**Docker Engine Setup using Package Manager**

Docker Engine can be installed using three different methods.   
1) Install manually and upgrade manually   
2) Install using Docker’s apt Repository  
3) Install using Convenience Scripts

**Docker Engine Installation Commands using Docker’s apt repository/ Package Manager**

* Login to the Virtual Machine using username (testuser) and password (\*\*\*\*) created in Oracle Virtual Box.
* After the first login, apt index page should be updated. For that apt update command is used. For other than root user (testuser), the command is as follows **sudo apt-get update**.
* If logged in as root user, skip sudo and run the command **apt-get update** as root is the administrator of the Operating System.
* Next install prerequisite packages like ca-certificate, curl, GnuPG, lsb-release which are required to not have any errors in the next Docker Engine installation processes.

**Note:** Ca-certificate is used for authentication & trustworthiness of multiple domains for security. Curl (Client URL) is used for data exchange between a device & a server through a terminal. GnuPG is a CLI tool which has versatile key management system that allows you to encrypt and sign your data & communications. Lsb-release is used to extract Linux Standard Based and distribution information.

* Multiple packages can be installed at once by mentioning it side by side. The command is as follows **sudo apt-get install ca-certificates curl gnupg lsb-release.**
* Next create a key rings directory to store Docker repository GPG Key by running the command **sudo mkdir -p /etc/apt/keyrings.** This creates a directory in apt directory which is present in etc directory in Root.
* Next download the GPG key in a Keyrings directory by running the command **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg.**
* As we are using Docker public repository, downloading GPG key is necessary.
* Next create a repository for docker. From the local system, Docker Engine packages have to be downloaded into the VM using the a URL – download.docker.com
* The below command is used to create repository entries in the local system in a file called **docker.list** under the path /etc/apt/sources.list.d. This docker.list file has entries regarding how to contact from the local system to the remote repository.

**echo \**

**"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \**

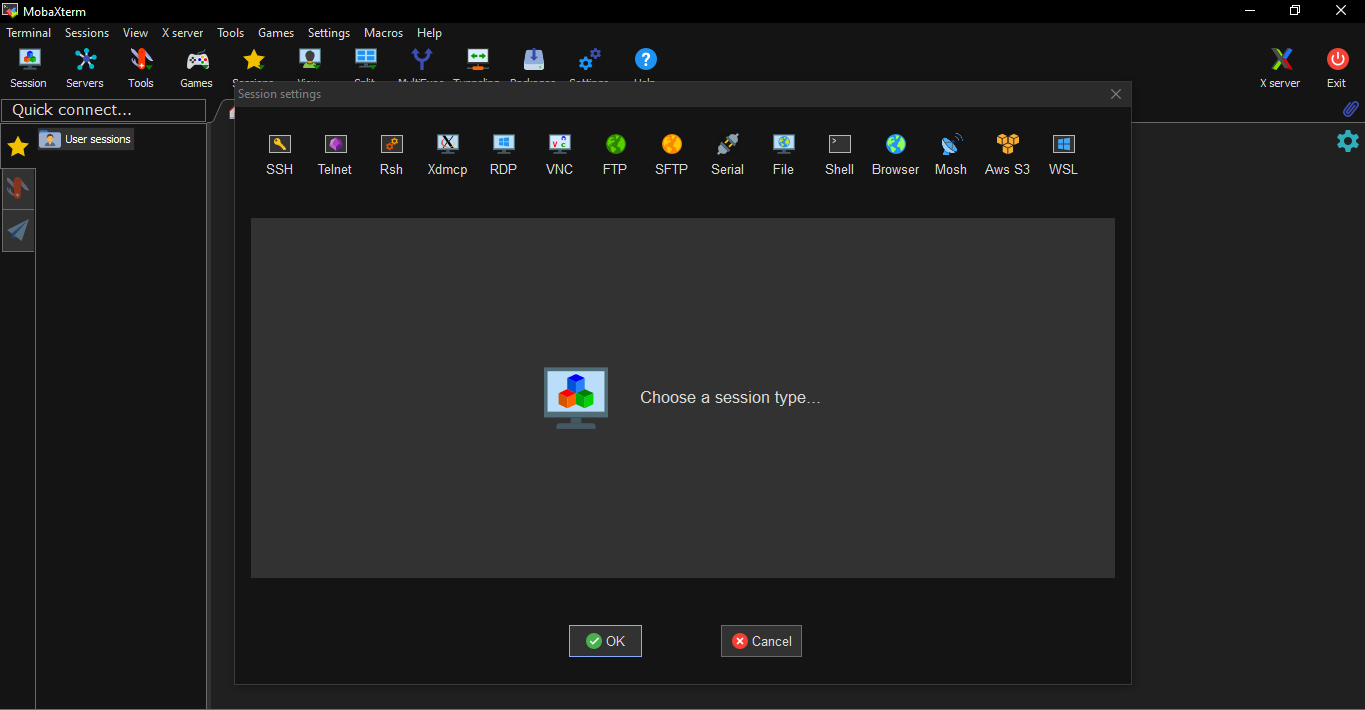
**$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

* Next, whenever there is a modification of entries or creation of new entries in sources.list file it is recommended to run **sudo apt-get update** command again.
* Next install Docker Engine CE Package on the system, which will be downloaded from the above repository using the command **sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin.** Here the packages that are installed are docker-ce (server package), docker-ce-cli (client package), containerd.io (backend tool to create containers) and docker-compose-plugin.
* Official document URL to install Docker Engine on Ubuntu VM - <https://docs.docker.com/engine/install/ubuntu/>

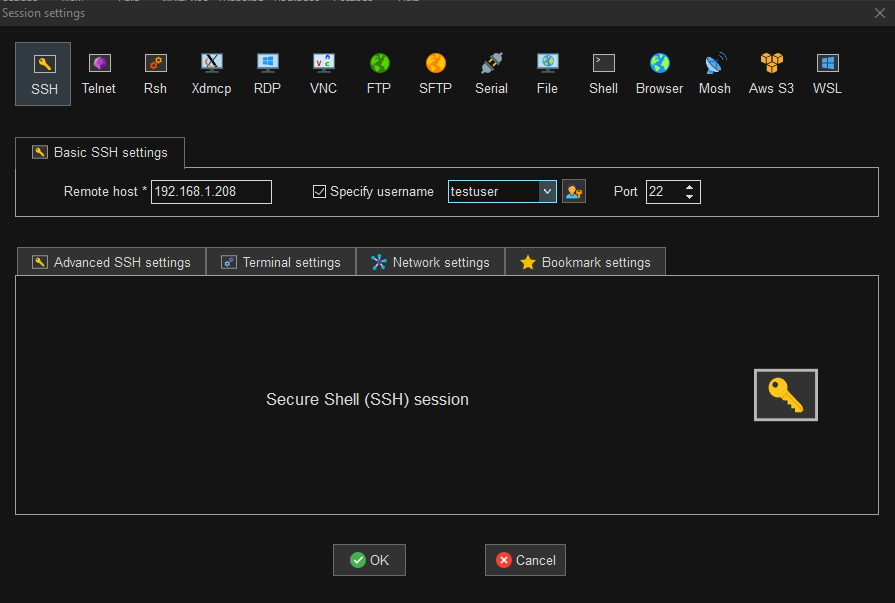
**Note:** To copy the commands from official document and paste them in the Oracle Virtual box terminal is not possible. Use either Mobaxterm or Command Prompt to work further.

**Using Mobaxterm to login to a VM**

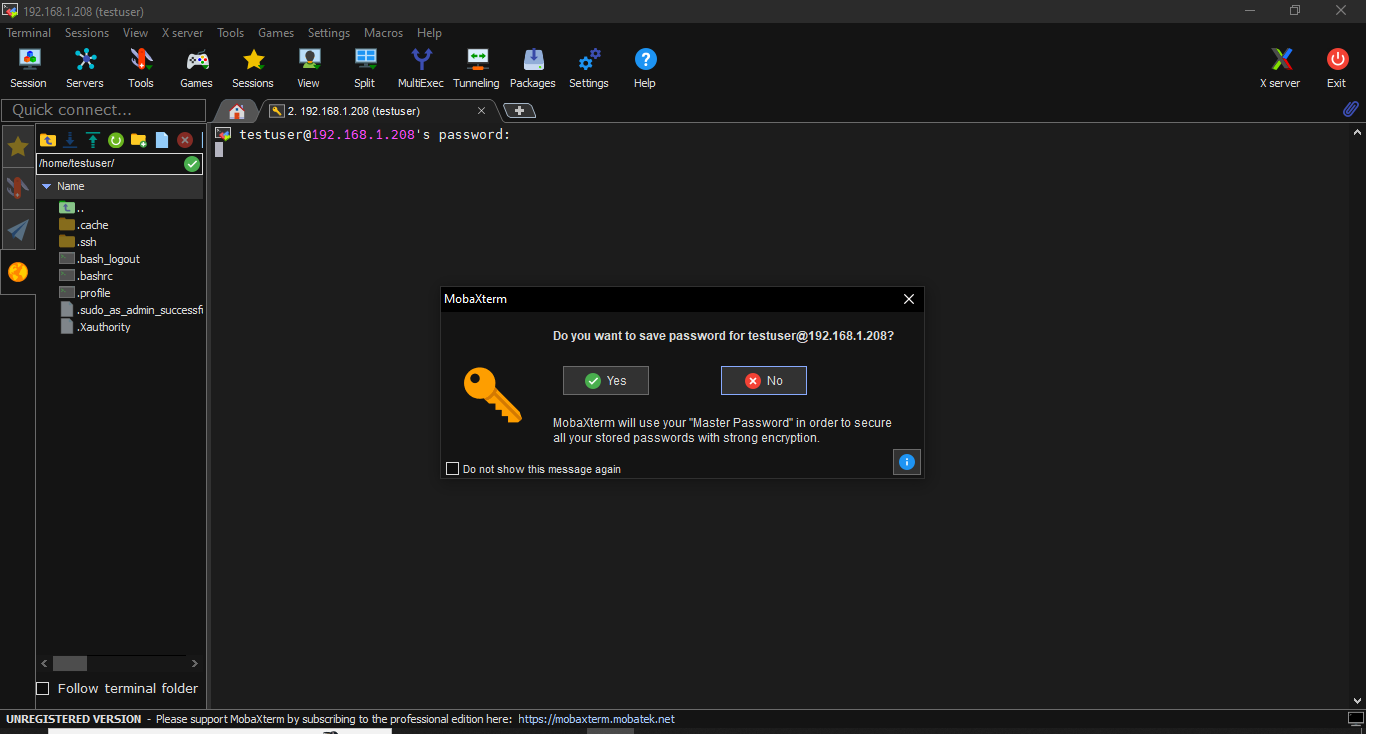
* Search for Mobaxterm download for windows in any browser and go the first displayed link - <https://mobaxterm.mobatek.net/download-home-edition.html>. Download the Portal edition of Mobaxterm. Extract the downloaded zip file into a folder. A Mobaxterm Application will be extracted from the zip file. Run this application as administor to install Mobaxterm.
* Once Mobaxterm is installed completely, open it. Click on sessions to add a new Session.

****

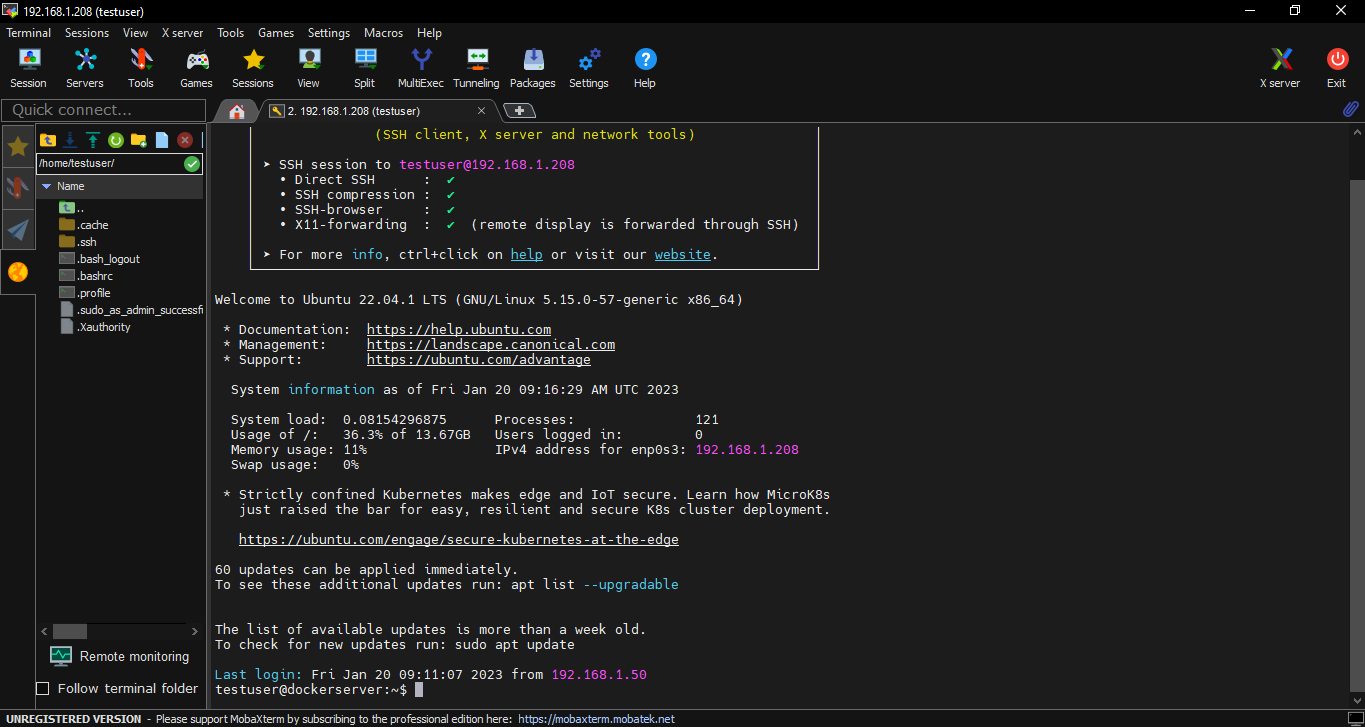
* Click on SSH and add Remote Host : IP address of the VM (192.186.1.207), Tick the Specify username box and give the username of the VM (testuser), Port number : 22 which is default and then click on



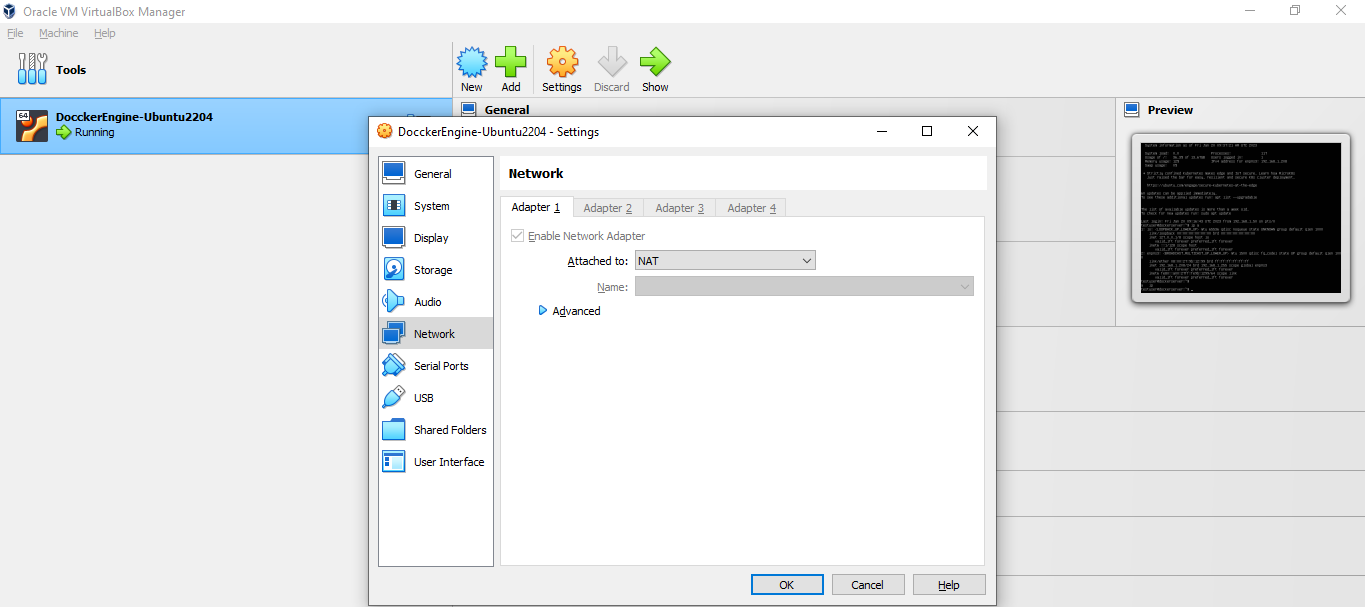
* The session asks for password of the user specified (testuser). Give the password (which will not be displayed) and click on enter. A pop up is displayed whether to save the password of the user or not. Select NO and click on enter.

****

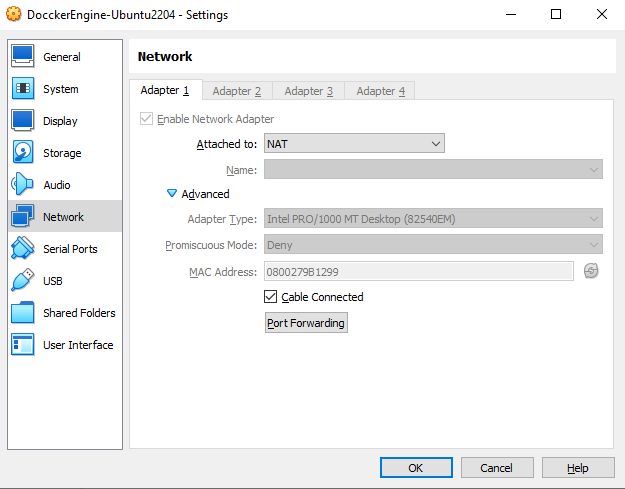
* A VM session logged in as testuser is ready for use



* If a network error message is displayed, then it is because of NAT network. NAT gateway can be used to connect to internet but can’t be used to login to the VM from any third-party tools even through the base system.
* To solve this issue, open the Oracle Virtual box console and click on setting. Open Network from the displayed options.

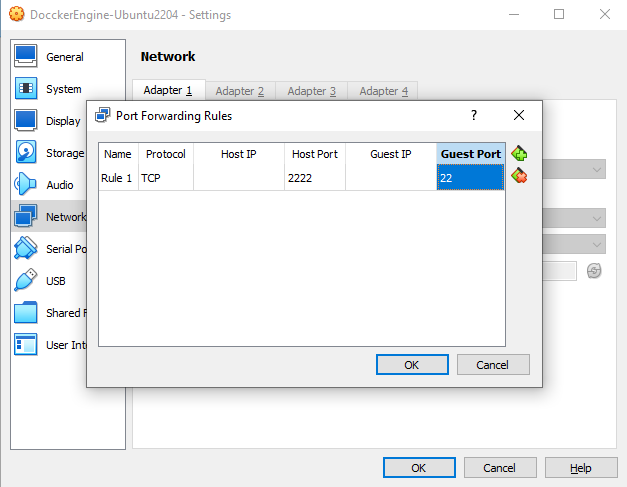


* Click on advanced and select port forwarding. In Port Forwarding, a port number on the Local system/ Base machine is bound to the Virtual Machine. When this port forwarding is done then on whenever we try to communicate to the base machine on the specified port number (given in Host port), it basically connects to the Virtual Machine

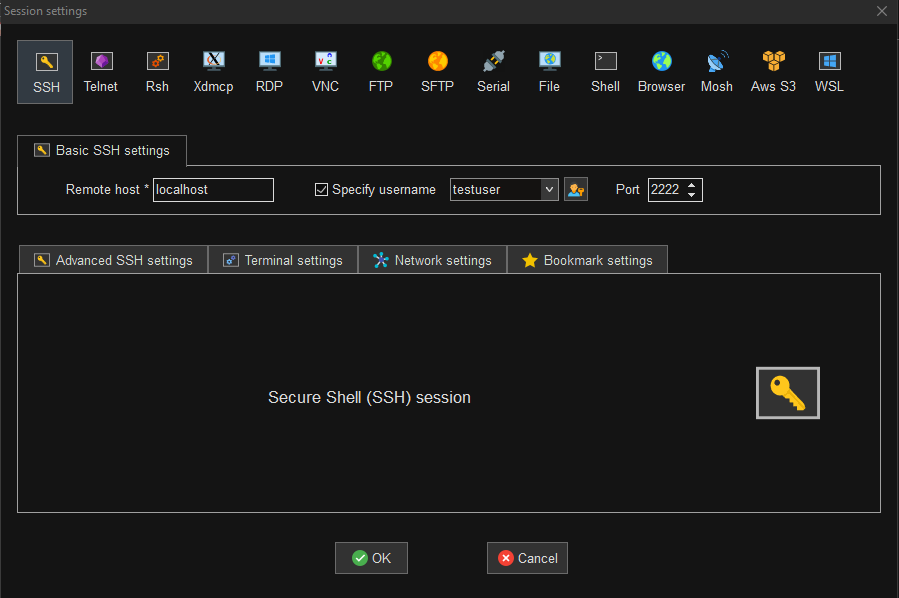


* A pop up is opened to add ports. Click on add symbol displayed in the right column to add host port and guest port. Host port is the Base Machine port that we are binding to the VM. Guest Port is the port used to SSH into the VM.

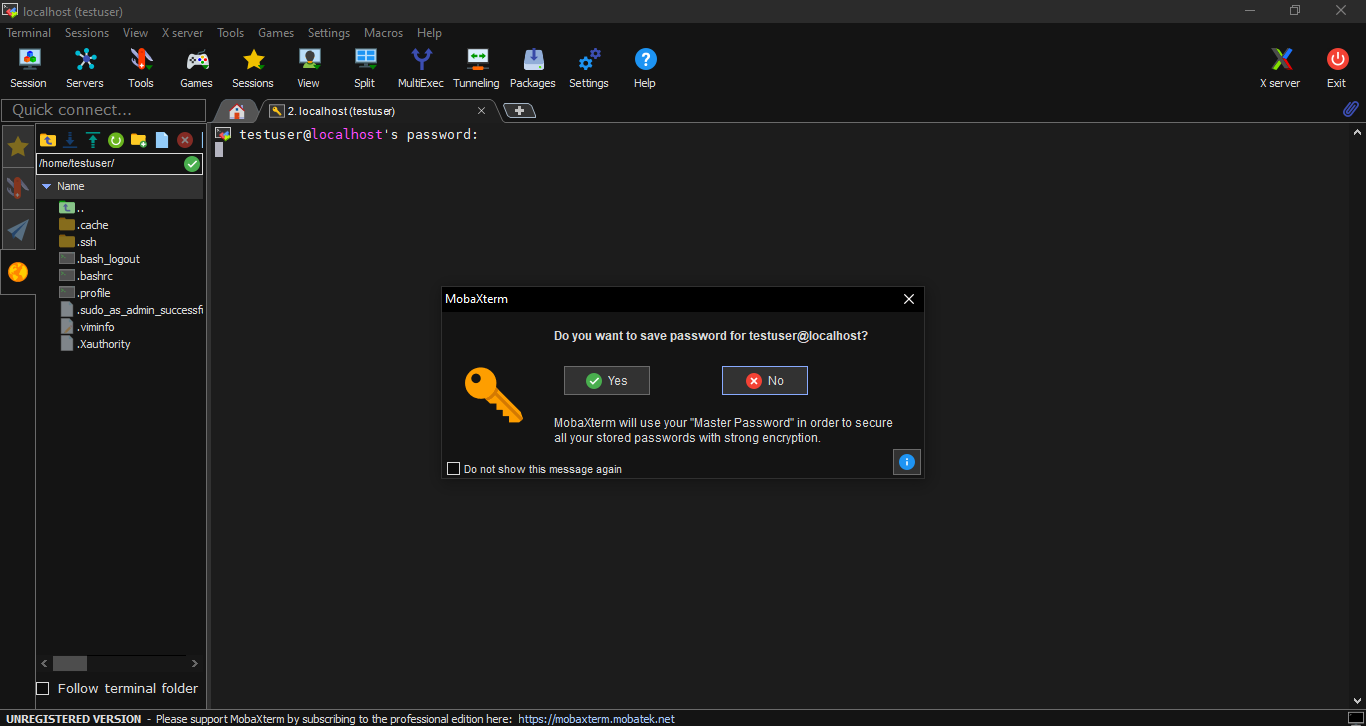
Give Host Port of your choice (2222) and Guest port (22) as we are using SSH to login. Click on OK to save the changes.



* Now, Go to Mobaxterm and close the added session and open a new session. Click on SSH and give Remote host as Local host (Local laptop/ Physical server), Tick the specific username box and give VM user name (testuser), Port as 2222 which is default and then click on OK.



* If the session asks for the VM user (testuser) password that means you are able to login to the VM. Give the password (will not be displayed). Don’t save the password, if asked. Click on No.



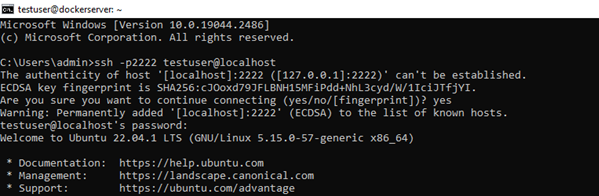
* Now you are successfully logged into an Ubuntu VM using Mobaxterm.

**Using Command Prompt to Login to VM**

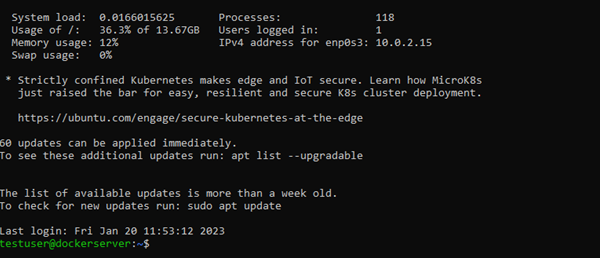
* Command prompt can also be used to login to the VM. Directly open Command Prompt in windows and run the following command –

**ssh -p2222 username@localhost (or) [ssh –p2222 testuser@ipaddress of the VM]** to login to the VM.

* Make sure that the network setting of the VM is NAT and Port forwarding is done. In Port Forwarding, a port number on the Local system/ Base machine is bound to the Virtual Machine. When this port forwarding is done then on whenever we try to communicate to the base machine on the specified port number (given in Host port), it basically connects to the Virtual Machine
* The Host port is 2222 (any port of your choice) and Guest port is 22. Host port is the Base Machine port that we are binding to the VM. Guest Port is the port used to SSH into the VM.
* A confirmation message will be displayed asked by the terminal. Type ‘yes’ and click enter.
* Next the terminal asks for the VM user password, give the password and click on enter. Then you will be able to successful login to the VM using Command Prompt.



* The path displayed will be username@VMname i.e., [testuser@dockerserver]



* Switch to the root user by running the command **sudo –i.** Terminal asks for the password, give the password the VM user. There is no need to add sudo before all the commands, if logged in as root user.

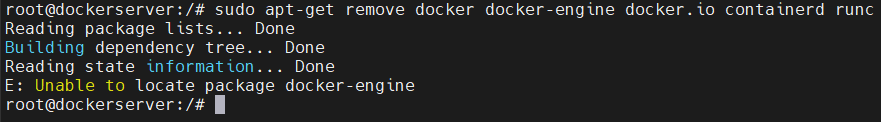
****

**Note: Using either Mobaxterm or Command Prompt login to the VM and follow the below steps.**

**Steps to install Docker Engine using Package manager on Ubuntu in Command Prompt**

* If an older version of Docker is available uninstall it and install the latest version of Docker. To uninstall the older version run the command - **sudo apt-get remove docker docker-engine docker.io containerd runc.**

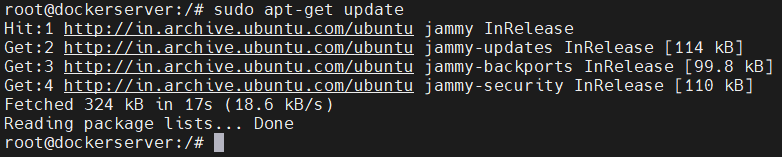
**Note: Use right click to paste the copied commands in command prompt.**



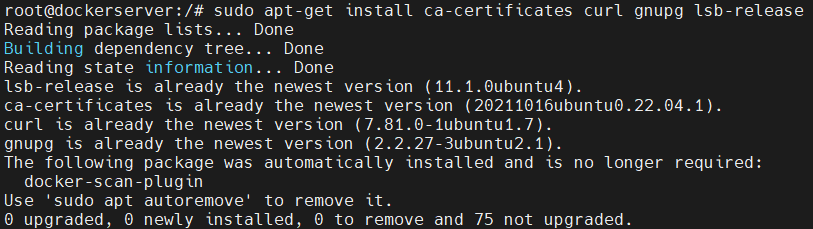
* As this is a new VM, there will be no Docker version available and to recheck whether docker is available or not check the folder availability by running the command – **ls -l /var/lib/docker.** This should display No such directory exists.

****

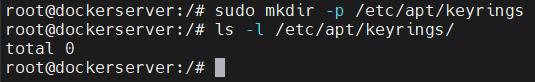
* Update the index page by running the command **apt –get update**. Add ‘sudo’ before the command if not logged in as root user.



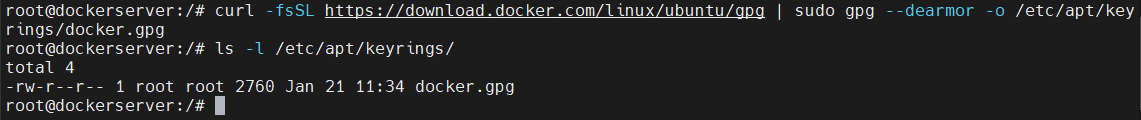
* Install ca-certificate, curl, GnuPG, lsb-release packages by running the command – **sudo** **apt-get install ca-certificates curl gnupg lsb-release.**



* Create a folder to store GPD keys i.e., public keys related to docker by running the command - **sudo mkdir -p /etc/apt/keyrings.** Verify the folder creation by running the command – **ls -l /etc/apt/keyrings/.** This should display the no.of files available in keyrings directory which should be zero files available as it is a newly created empty directory.



* To download the GPG key and save it in Keyrings directory with the name docker.gpg, run the following command - **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg.** As we are using Docker public repository, downloading GPG key is necessary.
* To verify the creation of docker.gpg file run the command - – **ls -l /etc/apt/keyrings/.**



* Create a Repository (docker.list) to install docker packages in it by running the following command -

**echo \**

**"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \**

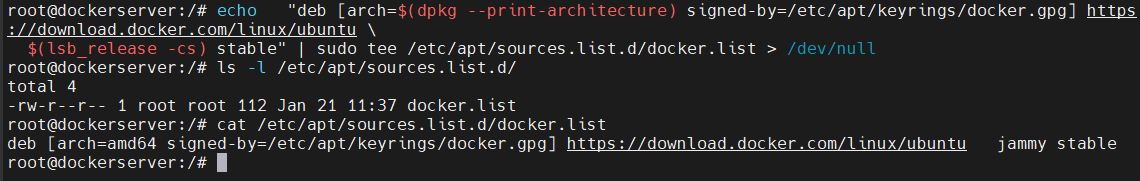
**$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null**

* To verify the file creation run the command –

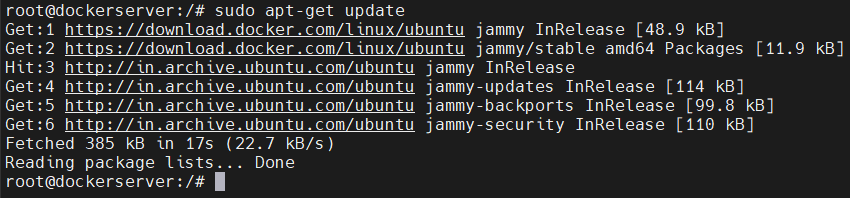
**ls -l /etc/apt/sources.list.d/**

* To read the contents of the docker repository file docker.list present in sources.list.d directory run the command -

**cat /etc/apt/sources.list.d/docker.list**



* As a new repository is added run **sudo apt-get update** command to update the index page regarding the updated/added repositories.



* If there is any error regarding GPG key, run the following two commands one after the other

**sudo chmod a+r /etc/apt/keyrings/docker.gpg**

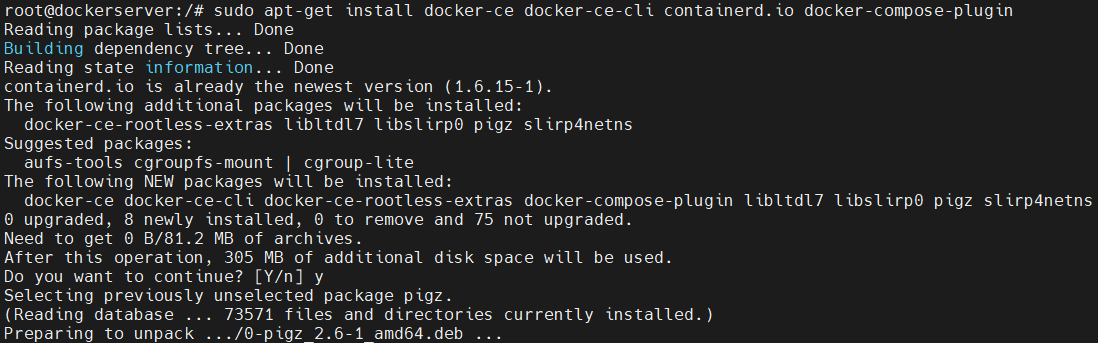
**sudo apt-get update**

* To install the latest packages of Docker Engine, containerd, and Docker compose run the following command –

**sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin**

The docker-ce is Community Edition server package, docker-ce-cli is Community Edition client package, containerd.io is a backend tool to create containers using namespaces & cgroups concept and docker-compose-plugin.

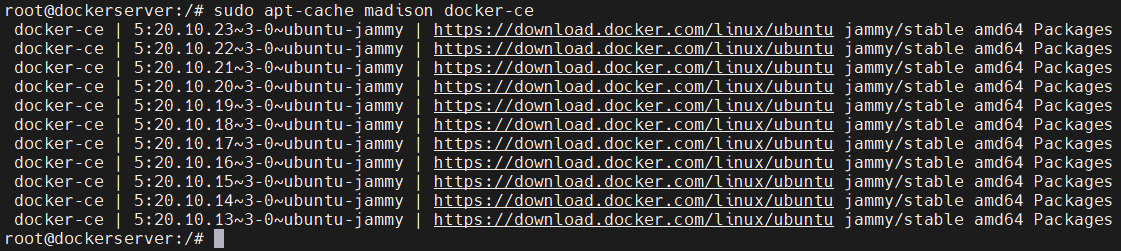
If asked for any permission to additional usage of disk give ‘Y’ and click enter.



* Specific version packages of Docker Engine can also be installed. To know the available versions run the command –

**sudo apt-cache madison docker-ce**

From the displayed version choose the version according to the requirement and install it.



* Once Docker Engine is installed successfully check the version of the installed Docker Engine by running the command - **docker** --**version.**

