
  - Installing packages: "sudo yum install <package-name>"
  - Updating packages: "sudo yum update"
  - Removing packages: "sudo yum remove <package-name>"
- pacman (Arch Linux):
  - Installing packages: "sudo pacman -S <package-name>"
  - Updating packages: "sudo pacman -Syu"
  - Removing packages: "sudo pacman -R <package-name>"

Comparison of Popular Distributions

Overview of the different Linux distributions available

There are many different Linux distributions available, each with its own unique features, strengths, and purposes. Some popular distributions include:

- **Fedora:** A community-driven, open-source distribution sponsored by Red Hat.
- **Ubuntu:** A popular, user-friendly distribution based on Debian.
- **CentOS:** A community-driven distribution based on Red Hat Enterprise Linux, designed for server use.
- **Debian:** A widely used, community-driven distribution known for its stability and large software repository.
- **Arch Linux:** A flexible and rolling release distribution that is popular among Linux enthusiasts and power users.
- **Mint:** A user-friendly distribution based on Ubuntu, designed for desktop use.
- **OpenSUSE:** A community-driven distribution that is known for its focus on open-source software and its easy-to-use interface.

Explanation of why comparing distributions is important

Comparing different Linux distributions is important because it helps you choose the right distribution for your specific needs and requirements. Some reasons why comparing distributions is important include:

- **Features:** Different distributions have different features, such as user interface, software repositories, and stability.
- **Performance:** Different distributions have different system requirements, which can affect performance.
- **Purpose:** Different distributions are designed for different purposes, such as desktop use, server use, or development.
- **Community:** Different distributions have different communities, with different levels of support, documentation, and contributions.

Overview of Fedora and its key features

Fedora is a community-driven, open-source distribution of Linux that is sponsored by Red Hat. It is known for its focus on open-source software and its cutting-edge technology. Some key features of Fedora include:

- **Latest software**: Fedora is known for being on the cutting edge of Linux technology, offering the latest software and features.
- **Stability**: Despite its focus on new technology, Fedora is also known for its stability and reliability.
- **Community**: Fedora has a large and active community of users, developers, and contributors, who provide support and help drive the development of the distribution.
- **Easy installation**: Fedora features an easy-to-use installation process that makes it simple to set up and start using.
- **Package management**: Fedora uses the DNF package manager, which is known for its speed, reliability, and easy-to-use interface.
- **Focus on open-source software**: Fedora places a strong emphasis on open-source software, and most of the software in its repositories is open-source.

Disadvantages of using Fedora

- **Less beginner-friendly:** Fedora is not as beginner-friendly as some other distributions, and may require more technical knowledge to set up and use.
- **Fewer commercial applications:** Since Fedora places a strong emphasis on open-source software, there may be fewer commercial applications available compared to other distributions.
- **Shorter support cycle:** Fedora has a shorter support cycle compared to other distributions, which means that older versions of Fedora may not receive updates and security fixes.
- **Cutting-edge technology:** While the focus on new technology is an advantage for some users, it can also be a disadvantage for others who may prefer more stability and reliability.

Overview of Ubuntu and its key features

Ubuntu is one of the most popular Linux distributions, known for its user-friendly interface, ease of use, and strong community support. Some of the key features of Ubuntu include:

- **User-friendly interface**: Ubuntu features a user-friendly interface with a clean and intuitive design that is easy to navigate.
- **Large community**: Ubuntu has a large and active community of users, developers, and contributors, who provide support and help drive the development of the distribution.
- **Easy installation**: Ubuntu features an easy-to-use installation process that makes it simple to set up and start using.
- **Package management**: Ubuntu uses the APT package manager, which is known for its speed, reliability, and easy-to-use interface.
- **Focus on open-source software**: Ubuntu places a strong emphasis on open-source software, and most of the software in its repositories is open-source.
- **Long-term support**: Ubuntu provides long-term support, with security updates and bug fixes for up to five years for each release.
- **Large software repository**: Ubuntu has a large software repository, with thousands of software packages available for easy installation.

Disadvantages of using Ubuntu

- **Lack of customization options**: Ubuntu has limited customization options compared to other distributions, which can be a drawback for users who like to have control over the look and feel of their operating system.
- **Stability issues**: In some cases, users have reported stability issues with Ubuntu, which can make it difficult to use for some tasks.
- **Hardware compatibility**: Some hardware components may not be compatible with Ubuntu, which can make it challenging to use for some users.
- **Upgrades can be problematic**: Some users have reported problems with upgrading to new versions of Ubuntu, which can cause issues with compatibility and stability.

Overview of CentOS and its key features

CentOS is a community-driven, open-source distribution of Linux, derived from the source code of Red Hat Enterprise Linux. Here are some of its key features:

- **Stability and reliability:** CentOS is known for its stability and reliability, and is widely used in production environments.
- **Long-term support:** CentOS provides long-term support, with security updates and bug fixes for up to ten years.
- **Focus on enterprise features:** CentOS is designed with enterprise features in mind, and is often used as a server operating system.
- **Compatibility with Red Hat Enterprise Linux:** CentOS is compatible with Red Hat Enterprise Linux, making it easy for users who are familiar with Red Hat to transition to CentOS.
- **Package management:** CentOS uses the YUM package manager, which is known for its ease of use and reliability.
- **Large community:** CentOS has a large and active community of users, developers, and contributors, who provide support and help drive the development of the distribution.
- **Compatibility with popular software**: CentOS is compatible with many popular software packages, making it easy to use for a wide range of tasks.

Disadvantages of using CentOS

- **Limited software availability**: While CentOS is compatible with many popular software packages, some newer or less popular software may not be available in the official repositories.
- **Older versions of software**: Since CentOS is focused on stability and reliability, it may not always provide the latest versions of software.
- **Lack of cutting-edge features**: Since CentOS is focused on enterprise features, it may not always have the latest and greatest features that are available in other distributions.
- **Complexity**: CentOS is designed for use in production environments, and as such, it may have a steeper learning curve for new users, or for those who are used to more user-friendly distributions.
- **Limited support for newer hardware**: Since CentOS is focused on stability, it may not always support the latest hardware, and you may need to use third-party repositories or install additional drivers.

Overview of Debian and its key features

Debian is a community-driven, open-source distribution of Linux. Here are some of its key features:

- **Wide software compatibility**: Debian is compatible with a wide range of software packages, making it easy to use for a wide range of tasks.
- **Large and active community**: Debian has a large and active community of users, developers, and contributors, who provide support and help drive the development of the distribution.
- **Package management**: Debian uses the APT package manager, which is known for its ease of use and reliability.
- **Stable and reliable**: Debian is known for its stability and reliability, and is widely used in production environments.
- **Focus on free software**: Debian is dedicated to providing only free software, and is one of the largest distributors of free software in the world.
- **Wide hardware compatibility**: Debian is compatible with a wide range of hardware, making it easy to use on many different types of computers.
- **Customization**: Debian provides a large range of customization options, allowing users to easily configure their systems to meet their specific needs.

Disadvantages of using Debian

- **Slower release cycle**: Debian's focus on stability and reliability means that new releases are not made as frequently as other distributions.
- **Older software versions**: Debian may not always provide the latest version of a particular software package, as it focuses on stability and reliability.
- **Complexity**: Debian is designed for more advanced users and can be more complex to set up and use, especially for those who are new to Linux or are used to more user-friendly distributions.
- **Limited commercial support**: While Debian has a large and active community, commercial support options for the distribution may be limited.
- **Dependency issues**: Debian's package management system can sometimes result in dependency issues, making it difficult to install certain software packages.

Comparison of these distributions to Fedora, Ubuntu, CentOS, and Debian

- **Arch Linux vs Fedora**: Both Arch Linux and Fedora are designed for advanced users who want complete control over their systems. However, Arch Linux is a rolling release distribution that is focused on cutting-edge software, while Fedora focuses on stability and security.
- **Mint vs Ubuntu**: Both Mint and Ubuntu are designed for user-friendliness and ease of use. However, Mint provides a more modern desktop environment and updated software packages, while Ubuntu focuses on stability and long-term support.
- **Manjaro vs Arch Linux**: Both Manjaro and Arch Linux are based on Arch Linux and provide a user-friendly interface and easy-to-use package management system. However, Manjaro is designed to be more user-friendly, while Arch Linux is designed for experienced users.
- **Solus vs Debian**: Both Solus and Debian are designed to be easy to use, fast, and stable. However, Solus provides a custom desktop environment and a user-friendly package management system, while Debian is more complex and focused on stability and reliability.
- **OpenSUSE vs CentOS**: Both OpenSUSE and CentOS are community-driven distributions that are known for their stability, security, and ease of use. However, OpenSUSE provides a range of desktop environments, while CentOS is focused on server applications and stability.

Introduction to Virtualization

Definition and explanation of virtualization

- Virtualization is a technology that allows multiple virtual machines to run on a single physical host machine, sharing its resources such as CPU, memory, storage, and network.
- Virtualization enables multiple operating systems and applications to run on a single hardware infrastructure, creating a virtual environment for each application or operating system.
- Virtualization provides a layer of abstraction between the virtual machines and the physical hardware, allowing each virtual machine to be isolated from the others and to run as if it were running on its own physical machine.
- Virtualization allows organizations to maximize their hardware utilization, reduce costs, and improve security, as well as providing a flexible and scalable infrastructure.

Why virtualization is important

Virtualization is important for several reasons:

- **Resource utilization**: Virtualization allows multiple virtual machines to run on a single physical host machine, making it possible to maximize hardware utilization and reduce the need for additional hardware.
- **Cost savings**: By using virtualization, organizations can reduce costs associated with hardware procurement, maintenance, and energy consumption.
- **Scalability and flexibility**: Virtualization enables organizations to quickly provision new virtual machines and easily allocate resources as needed, providing a flexible and scalable infrastructure.
- **Improved security**: Virtualization allows organizations to isolate applications and operating systems, reducing the risk of security breaches and making it easier to manage and secure the environment.
- **Disaster recovery and business continuity**: Virtualization provides the ability to easily backup, restore, and move virtual machines, making it easier to implement disaster recovery and business continuity plans.
- **Application compatibility and testing**: Virtualization makes it possible to test and run applications on different operating systems and hardware configurations, improving compatibility and reducing the risk of application failure.

Types of Virtualization

There are three main types of virtualization:

- **Server virtualization**: The process of creating and running multiple virtual servers on a single physical server.
- **Desktop virtualization**: The process of creating and running multiple virtual desktops on a single physical machine, providing remote access to desktops and applications.
- **Application virtualization**: The process of running applications in a virtual environment, isolated from the underlying operating system and hardware. This allows for compatibility and simplifies application deployment.

Types of Virtualization

In addition to these main types, there are also:

- **Storage virtualization**: The process of pooling physical storage from multiple networked storage devices into what appears to be a single storage device that is managed from a central console.
- **Network virtualization**: The process of creating virtual networks that run on top of a physical network, providing isolated and scalable network resources.
- **Hardware virtualization**: The process of abstracting the physical hardware of a computer and presenting it to the operating system as virtual hardware. This allows multiple operating systems to run on the same physical machine.

The key components of virtualization

The key components of virtualization include

- **Virtualization software (hypervisor):** This software acts as a layer between the physical hardware and virtual machines, providing the virtual machines with resources such as CPU, memory, and storage. Examples of hypervisors include Vmware, VirtualBox.
- **Virtual machines (guests):** These are software-based representations of physical computers that run their own operating systems and applications. Virtual machines are isolated from each other and can run different operating systems and applications.
- **Virtualization Management Tools**: These are tools used to manage and monitor the virtual environment, including creating and deleting virtual machines, allocating resources, and monitoring performance. Examples include VMware vCenter, Microsoft System Center Virtual Machine Manager, and Citrix XenCenter.
- **Virtual Storage**: Virtual storage is used to store virtual machine data, including virtual hard disks and virtual machine snapshots.
- **Virtual Network**: Virtual networks provide network connectivity for virtual machines, allowing them to communicate with each other and with the outside world. Examples include virtual switches, virtual firewalls, and virtual routers.
- **Virtual Devices**: Virtual devices include virtual hard disks, virtual CD/DVD drives, and virtual serial and parallel ports. These devices are used to store data and provide input/output for virtual machines.

Virtualization Techniques

There are two main virtualization techniques:

- **Full Virtualization**: Full virtualization allows multiple virtual machines to run on a single physical host, each with its own operating system. The hypervisor emulates the underlying hardware and provides each virtual machine with its own virtualized hardware, including virtual CPU, memory, storage, and network interface.
- **Paravirtualization**: Paravirtualization requires each virtual machine to run a modified operating system that is aware of the virtual environment. In paravirtualization, virtual machines share the underlying physical hardware, but the hypervisor provides each virtual machine with its own virtualized hardware resources.
- **Other Virtualization Techniques**: There are other virtualization techniques, such as hardware-assisted virtualization, container virtualization, and operating system virtualization. These tech niques are used for specific purposes, such as improving performance or providing isolation at the operating system level.

Popular Virtualization Platforms

- **VMware**: VMware is a leading virtualization platform that offers both server and desktop virtualization solutions. It provides a full range of virtualization solutions, including VMware Workstation, VMware vSphere, and VMware vCloud.
- **Hyper-V:** Hyper-V is Microsoft's virtualization platform, which is integrated into Windows Server. Hyper-V allows multiple virtual machines to run on a single physical host, providing a scalable and flexible virtualization solution.
- **VirtualBox**: VirtualBox is a free and open-source virtualization platform that is designed for desktop virtualization. It provides a simple and easy-to-use interface for creating and managing virtual machines.
- **KVM**: KVM (Kernel-based Virtual Machine) is an open-source virtualization platform that is integrated into the Linux kernel. It provides a high-performance virtualization solution for Linux servers and desktops.
- **Xen**: Xen is an open-source virtualization platform that provides both server and desktop virtualization solutions. It is used in many popular Linux distributions, including CentOS, Debian, and Ubuntu.

Use Cases for Virtualization

- **Server Consolidation**: Virtualization can be used to consolidate multiple servers onto a single physical host, reducing hardware costs and improving resource utilization.
- **Disaster Recovery**: Virtualization allows organizations to quickly recover from disaster by quickly spinning up virtual machines on alternative physical hosts.
- **Testing and Development**: Virtualization enables developers to test their applications in different operating systems and environments, without the need for multiple physical machines.
- **Cloud Computing**: Virtualization is a key component of cloud computing, allowing multiple virtual machines to run on a single physical host, with each virtual machine isolated from the others.
- **Security**: Virtualization provides isolation between virtual machines, allowing organizations to securely run multiple applications on the same physical host.
- **Application Portability**: Virtualization makes it possible to run applications on multiple operating systems and environments, without the need to recompile or modify the application.

Advantages of Virtualization

- **Cost Savings**: Virtualization reduces hardware costs by allowing multiple virtual machines to run on a single physical host.
- **Improved Resource Utilization**: Virtualization improves resource utilization by allowing multiple virtual machines to share the same physical resources.
- **Improved Availability**: Virtualization can improve availability by allowing virtual machines to be quickly moved to alternative physical hosts in the event of a hardware failure.
- **Flexibility**: Virtualization enables organizations to quickly spin up new virtual machines as needed, without the need to purchase and install new hardware.
- **Security**: Virtualization provides isolation between virtual machines, allowing organizations to securely run multiple applications on the same physical host.

Disadvantages of Virtualization

- **Performance Overhead**: Virtualization introduces a performance overhead, as virtual machines must share the same physical resources.
- **Complexity**: Virtualization can add complexity to an IT infrastructure, as virtual machines must be managed and maintained.
- **Licensing Costs**: Some virtualization platforms may require additional licensing costs, which can be expensive for organizations with large numbers of virtual machines.
- **Dependency on the Physical Host**: Virtual machines are dependent on the physical host, and if the host fails, all virtual machines running on that host will be impacted.
- **Security Risks**: Virtualization can introduce security risks, as virtual machines may be vulnerable to attack from the physical host or other virtual machines.

Setting up Fedora Server on a Virtual Machine

Prerequisites

System Requirements:

- **Operating System**: Windows 7 or later, MacOS X 10.11 or later, or Linux distribution with graphical user interface.
- **Processor**: Intel Core i5 or higher
- **RAM**: 4GB or more
- **Hard Disk Space**: 10GB or more
- **Graphics Card**: Intel HD Graphics or higher
- **Network**: Broadband internet connection
- **Screen Resolution**: Minimum 1024x768 pixels.
- Additional software or hardware may be required based on the specific use case.

ISO files

- ISO files are image files that contain the entire contents of a bootable CD or USB drive.
- ISO files are commonly used to distribute software, including operating systems, because they provide a complete and self-contained environment that can be run on any compatible machine.
- In virtualization, ISO files are used to install an OS in a virtual machine.
- The virtual machine acts as a virtual computer that runs on top of the host operating system.
- The ISO file is loaded into the virtual machine and the virtual machine's CD/DVD drive is set to boot from the ISO file.
- This will start the installation process of the operating system contained in the ISO file, allowing you to install and run the operating system in a virtual environment.

Step-by-step instructions for installing Fedora Server ISO on the virtual machine

- **Choose a virtualization platform**: Choose a virtualization platform that suits your system requirements and the type of virtualization you want to perform.
- **Create a new virtual machine**: Create a new virtual machine on your chosen virtualization platform.
- **Configure virtual machine settings**: Configure the virtual machine settings such as RAM, hard disk, network, and ISO image.
- **Boot the virtual machine**: Boot the virtual machine and select the Fedora Server ISO image.
- **Begin the installation process**: Begin the installation process and follow the on-screen instructions to install Fedora Server on the virtual machine.
- **Configure system settings**: Configure the system settings such as language, keyboard, and time zone.
- **Partition the hard disk**: Partition the hard disk according to your needs.
- **Install the operating system**: Install the operating system on the virtual machine by selecting the appropriate partition.
- **Configure user accounts**: Configure user accounts and set a password for the root user.
- **Complete the installation process**: Complete the installation process and reboot the virtual machine.
- **Log in to Fedora Server**: Log in to Fedora Server and verify that it is installed correctly.

Creating users

- Open the terminal by pressing Ctrl + Alt + T or by searching for "terminal" in the applications menu.
- Become the root user by entering the following command: sudo su
- Type the following command to add a new user: useradd [username] where [username] is the name of the new user.
- Create a password for the new user by entering the following command: passwd [username] where [username] is the name of the new user.
- Create the home directory for the new user: mkdir /home/[username] where [username] is the name of the new user.
- Assign the new user to their home directory: usermod -d /home/[username] [username] where [username] is the name of the new user.
- Add the new user to the appropriate groups by entering the following command: usermod -aG [group1],[group2],... [username] where [group1],[group2],... are the names of the groups to be added, and [username] is the name of the new user.
- Verify that the new user has been added by entering the following command: cat /etc/passwd
- Log out of the root user by entering the following command: exit
- Log in as the new user to verify that everything has been set up correctly.

Setting the hostname

- Open terminal and log in as the root user by typing su and entering the root password
- Use the hostnamectl command to view and set the hostname:
- To view the current hostname, use hostnamectl status
- To set the hostname, use hostnamectl set-hostname <new-hostname> where <new-hostname> is the desired hostname
- Verify the changes by checking the hostname again with hostnamectl status

Network setting on VirtualBox

To configure network settings in a virtual machine running Fedora Server using VirtualBox, follow these steps:

- Open the VirtualBox application and select the virtual machine you want to configure network settings for.
- Click on the "Settings" button in the top navigation bar.
- Navigate to the "Network" section in the settings window.
- Choose the type of network adapter you want to use for your virtual machine. By default, the "NAT" option is selected, but you can also choose "Bridged Adapter," "Internal Network," or "Host-only Adapter."

Network setting on VirtualBox

When setting up a virtual machine in VirtualBox, you have the option to configure the network adapter to suit your needs. There are several options available:

- **Not attached**: The network adapter is not connected to any network, and the virtual machine will not have access to the internet or any other network resources.
- **NAT (Network Address Translation)**: The virtual machine will have access to the internet through the host machine's network connection. The virtual machine will be assigned an IP address from a virtual network, and traffic from the virtual machine will appear to come from the host machine's IP address.
- **Bridged adapter:** The virtual machine will appear as a separate device on the network and will have its own IP address. This allows the virtual machine to be accessed from other devices on the network, and for the virtual machine to access other network resources.
- **Internal network:** This option creates a virtual network between multiple virtual machines, allowing communication between them without exposing them to the physical network.
- **Host-only adapter:** This option creates a virtual network between the host machine and virtual machines, allowing communication between them without exposing them to the physical network.

Network setting on VMware

- Open VMware and start the virtual machine.
- Go to the virtual machine settings.
- Select the network adapter and click on the "Edit" button.
- Change the adapter type to the desired option (Bridged, NAT, Host-only, or Custom).
- For Bridged Networking, select the physical network adapter to use for bridging.
- For NAT, select the NAT network to use for network mapping.
- For Host-only Networking, select the host-only network to use for communication between the host and virtual machine.
- For Custom, configure the custom network settings as required.
- Save the changes and restart the virtual machine for the changes to take effect.

Network setting on VMware

VMware offers several options for adapter network settings:

- **Bridged Networking**: This option connects the virtual machine directly to the physical network, allowing it to have its own IP address and access to the network resources as if it were a physical machine.
- **NAT**: This option maps the virtual machine's IP address to the host machine's IP address and allows the virtual machine to access the internet but not the local network.
- **Host-only Networking**: This option creates a private network between the host and virtual machine, allowing communication between them but not access to the internet or other physical network resources.
- **Custom:** This option allows custom network settings to be configured, such as network settings for complex network configurations.

Network Manager using CLI

- Network management refers to the process of organizing, configuring, and maintaining computer networks to ensure they are operating efficiently and securely.
- Network management is important for ensuring the smooth functioning of networks and for troubleshooting network issues.
- Command-line interfaces (CLIs), are text-based interfaces for interacting with computer systems.
- CLIs are an effective way to manage networks, as they allow administrators to perform a wide range of tasks, from simple network configuration to complex network troubleshooting, quickly and efficiently.
- CLIs are often preferred by experienced network administrators, as they offer more control and flexibility compared to graphical user interfaces.
- Additionally, CLIs can be automated using scripts, which makes them ideal for managing large networks and performing repetitive tasks.

Network Manager Tool : nmcli

nmcli is a command-line tool for managing network connections in Fedora and other Linux distributions. It provides a simple and efficient way to configure and manage network connections and network settings. Some of the capabilities of nmcli include:

- **Configuring network interfaces**: You can use nmcli to configure various network settings, such as IP addresses, DNS settings, and network interfaces.
- **Managing network connections**: With nmcli, you can easily manage network connections, such as creating, modifying, and deleting connections, as well as controlling network connections (e.g. connecting, disconnecting, and restarting connections).
- **Displaying network information**: nmcli provides a way to view network information, such as the status of network connections and devices, network configuration, and other network-related information.

Configuring network interfaces

To configure network interfaces using nmcli, you can use the following steps:

- Identify the name of the network interface you want to configure. You can use the ***nmcli device*** command to list all available network interfaces.
- Use the **nmcli connection modify** command to modify the settings for a specific network interface. For example, to assign a static IP address to the interface eth0, you would run the following command:
  - **nmcli connection modify eth0 ipv4.addresses "192.168.1.100/24"**
- Set other network configuration options, such as the default gateway, DNS servers, using similar commands. For example, to set the default gateway for the interface eth0:
  - **nmcli connection modify eth0 ipv4.gateway "192.168.1.1"**
- Use the **nmcli connection up** command to activate the network interface and apply the changes. For example:
  - **nmcli connection up eth0**
- Verify the network configuration by using the **ip addr** or **ifconfig** command to view the IP address and other network settings for the interface.

Configuring network interfaces

To configure network interfaces using nmcli, you can use the following steps:

- Identify the name of the network interface you want to configure. You can use the ***nmcli device*** command to list all available network interfaces.
- Use the **nmcli connection modify** command to modify the settings for a specific network interface. For example, to assign a static IP address to the interface eth0, you would run the following command:
  - **nmcli connection modify eth0 ipv4.addresses "192.168.1.100/24"**
- Set other network configuration options, such as the default gateway, DNS servers, using similar commands. For example, to set the default gateway for the interface eth0:
  - **nmcli connection modify eth0 ipv4.gateway "192.168.1.1"**
- Use the **nmcli connection up** command to activate the network interface and apply the changes. For example:
  - **nmcli connection up eth0**
- Verify the network configuration by using the **ip addr** or **ifconfig** command to view the IP address and other network settings for the interface.

Q&A

Q&A

- What is the difference between open source and free?

➔Open source refers to the license of the software, while free refers to the price. Open source software is software that is released with a license that allows the source code to be freely available to the public, allowing anyone to use, modify, or distribute the software. Free software, on the other hand, refers to software that is available at no cost to the user.

Q&A

- What is the difference between Linux and other operating systems like Windows and Mac OS?

➔Linux is different from Windows and Mac OS in that it is open source and freely available to the public. It also has a different architecture and design philosophy, with a focus on customization and flexibility. Additionally, Linux is known for its stability and security.

Q&A

- What is a shell in Linux?

➔A shell in Linux is a command-line interface that allows you to interact with the operating system and run various commands and programs.

Q&A

- What is package management in Linux?

➔Package management in Linux is a system for managing the installation, removal, and updating of software packages on a Linux system. Package management systems typically use a centralized repository of software packages, making it easy to install and manage software.

.

Q&A

- What is virtualization?

➔Virtualization is the process of creating a virtual version of a physical environment, such as an operating system, a server, or a storage device. This allows multiple virtual environments to run on a single physical machine, increasing the utilization of hardware resources and providing greater flexibility in managing resources.

.

Practice

Practice

1. Bring your Laptop.
2. Download the latest version of Fedora server (Fedora 37)
    1. https://getfedora.org/en/server/download/
3. Choose to work between VirtualBox and VMware,
4. Install Fedora Server
5. Install the GUI
6. Configure your network and choose NAT in order to have internet in your virtual machine
7. Try to ping www.google.com
8. Clone your virtual machine