PROJECT REPORT
ON

“Downloading Fedora Linux within Red Hat Linux”

Submitted By:

Ashutosh Katoch,

UID- 24MCC20043

Under The Guidance of:

Mr. Rishabh Tomar

Oct, 2024

University Institute of Computing

Chandigarh University,

Mohali, Punjab

CERTIFICATE

This is to certify that Ashutosh Katoch (UID- 24MCC20043) have successfully completed the minor project title **"Downloading Fedora Linux within Red Hat Linux"** at University Institute of Computing under my supervision and guidance in the fulfilment of requirements of Ist semester, **Master of Computer Application-Specialization in Cloud Computing and DevOps.** Of Chandigarh University, Mohali, Punjab.

Dr. Abdullah	Mr. Rishabh Tomar
Head of the Department	Project Guide Supervisor
University Institute of Computing	University Institute of Computing

ACKNOWLEDGEMENT

We deem it a pleasure to acknowledge our sense of gratitude to our project guide Mr. Rishabh Tomar under whom we have carried out the project work. His incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish to reciprocate in full measure the kindness shown by Dr. Abdullah (H.O.D, University Institute of Computing) who inspired us with his valuable suggestions in successfully completing the project work.

We shall remain grateful to Dr. Manisha Malhotra, Additional Director, University Institute of Technology, for providing us a strong academic atmosphere by enforcing strict discipline to do the project work with utmost concentration and dedication.

Finally, we must say that no height is ever achieved without some sacrifices made at some end and it is here where we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.

Date: 21.10.2024

Place: Chandigarh University, Mohali, Punjab

Ashutosh Katoch,
UID- 24MCC20043

ABSTRACT

Virtualization has revolutionized the computing landscape by allowing multiple operating systems to coexist on a single physical machine, enhancing resource efficiency, flexibility, and scalability. It has become an essential component in modern IT infrastructure, enabling developers, system administrators, and organizations to create isolated environments for testing, development, and production purposes without the need for additional hardware. This project report focuses on the process of downloading and installing the Fedora Linux distribution inside a Red Hat Linux operating system using **KVM (Kernel-based Virtual Machine)**, a widely adopted open-source hypervisor for Linux-based systems.

The project details the step-by-step process of configuring KVM on Red Hat Linux, followed by the installation of Fedora Linux as a guest operating system, utilizing Virt-Manager to simplify the management of virtual machines. It covers essential tasks such as resource allocation, virtual disk and storage configuration, network setup, and post-installation management, providing a comprehensive guide for users seeking to virtualize Fedora within a Red Hat environment.

Throughout the report, we emphasize the advantages of virtualizing Fedora within Red Hat, such as the ability to experiment with Fedora's cutting-edge technologies while maintaining the stability of Red Hat's enterprise-grade environment. Additionally, the report addresses key challenges encountered during the process, including network bridging, BIOS settings for virtualization, disk allocation, and performance optimization, offering solutions to mitigate these issues.

Moreover, the project investigates potential future applications of virtualization, such as integrating containerization technologies (like Docker or Podman) within virtualized environments, automating VM management through scripting or tools like Ansible, and exploring advanced network configurations for enhanced security and scalability. This report also highlights opportunities for improving virtualization technology, focusing on areas like resource optimization, performance tuning, and advanced virtualization features like nested virtualization.

TABLE OF CONTENTS

Introduction

In today's rapidly advancing technological landscape, virtualization plays a crucial role in enhancing the flexibility, scalability, and efficiency of computing environments. Virtualization allows users to run multiple operating systems on a single machine, facilitating testing, development, and deployment in isolated, controlled environments. This project, titled **"Downloading Fedora Linux Distribution Inside Red Hat Linux,"** aims to explore the process of virtualizing Fedora within a Red Hat Enterprise Linux (RHEL) environment using KVM-based virtualization. By creating a Fedora Virtual Machine (VM) inside Red Hat Linux, we can leverage the benefits of both distributions simultaneously, without compromising the stability of the host OS.

Red Hat Enterprise Linux is renowned for its enterprise-grade stability, long-term support, and secure production environment, whereas Fedora is the cutting-edge, rapidly evolving distribution known for introducing new features and technologies. Virtualizing Fedora within RHEL allows developers and IT professionals to experiment with Fedora's latest innovations while maintaining Red Hat's reliability. This project delves into the steps necessary to install Fedora within Red Hat using **KVM (Kernel-based Virtual Machine)** and **Virt-Manager**, a popular VM management tool for Linux systems.

In this project, we will explore various aspects of virtualization, including preparing the Red Hat system for VM creation, configuring resources for optimal performance, installing and configuring Fedora Minimal, and addressing challenges such as network configuration, disk allocation, and performance tuning. We will also examine post-installation setups, such as installing optional graphical environments (e.g., XFCE) and managing the Fedora VM using Virt-Manager.

This project not only demonstrates the process of setting up a virtualized environment but also explores the potential for optimizing resource utilization and improving system performance. Moreover, it serves as a practical guide for Linux users, developers, and IT professionals interested in using multiple distributions on a single machine while maintaining a secure and isolated environment for testing and development purposes.

1.1 Background

In the evolving landscape of modern computing, virtualization has emerged as a key technology driving innovation, efficiency, and resource optimization. Virtualization allows a single physical machine to run multiple operating systems concurrently, enabling better utilization of hardware resources, isolation of workloads, and flexibility in managing different environments. This capability has led to the rise of virtual machines (VMs), which can be created, configured, and destroyed easily without affecting the underlying hardware. Virtualization is widely used in data centers, cloud computing, and personal computing, providing cost-effective solutions for businesses and individuals alike.

Red Hat Linux, known for its stability, security, and enterprise-level support, is a leading distribution in the Linux ecosystem. It has been the backbone for many organizations due to its robust features and long-term support options. Alongside Red Hat Linux, Fedora has gained popularity as a cutting-edge, community-driven distribution that acts as an upstream source for Red Hat Enterprise Linux (RHEL). Fedora offers the latest in open-source software and technology, making it an attractive choice for developers, system administrators, and enthusiasts who seek to work with the newest Linux features.

This project focuses on installing **Fedora Linux** within **Red Hat Linux** using virtualization techniques, specifically through **KVM (Kernel-based Virtual Machine)**, a powerful open-source hypervisor. By doing so, users can experience the features of Fedora without needing to dual-boot or replace their existing Red Hat Linux environment. The installation and configuration process provides insight into how virtual machines operate, how resources are allocated, and the benefits of having multiple operating systems on a single machine for testing, development, and educational purposes.

Virtualizing Fedora within Red Hat offers numerous advantages, including creating isolated environments for different projects, testing new software without risking system stability, and learning more about Linux systems. For system administrators and developers, this is an invaluable tool to experiment with new technologies in a safe and controlled environment.

1.2 Purpose of the Project

The primary purpose of this project is to provide a comprehensive, step-by-step guide to downloading and installing Fedora Linux inside a Red Hat Linux environment using KVM. The project aims to explore virtualization technology as a tool for optimizing system resources, isolating workloads, and simplifying the management of multiple operating systems on a single machine. More specifically, the goals of this project include:

1. **Exploration of Virtualization Concepts**: The project seeks to explain the concepts of virtualization and how it applies to modern computing. By using KVM, users will gain a better understanding of how hypervisors work, how virtual machines are created, and how system resources like memory, CPU, and storage are allocated within a virtualized environment.

2. **Hands-On Learning**: This project provides users with practical, hands-on experience in configuring virtualization on a Red Hat Linux system. The guide details the entire process of setting up KVM, creating a virtual machine for Fedora, configuring storage and network settings, and optimizing the virtual machine's performance. By following the project, users will not only learn about the technical aspects of virtualization but also enhance their problem-solving skills.

3. **Use Case for Developers and System Administrators**: For developers, running Fedora inside Red Hat Linux allows for testing software across different Linux distributions without requiring additional physical machines. System administrators can use this approach to simulate different environments, perform software testing, or learn new tools without affecting the production system. The project demonstrates how to set up such a versatile environment efficiently.

4. **Performance and Resource Optimization**: Another key objective of this project is to show how virtual machines can be optimized to ensure that system resources are used efficiently. Fedora is installed with minimal resource consumption, and configuration techniques will be demonstrated to optimize the performance of the VM on Red Hat Linux. By doing so, users can run multiple virtual machines without overloading their host system, even on machines with limited resources.

5. **Future Scalability and Flexibility**: As virtualization becomes more prevalent in both personal and enterprise IT, understanding how to set up and manage virtual machines is a valuable skill. This project lays the groundwork for more complex virtualization scenarios, such as managing multiple VMs for different operating systems, automating VM creation, or even migrating VMs across different hosts. By mastering the basic steps of installing Fedora inside Red Hat, users will be prepared to expand their knowledge into more advanced virtualization topics.

6. **Compatibility and Experimentation**: Fedora is known for including the latest open-source software and technology, making it a perfect environment for experimentation. Users can leverage the power of virtualization to test new Fedora releases, explore the latest kernel features, or try out different desktop environments without disrupting their stable Red Hat system. This project serves as a safe platform for such experimentation, ensuring that users can easily revert or discard changes in a virtual machine if something goes wrong.

7. **Educational and Research Applications**: This project is also highly beneficial for educational purposes. Students, researchers, and IT professionals can use virtual machines to study operating systems, networking, or software development within an isolated environment. Fedora, being an up-to-date Linux distribution, provides the latest development tools and software packages, making it an ideal system for learning and experimentation.

1.3 Objectives

This report aims to:

- Explain how virtualization works and how to set up a virtual environment in Red Hat Linux.
- Provide a detailed, step-by-step guide to downloading and installing Fedora Linux inside Red Hat Linux using KVM and Virt-Manager.
- Troubleshoot common challenges and provide solutions to optimize performance.
- Highlight the benefits of running Fedora in a virtualized environment within Red Hat.

1.4 Structure of the Report

This report is structured as follows:

- Introduction: Overview of the project and its objectives.
- Virtualization and Linux Distributions: Introduction to virtualization and the Linux distributions involved.
- Tools and Setup Requirements: Description of the hardware and software requirements for the project.
- Downloading Fedora ISO: Instructions for obtaining Fedora Linux.
- Installing Fedora in KVM: Step-by-step process of creating and configuring a Fedora VM.
- Post-Installation Configuration: Steps to configure the Fedora virtual machine after installation.
- Challenges and Solutions: A discussion of common issues and how to resolve them.
- Conclusion and Future Scope: Summary of the project and recommendations for future work.
- References: List of resources and references used in the report.
- Plagiarism Report: Document proving the originality of the content.

2. Virtualization and Linux Distributions

2.1 Overview of Virtualization

Virtualization allows multiple operating systems to run on a single physical machine by abstracting the underlying hardware. By creating virtual machines (VMs), each operating system operates as though it has dedicated hardware. This technology is widely used in data centers and enterprise environments to optimize hardware utilization, improve scalability, and ensure that workloads remain isolated from one another.

There are several types of virtualization, including:

- Full Virtualization: The entire hardware is simulated, allowing unmodified guest operating systems to run.
- Paravirtualization: The guest OS is modified to work with a hypervisor, improving performance by eliminating certain overhead.
- Containerization: Instead of virtualizing an entire operating system, only the

application and its dependencies are isolated.

The hypervisor is the core component of virtualization, responsible for managing VMs. For this project, we use KVM (Kernel-based Virtual Machine), a Linux-based hypervisor built into the kernel, which makes it highly efficient for virtualizing Linux distributions.

2.2 Fedora Linux Overview

Fedora is a free, open-source Linux distribution developed by the Fedora Project, which is sponsored by Red Hat. Fedora is known for its focus on innovation, regularly adopting new technologies that may later be incorporated into Red Hat Enterprise Linux. It provides multiple versions tailored to different needs, including Fedora Workstation (for desktop use) and Fedora Server (for server environments).

Fedora is highly regarded for:

- Rapid release cycles, providing the latest features and software.
- A community-driven development process.
- A commitment to open-source principles and software.

2.3 Red Hat Linux Overview

Red Hat Linux holds a significant place in the history of open-source software and Linux distributions. Founded in 1994 by **Marc Ewing** and **Bob Young**, Red Hat Linux quickly became one of the most popular and influential Linux distributions. It was designed to provide both stability and cutting-edge features for businesses, developers, and enthusiasts alike, and became known for its commitment to open-source principles.

Red Hat Linux was the foundation for what would later become **Red Hat Enterprise Linux (RHEL)**, the flagship enterprise distribution from Red Hat, Inc. Over time, Red Hat Linux evolved from a community-focused operating system intoa robust, commercial-grade product designed to meet the needs of enterprise IT environments. This overview traces the history, features, and influence of Red Hat Linux, examining its journey from an open-source community project to a powerhouse in the enterprise world, shaping the future of Linux distributions globally.

3. Tools and Setup Requirements

3.1 Hardware and Software Requirements

To successfully install Fedora inside Red Hat Linux, the following requirements should be met:

Hardware Requirements:

- CPU: A processor that supports virtualization (e.g., AMD Ryzen 5500 or Intel equivalent).
- RAM: At least 8GB of memory (with 2GB or more dedicated to the Fedora VM).
- Storage: At least 512GB of available space, with 20GB or more reserved for the Fedora VM.
- Network: A stable internet connection for downloading Fedora ISO and required software packages.

Software Requirements:

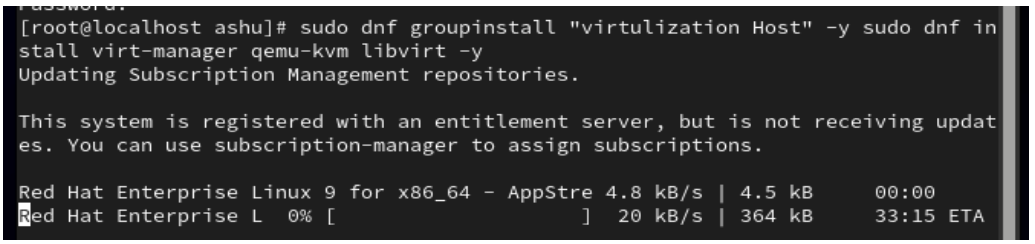
- Host OS: Red Hat Enterprise Linux 8 or later.
- Fedora ISO: The latest version of Fedora Workstation or Fedora Minimal.
- KVM: Kernel-based Virtual Machine, a hypervisor built into the Linux kernel.
- Virt-Manager: A graphical tool for managing virtual machines.
- QEMU: A machine emulator and virtualizer that works with KVM to virtualize hardware.

3.2 Installing and Configuring KVM on Red Hat Linux

KVM, along with additional virtualization tools, must be installed and configured on Red Hat to support the creation and management of virtual machines.

Step 1: Install Virtualization Packages

Use the dnf package manager to install the necessary virtualization tools:

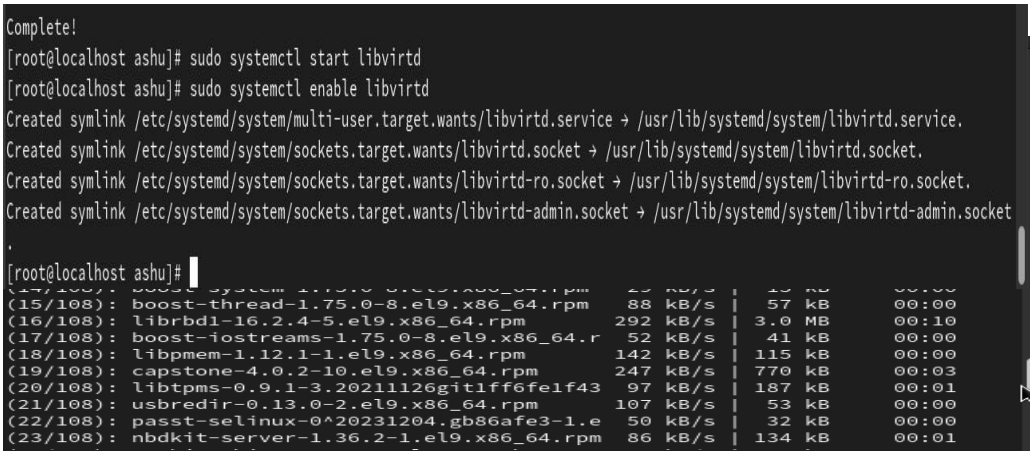


sudo dnf groupinstall "Virtualization Host" -y

sudo dnf install virt-manager qemu-kvm libvirt libvirt-python virt-install virt-viewer -y

Step 2: Enable and Start libvirtd Service

Start and enable the libvirtd service, which manages virtual machines on Red Hat Linux:

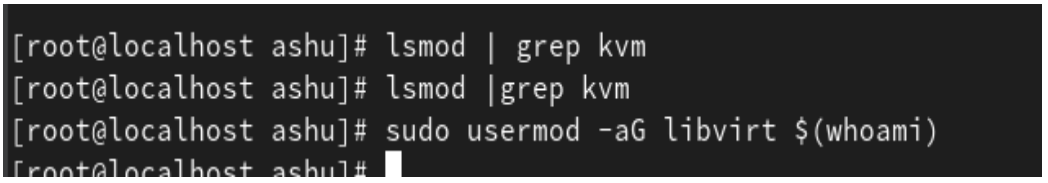


sudo systemctl start libvirtd

sudo systemctl enable libvirtd

Step 3: Verify KVM Installation

Ensure that KVM is properly installed by running the following command:

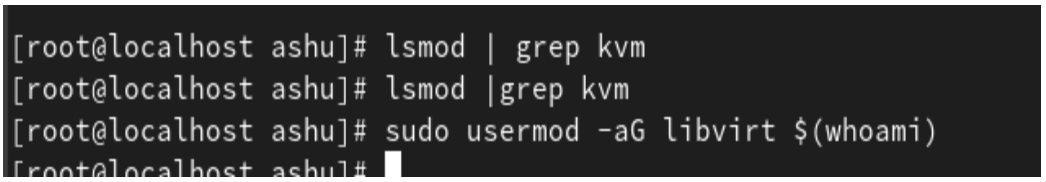


lsmod | grep kvm

This should display kvm_intel or kvm_amd depending on your processor.

Step 4: Adding User to libvirt Group

To manage virtual machines without needing superuser privileges, add your user to the libvirt group:



sudo usermod -aG libvirt $(whoami)

newgrp libvirt

4. Downloading Fedora ISO

4.1 Choosing the Fedora Version.

4.2 Downloading Fedora Workstation and Minimal Versions

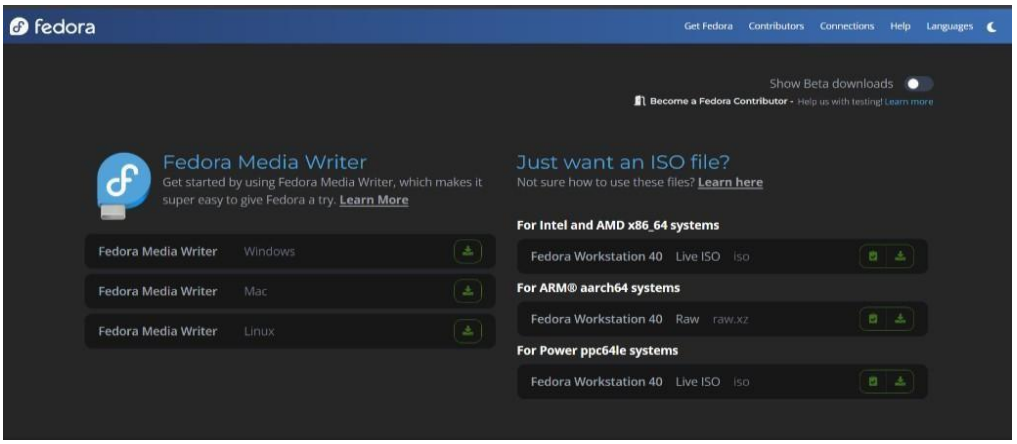
To download Fedora, follow these steps:

Step 1: Visit the Official Fedora Website

Navigate to getfedora.org to download the Fedora ISO.

Step 2: Select the Version

Choose either the Fedora Workstation or Fedora Minimal version based on your requirements. Download the ISO file to your system.



Step 3: Verify the ISO

After downloading the ISO, verify its integrity by comparing the checksum with the one provided on the Fedora website:

sha256sum /path/to/fedora.iso

The output should match the checksum on the Fedora website.

5. Installation of Fedora Linux inside Red Hat Linux

This section outlines the detailed steps involved in installing Fedora Minimal inside Red Hat Linux using the Virt-Manager tool.

5.1. Preparing Virt-Manager on Red Hat

Before setting up the Fedora VM, you need to ensure that your Red Hat environment has virtualization enabled and that **Virt-Manager** is installed. Follow these steps:

Step 1: Install Virtualization Packages

First, install the virtualization packages including Virt-Manager and KVM:

```
[root@localhost ashu]# sudo dnf install virt-manager libvirt qemu-kvm
Updating Subscription Management repositories.
Last metadata expiration check: 0:24:24 ago on Mon 21 Oct 2024 12:08:35 AM IST.
Package virt-manager-4.1.0-5.el9.noarch is already installed.
Package libvirt-10.0.0-6.7.el9_4.x86_64 is already installed.
Package qemu-kvm-17:8.2.0-11.el9_4.6.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

sudo dnf install virt-manager libvirt qemu-kvm

Ensure the **libvirtd** service is enabled and running:

```
Complete!
[root@localhost ashu]# sudo systemctl start libvirtd
[root@localhost ashu]# sudo systemctl enable libvirtd
Created symlink /etc/systemd/system/multi-user.target.wants/libvirtd.service → /usr/lib/systemd/system/libvirtd.service.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd.socket → /usr/lib/systemd/system/libvirtd.socket.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd-ro.socket → /usr/lib/systemd/system/libvirtd-ro.socket.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd-admin.socket → /usr/lib/systemd/system/libvirtd-admin.socket
.
[root@localhost ashu]#
```

sudo systemctl enable libvirtd

sudo systemctl start libvirtd

Step 2: Verify Virtualization Support

Check if your system supports virtualization with the following command:

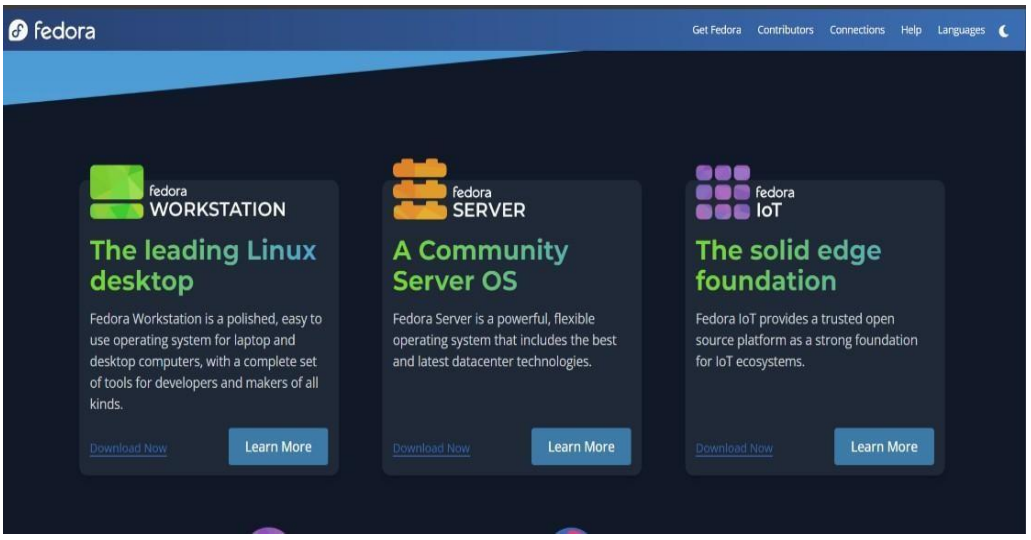
```
[root@localhost ashu]# lscpu | grep virtualization
```

lscpu | grep Virtualization

Ensure that **VT-x** (Intel) or **AMD-V** is enabled.

Step 3: Download Fedora ISO

Visit the Fedora website and download the latest **Fedora Minimal ISO**. This lightweight version of Fedora will save on resources.



5.2 Allocating Resources for Fedora VM

Once you have selected the Fedora ISO in Virt-Manager, you will need to allocate resources for the virtual machine. It is essential to ensure that the Fedora VM has enough CPU, memory, and disk space for smooth operation without over-allocating resources from the Red Hat host.

Step 1: Allocate CPU Cores

- CPU Cores: Choose at least 2 cores for the Fedora VM to ensure decent performance. Depending on your host system's CPU, you can allocate more cores if required.

Step 2: Allocate Memory

- RAM: For Fedora Minimal, at least 1GB of RAM is recommended. For Fedora Workstation (if using a GUI), allocate a minimum of 2GB or more. Avoid allocating more than half of the total available memory to the VM, as this may impact the host system's performance.

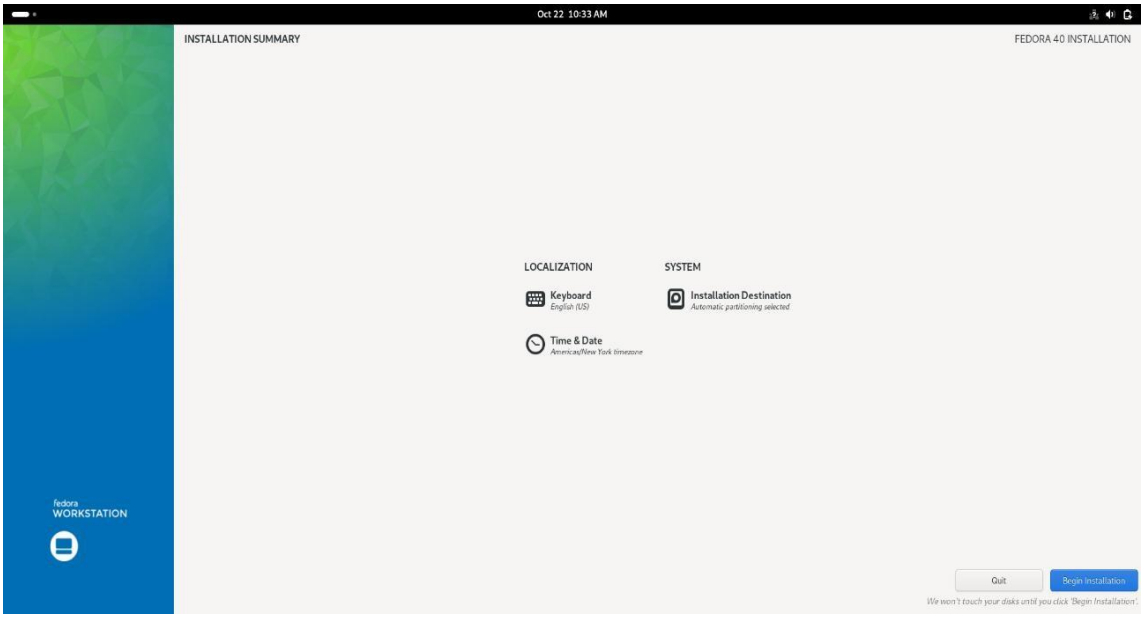
Step 3: Set Virtual Disk Size

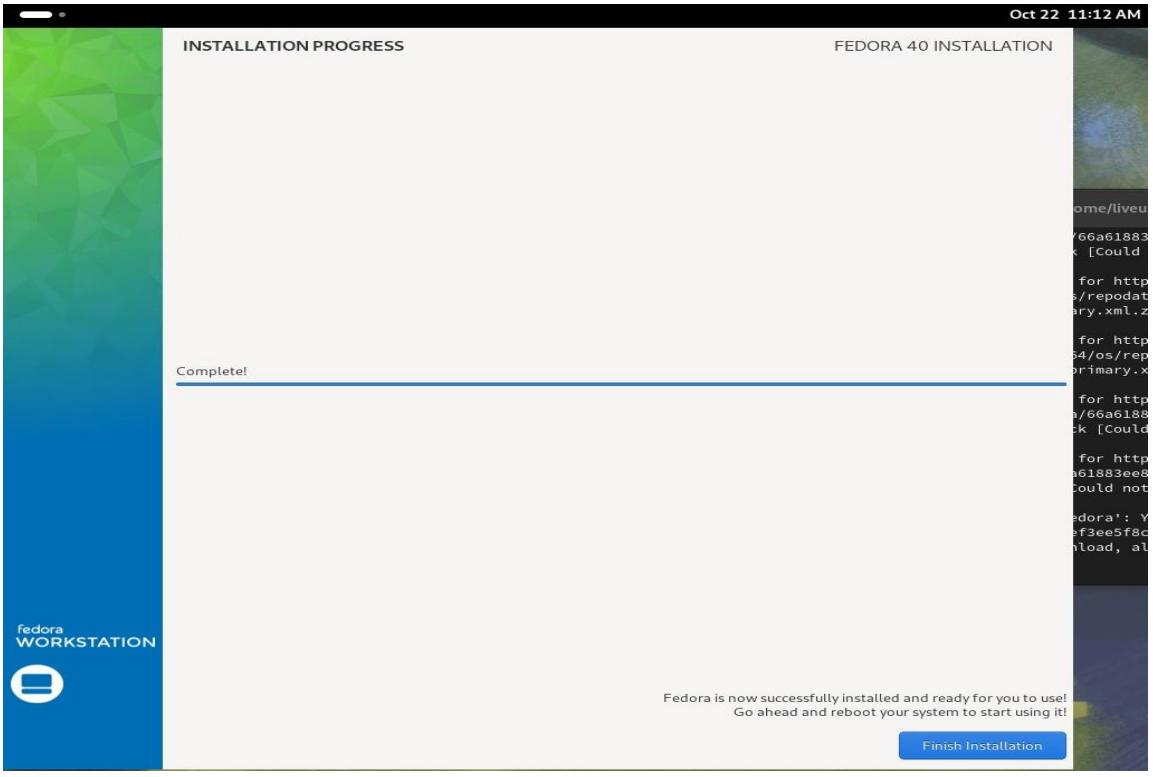
- Disk Space: For Fedora Minimal, allocate 10GB to 15GB of storage. If you're installing Fedora Workstation with a graphical user interface (GUI), it's best to allocate 20GB or more.

5.3 Installing Fedora Minimal

Step 1: Begin Fedora Installation

After allocating resources, click on Begin Installation in Virt-Manager. Fedora Minimal will boot from the ISO, and the installation process will start.





Step 2: Select Installation Destination

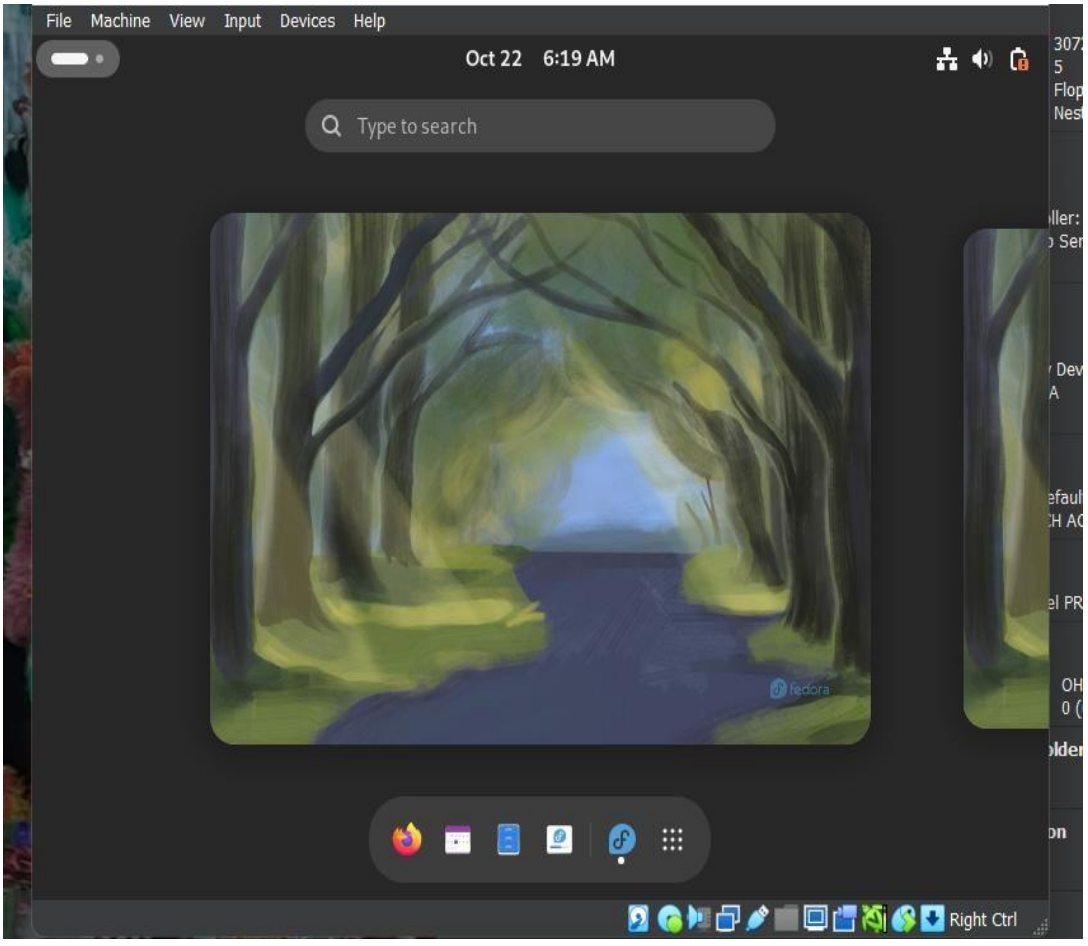
Choose the virtual disk you allocated during the setup. Ensure that the disk is correctly configured, then click Done to proceed

Step 3: Set Up User Accounts

During the installation, set up the root password and create a user account. This user will have administrative privileges, so choose a strong password.

Step 4: Complete Installation

Once all configurations are made, Fedora will begin installing. The process should take a few minutes, depending on your system's performance. After completion, reboot the VM to launch Fedora Minimal.



5.4 Configuring Virtual Disks and Storage

Managing virtual storage is essential for performance and proper resource allocation. Here's how to configure the disk space optimally:

Step 1: Use a Thin-Provisioned Disk

Virt-Manager allows you to allocate storage dynamically with thin provisioning. This means the VM will only use the disk space it needs, up to the allocated limit. It's a more efficient way to manage storage on the host system.

Step 2: Disk Caching

Enable disk caching options in Virt-Manager to improve the performance of read/write operations on the virtual disk. This will increase the overall responsiveness of the Fedora VM.

5.5 Network Configuration

Networking is crucial for connecting the Fedora VM to the internet or the local network.

Step 1: Choose Network Mode

There are two common network modes to choose from:

- NAT (Network Address Translation): Provides internet access to the VM through the host's network interface. It is simple to set up and works out of the box.
- Bridge Mode: Allows the VM to appear as another device on the network, with its own IP address. This is useful for more complex networking setups.

Step 2: Configure Network Interface

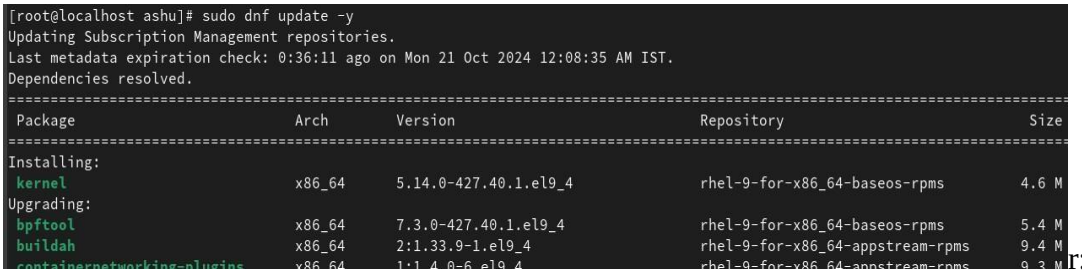
After selecting the network mode, make sure the Fedora VM has a working network interface. You can manually configure IP addresses or use DHCP to automatically assign one.

6. Post-Installation Configuration

6.1 Initial Setup of Fedora Minimal

Once Fedora Minimal is installed and running, you may need to perform initial configuration tasks such as updating the system and installing essential packages. Step 1: Update System Packages

Update Fedora using the dnf package manage



r:

sudo dnf update -y

Step 2: Install Basic Utilities

You can install basic packages such as wget, vim, and net-tools:

```
[root@localhost ashu]# sudo dnf install wget vim net-tools -y
Updating Subscription Management repositories.
Last metadata expiration check: 0:38:52 ago on Mon 21 Oct 2024 12:08:35 AM IST.
Package wget-1.21.1-8.el9_4.x86_64 is already installed.
Package vim-enhanced-2:8.2.2637-20.el9_1.x86_64 is already installed.
Package net-tools-2.0-0.62.20160912git.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

sudo dnf install wget vim net-tools -y

7. Challenges and Solutions in Virtualizing Fedora on Red Hat Linux with KVM

Virtualization can pose various challenges, especially when configuring Fedora within Red Hat Linux using KVM. Below are common issues and their solutions:

7.1 BIOS Virtualization Settings

Challenge: Virtualization must be enabled in BIOS (Intel VT-x/AMD-V), or KVM won't work. Solution: Access BIOS/UEFI, enable virtualization, and reboot. Then, install and configure KVM.

7.2 Network Configuration

Challenge: Improper network bridging can block VM access to the network. Solution: Use virt-manager to configure a bridged network, or manually set up a bridge using nmcli to ensure VMs can access the network.

7.3 Disk Allocation

Challenge: Insufficient disk space or misconfigured storage can lead to performance issues. Solution: Use logical volume management (LVM) for flexible storage and allocate adequate disk space during VM creation.

7.4 Performance Optimization

Challenge: VMs may experience lag due to limited resources. Solution: Allocate adequate CPU and RAM, enable hardware virtualization (Intel VT-x/AMD-V), and optimize storage performance with SSDs and virtio drivers.

7.5 Managing Multiple VMs

Challenge: Resource contention occurs when running multiple VMs. Solution: Use cgroups to limit resource usage, prioritize critical VMs, and dynamically adjust

resources. Monitor performance with tools like virt-top.

8. Conclusion and Future Scope

The virtualization of Fedora Linux within a Red Hat Linux environment using KVM represents a significant advancement in how users can manage and utilize multiple operating systems on a single machine. This project has highlighted the myriad benefits associated with virtualization, including resource optimization, improved testing environments, and enhanced educational opportunities. By allowing users to run Fedora in a controlled virtual environment, we can explore the unique features and applications of this popular Linux distribution without impacting the stability of the host operating system.

Final Thoughts

In conclusion, the project of virtualizing Fedora Linux within a Red Hat Linux environment using KVM has provided valuable insights into the practical applications of virtualization technology. By embracing virtualization, users can leverage the strengths of multiple operating systems, foster learning and experimentation, and ultimately enhance productivity in various computing environments. As technology continues to advance, the potential for virtualization to play an increasingly critical role in IT infrastructure and software development remains vast, inviting ongoing exploration and innovation in this dynamic field.

9. References

- Fedora Project. (n.d.). Get Fedora. Retrieved from https://getfedora.org
- Red Hat. (n.d.). Red Hat Enterprise Linux Documentation. Retrieved from-https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/
- KVM (Kernel-based Virtual Machine). (n.d.). Documentation. Retrieved from **https://www.linux-**kvm.org/page/Main_Page
- Virt-Manager Documentation. (n.d.). Retrieved from https://virt-manager.org