Step 2: Docker Engine Setup using Package Manager

This document covers the installation of Docker Engine on an Ubuntu Virtual Machine using Docker’s apt package manager repository.

## 1. Remove Older Versions (if any)

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| sudo apt-get remove docker docker-engine docker.io containerd runc | Uninstalls any older versions of Docker or related components to avoid conflicts. |
| ls -l /var/lib/docker | Checks if Docker was previously installed (directory exists). If it shows 'No such file or directory', Docker isn’t installed yet. |

## 2. Update apt Index

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| sudo apt-get update | Updates the apt package index with the latest package information. |

## 3. Install Prerequisite Packages

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| sudo apt-get install ca-certificates curl gnupg lsb-release | Installs necessary packages: • ca-certificates → validates HTTPS connections • curl → downloads files from URLs • gnupg → manages encryption keys • lsb-release → identifies Ubuntu distribution details |

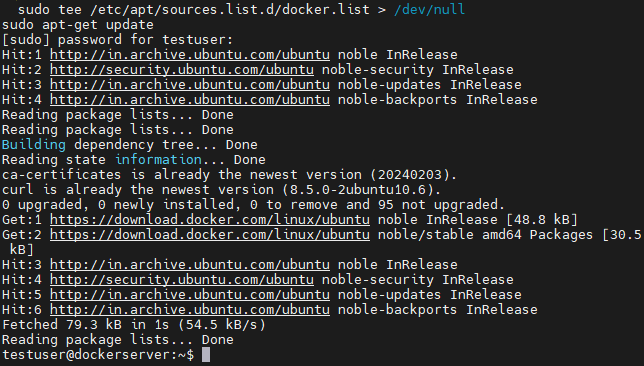
## 4. Add Docker’s Official GPG Key

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| sudo install -m 0755 -d /etc/apt/keyrings | Creates a secure directory for storing repository keys. |
| sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc | Downloads Docker’s official GPG key and saves it in the keyrings folder. |
| sudo chmod a+r /etc/apt/keyrings/docker.asc | Sets read permissions for all users so apt can use the key. |

## 5. Add Docker Repository

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "${UBUNTU\_CODENAME:-$VERSION\_CODENAME}") stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null | Adds Docker’s official repository to apt sources. • Detects architecture (dpkg --print-architecture) • Uses Docker’s GPG key for package signing • Detects Ubuntu codename (jammy, focal, etc.) • Saves repository entry in /etc/apt/sources.list.d/docker.list |
| sudo apt-get update | Updates apt index to include Docker’s repository packages. |

## 

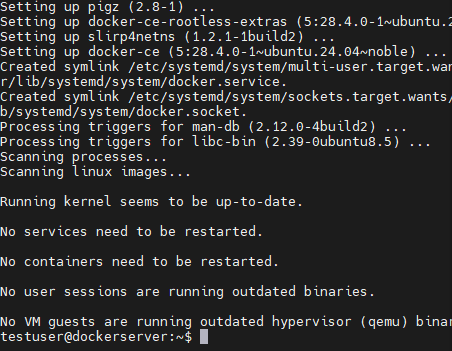


## 6. Install Docker Engine

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin | Installs Docker Engine and related components: • docker-ce → Docker Engine (Community Edition) • docker-ce-cli → Docker Command-Line Client • containerd.io → container runtime • docker-compose-plugin → enables Docker Compose v2 |

## 

## 



## 7. Verify Installation

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| docker --version | Displays the installed Docker version. |
| sudo systemctl status docker | Checks if the Docker service is running. |
| sudo docker run hello-world | Runs a test container to confirm Docker works correctly. |

## 

## s

## 8. Install Specific Docker Version (Optional)

|  |  |
| --- | --- |
| Command | Explanation / Purpose |
| sudo apt-cache madison docker-ce | Lists available Docker versions from the repository. |
| sudo apt-get install docker-ce=<VERSION\_STRING> docker-ce-cli=<VERSION\_STRING> containerd.io docker-compose-plugin | Installs a specific version of Docker instead of the latest. |

## 9. Install Specific Version (Optional)

apt-cache madison docker-ce

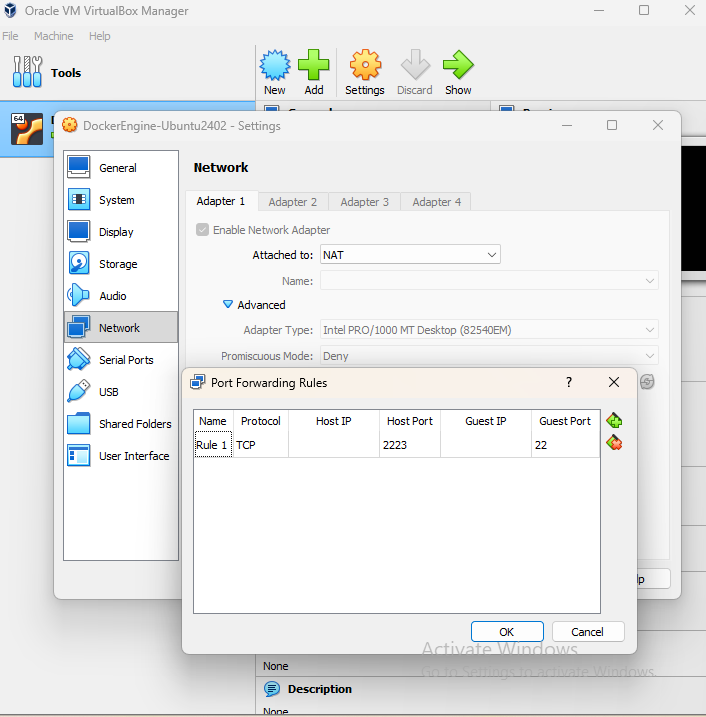
sudo apt-get install docker-ce=<VERSION\_STRING> docker-ce-cli=<VERSION\_STRING> containerd.io docker-compose-plugin

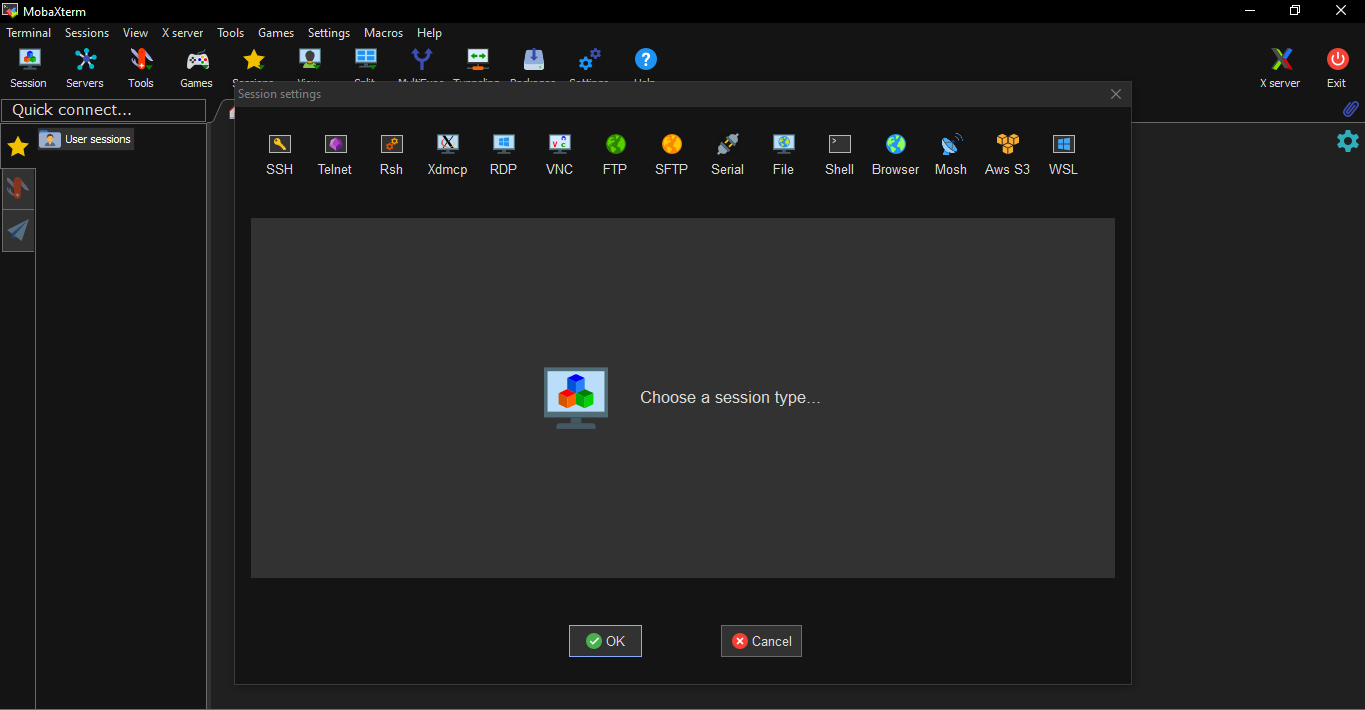
## Using Mobaxterm to Login

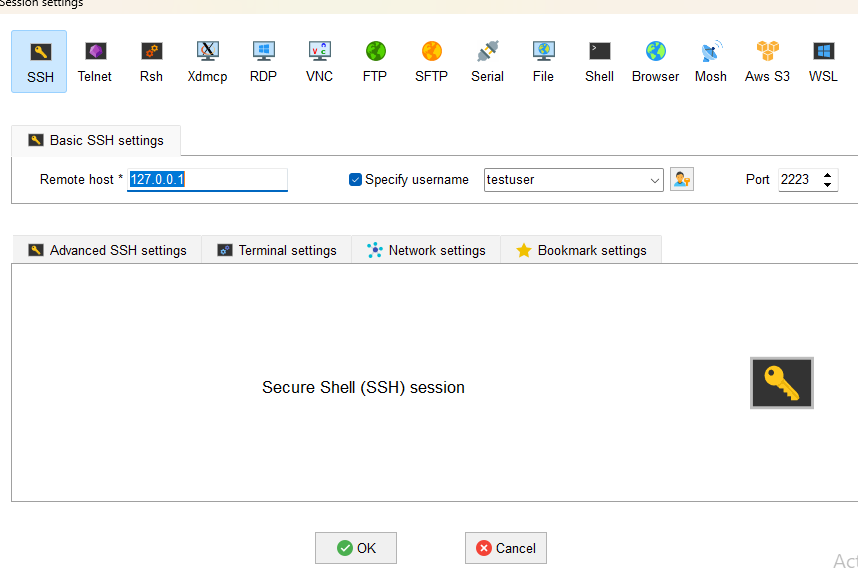
* Download Mobaxterm:  
  - Open browser → Search 'Mobaxterm download for Windows'  
  - Go to https://mobaxterm.mobatek.net/download-home-edition.html  
  - Download Portable Edition, extract ZIP, run Mobaxterm.exe as Administrator.
* Create SSH Session:  
  - Open Mobaxterm → Click 'Session' → Select 'SSH'.  
  - Remote Host: 192.186.1.207  
  - Port: 2223  
  - Tick 'Specify Username' → enter testuser  
  - Click OK.
* Login with Password:  
  - Enter testuser password (hidden while typing).  
  - When asked to save password → Select No.  
  - You are now logged into the VM.

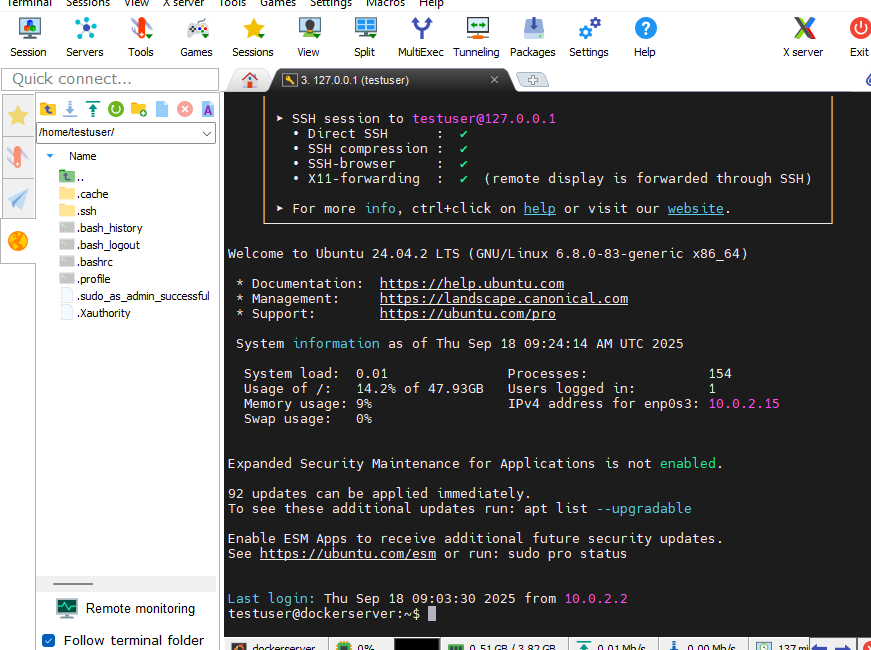
### Troubleshooting Mobaxterm Network Error

If you see a 'network error', it is usually because of NAT networking in VirtualBox. NAT allows internet access but does not allow external SSH connections.  
  
Fix: Enable Port Forwarding in Oracle VirtualBox:  
- Open VirtualBox → Select VM → Click Settings.  
- Go to Network → Advanced → Port Forwarding.  
- Add Rule:  
 Host Port: 2223  
 Guest Port: 22  
- Save changes.



Reconnect via Mobaxterm with Port Forwarding:  
- Close old session.  
- Create new SSH session in Mobaxterm:  
 Remote Host: localhost  
 Port: 2223  
 Username: testuser  
- Enter password → Login successful.  
****





## 2. Using Command Prompt to Login

* Open Command Prompt in Windows.
* Run the SSH command:  
   ssh -p 2222 testuser@localhost  
   (or ssh -p 2222 testuser@<VM\_IP>)
* On first login, type 'yes' when asked for confirmation.
* Enter testuser password.
* Now you are logged in, prompt will show: testuser@dockerserver:~$

## Switch to Root User

Once logged in, you can switch to root (administrator) user by running:

sudo -i

Enter the password when asked. Now you can run commands without typing 'sudo' each time.