Step-by-Step Installation Guide

Step 1: Update Your System

First, update your package database to ensure all existing packages are up-to-date:

```bash
sudo apt update
```

Step 2: Install Prerequisite Packages

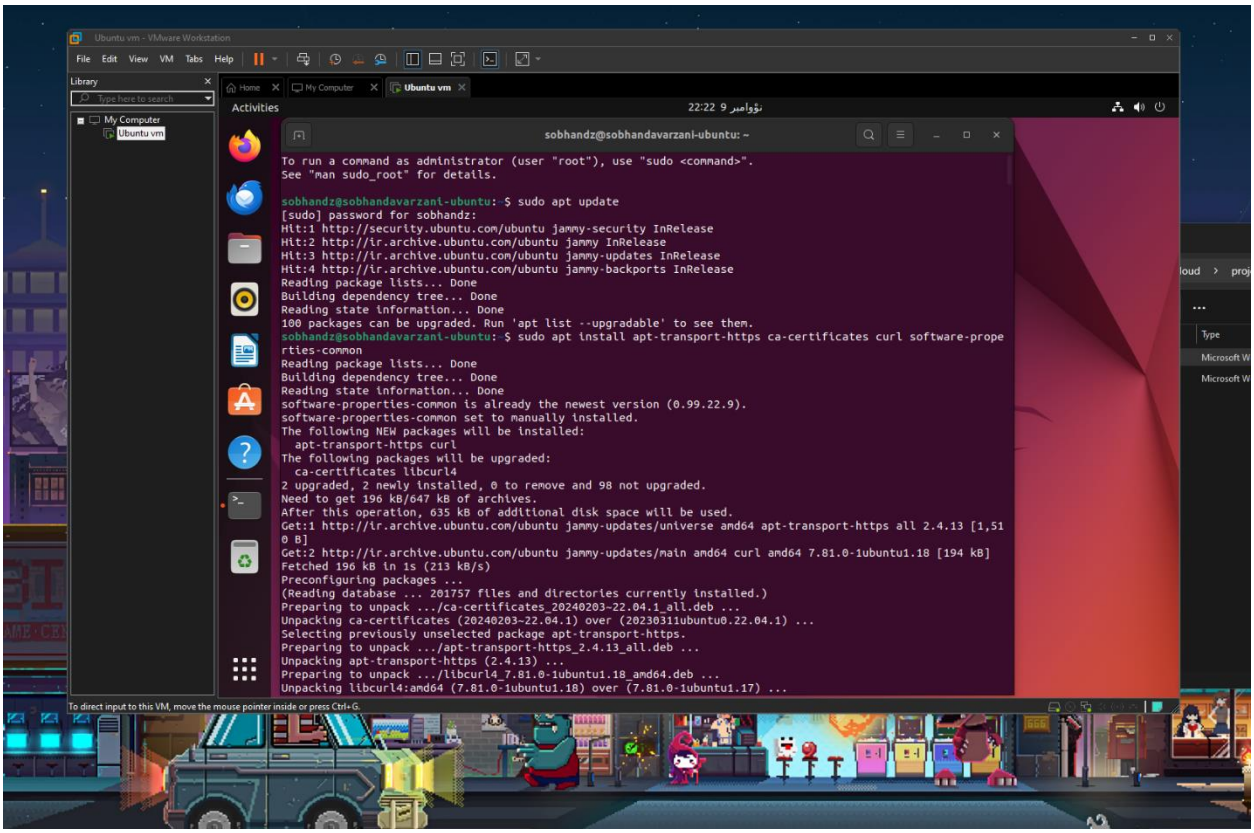
Install packages that allow `apt` to access repositories over HTTPS:

```bash
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Step 3: Add Docker’s Official GPG Key

Docker packages are signed, so add Docker's GPG key to verify the package’s integrity:

```bash
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```



Step 4: Add Docker’s APT Repository

Add the Docker repository to `apt` sources, allowing your system to install Docker packages from Docker’s official repository:

```bash

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

Step 5: Update Package Database

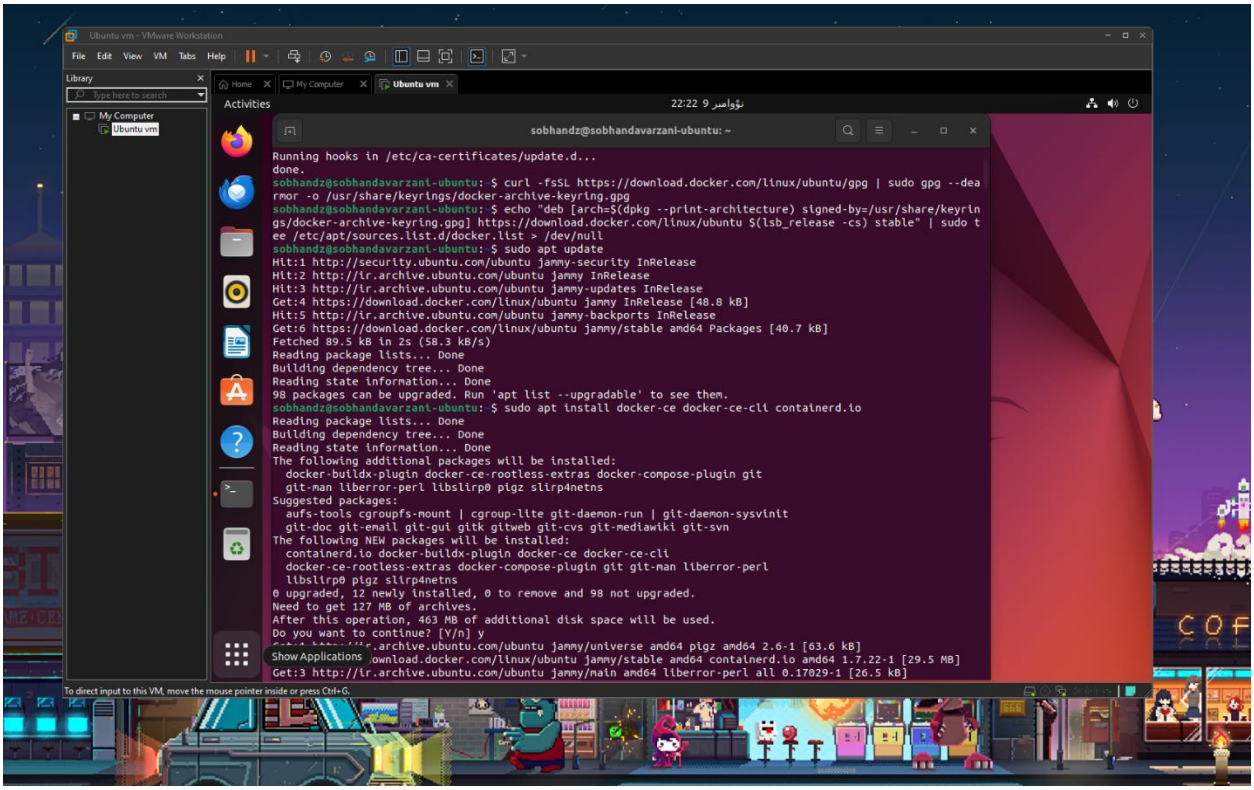
After adding the Docker repository, update your package database to include Docker packages:

```bash
sudo apt update
```

Step 6: Install Docker Engine

Now, install Docker using the following command:

```bash
sudo apt install docker-ce docker-ce-cli containerd.io
```



- **docker-ce**: Docker Community Edition

- **docker-ce-cli**: Command-line interface for Docker

- **containerd.io**: Container runtime

Step 7: Start and Enable Docker

Enable Docker to start on boot and start the Docker service:

```bash
sudo systemctl enable docker
sudo systemctl start docker
```

Step 8: Verify the Installation

To check if Docker is installed and running correctly, run:

```bash
docker --version
```

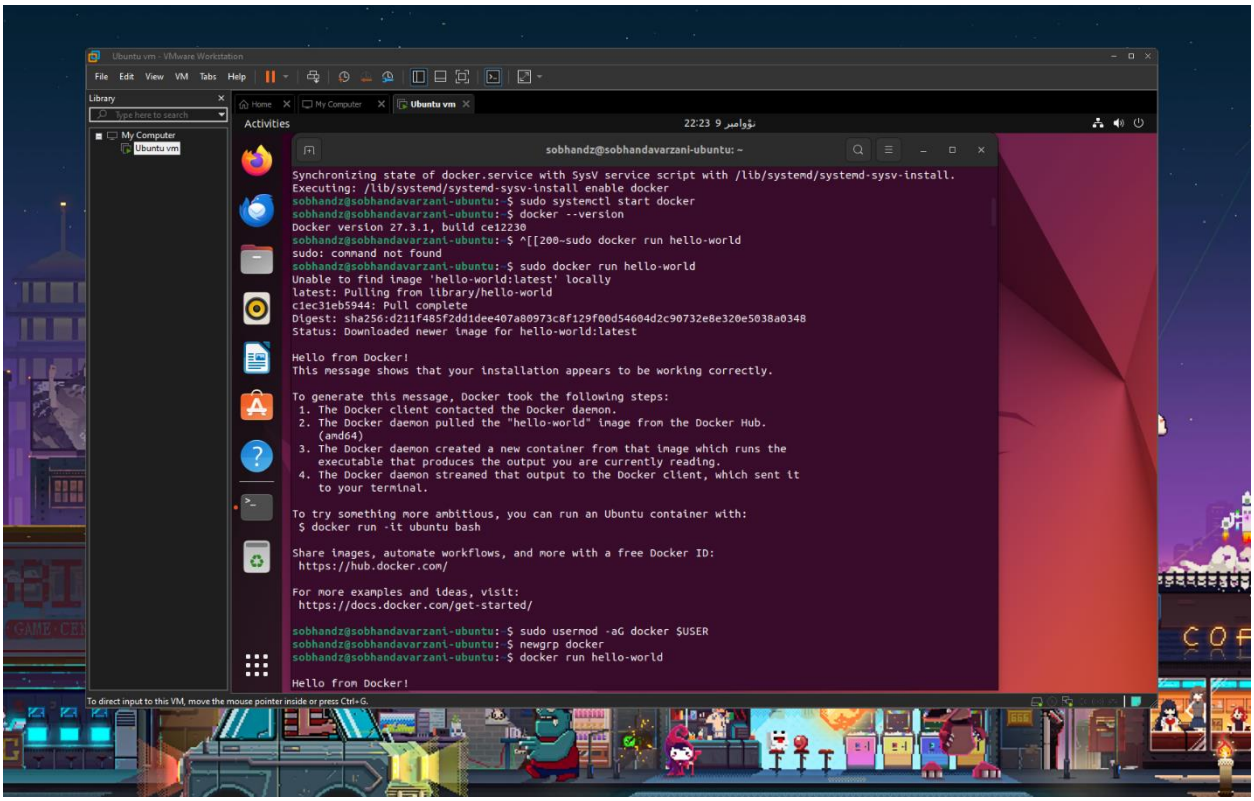
This command should display the version of Docker installed on your system.

Step 9: Test Docker Installation

Run a simple test image to ensure Docker is working:

```bash
sudo docker run hello-world
```

This command downloads a small test image and runs it in a container, printing a "Hello from Docker!" message if everything is set up correctly.



Post-Installation Steps

1. Manage Docker as a Non-Root User

By default, Docker requires root privileges. To run Docker without `sudo`, add your user to the `docker` group.

Step 1: Add Your User to the Docker Group

Replace `$USER` with your username, or use `$USER` to refer to the currently logged-in user:

```bash
sudo usermod -aG docker $USER
```

Step 2: Apply Group Changes

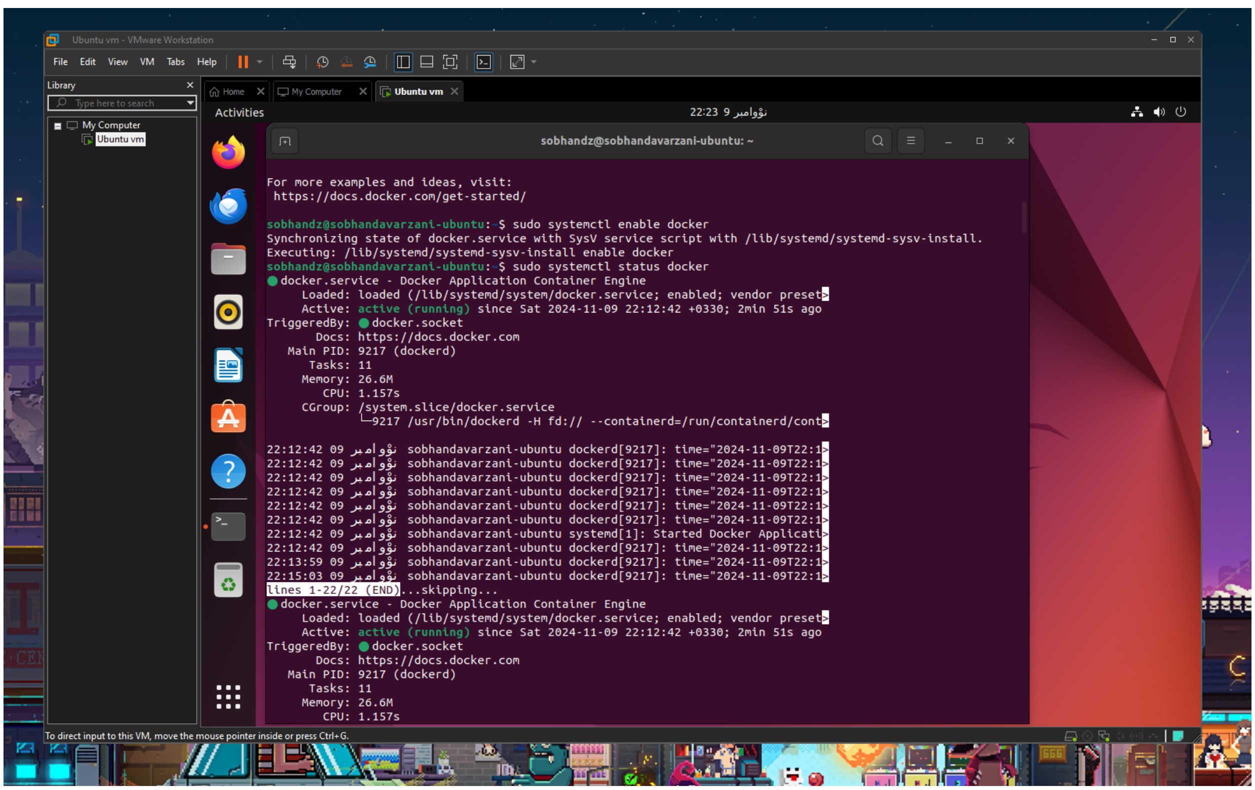
Log out and back in to refresh group membership, or run the following to apply group changes without logging out:

```bash
newgrp docker
```

Step 3: Verify Non-Root Access

After joining the Docker group, test if Docker can be used without `sudo`:

```bash
docker run hello-world
```



2. Enable Docker to Start on Boot

If Docker is not set to start automatically on boot, enable it manually:

```bash
sudo systemctl enable docker
```