**Docker – Installing and configuring**

To start the installation of Docker, we are going to use an Ubuntu instance. You can use Oracle Virtual Box to setup a virtual Linux instance, in case you don’t have it already.

The following screen shot shows a simple Ubuntu server which has been installed on Virtual Box. There is an OS user named **demo** which has been defined on the system having entire root access to the sever.



To install Docker, we need to follow the steps given below.

**Step 1**: Before installing Docker, you first have to ensure that you have the right Linuxkernel version running. Docker is only designed to run on Linux kernel version 3.8 and higher. We can do this by running the following command:

**uname**

This method returns the system information about the Linux system.

**Syntax**

uname -a

**Options**

**a** –This is used to ensure that the system information is returned.

**Return Value**

This method returns the following information on the Linux system:

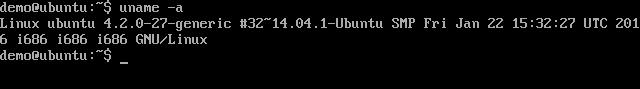
* kernel name
* node name
* kernel release
* kernel version
* machine
* processor
* hardware platform
* operating system

**Example**

uname –a

**Output**

When we run above command, we will get the following result:



From the output, we can see that the Linux kernel version is 4.2.0-27 which is higher than version 3.8, so we are good to go.

**Step 2**: You need to update the OS with the latest packages, which can be done via thefollowing command:

apt-get

This method installs packages from the Internet on to the Linux system.

**Syntax**

sudo apt-get update

**Options**

* **sudo** - The **sudo** command is used to ensure that the command runs with rootaccess.
* **update** - The **update** option is used ensure that all packages are updated on theLinux system.

**Return Value**

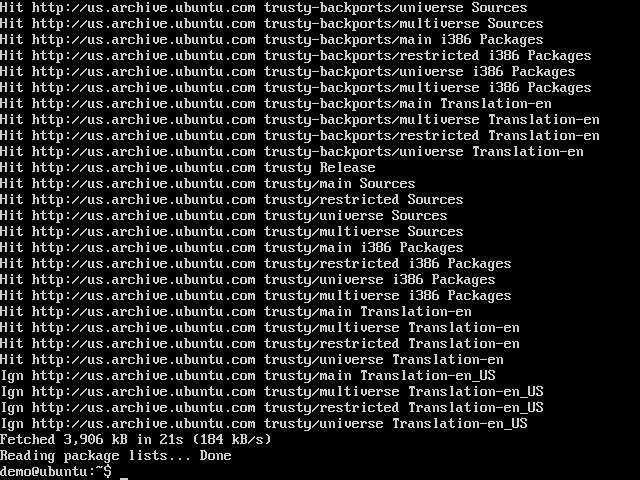
None

**Example**

sudo apt-get update

**Output**

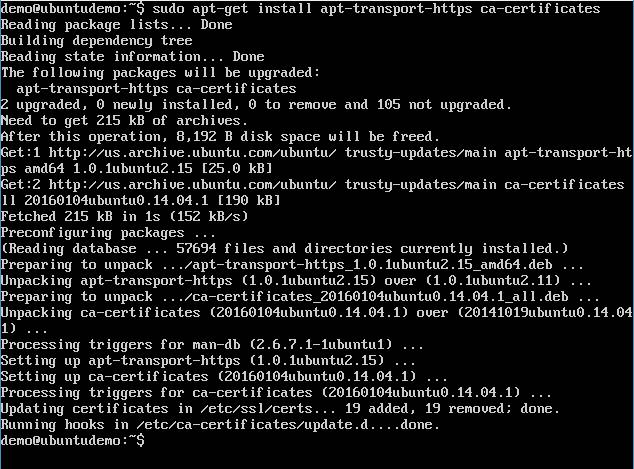
When we run the above command, we will get the following result:



This command will connect to the Internet and download the latest system packages for Ubuntu.

**Step 3**: The next step is to install the necessary certificates that will be required to workwith the Docker site later on to download the necessary Docker packages. It can be done with the following command:

sudo apt-get install apt-transport-https ca-certificates



**Step 4**: The next step is to add the new GPG key. This key is required to ensure that alldata is encrypted when downloading the necessary packages for Docker.

The following command will download the key with the ID

58118E89F3A912897C070ADBF76221572C52609D from the **keyserver** hkp://ha.pool.sks-keyservers.net:80 and adds it to the **adv** keychain. Please note that this particular key is required to download the necessary Docker packages.

sudo apt-key adv \

--keyserver hkp://ha.pool.sks-keyservers.net:80 \

--recv-keys 58118E89F3A912897C070ADBF76221572C52609D



**Step 5**: Next, depending on the version of Ubuntu you have, you will need to add therelevant site to the **docker.list** for the **apt package manager**, so that it will be able to detect the Docker packages from the Docker site and download them accordingly.

* Precise 12.04 (LTS) ─ deb https://apt.dockerproject.org/repo ubuntu-precise main
* Trusty 14.04 (LTS) ─ deb https://apt.dockerproject.org/repo ubuntu-trusty main
* Wily 15.10 ─ deb https://apt.dockerproject.org/repo ubuntu-wily main
* Xenial 16.04 (LTS) ─ deb https://apt.dockerproject.org/repo ubuntu-xenial main

Since our OS is Ubuntu 14.04, we will use the Repository name as “deb https://apt.dockerproject.org/repo ubuntu-trusty main”

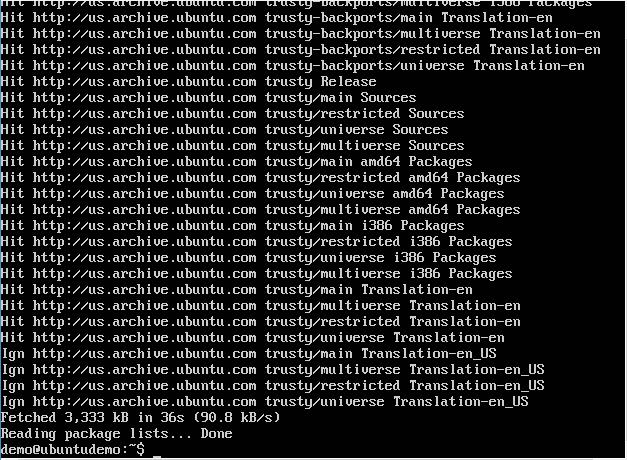
And then, we will need to add this repository to the **docker.list** as mentioned above.



echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main” | sudo tee /etc/apt/sources.list.d/docker.list



**Step 6**: Next, we issue the **apt-get update command** to update the packages on theUbuntu system.

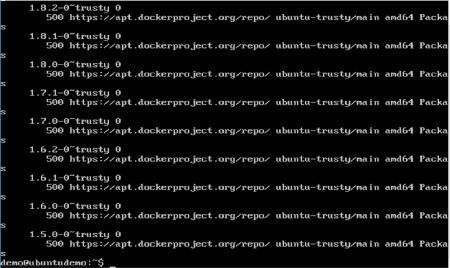


**Step 7**: If you want to verify that the package manager is pointing to the right repository,you can do it by issuing the **apt-cache command**.

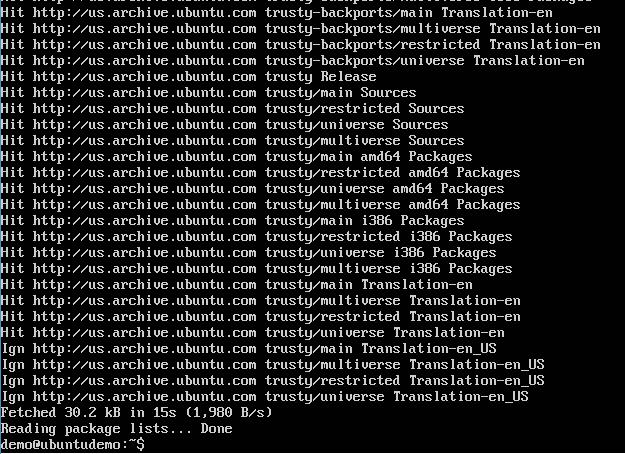


apt-cache policy docker-engine

In the output, you will get the link to <https://apt.dockerproject.org/repo/>



**Step 8**: Issue the **apt-get update command** to ensure all the packages on the localsystem are up to date.

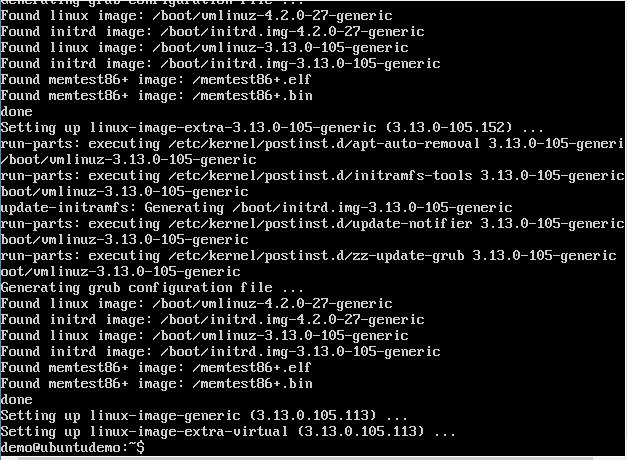


**Step 9**: For Ubuntu Trusty, Wily, and Xenial, we have to install the linux-image-extra-\*kernel packages, which allows one to use the **aufs storage driver**. This driver is used by the newer versions of Docker.

It can be done by using the following command:



sudo apt-get install linux-image-extra-$(uname -r) linux-image-extra-virtual



**Step 10**: The final step is to install Docker and we can do this with the following command:

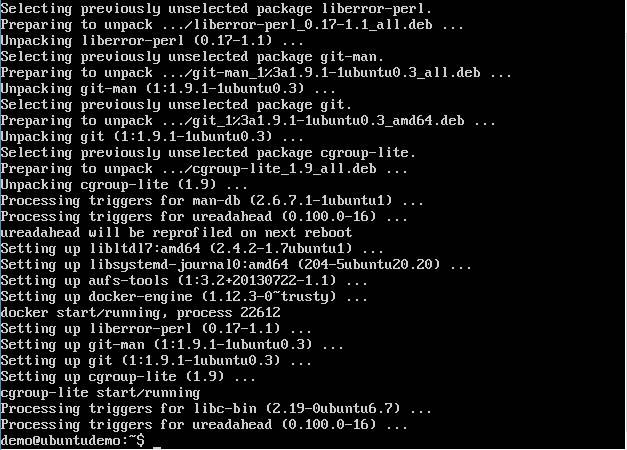


sudo apt-get install –y docker-engine

Here, **apt-get** uses the install option to download the Docker-engine image from the Docker website and get Docker installed.

The Docker-engine is the official package from the Docker Corporation for Ubuntu-based systems.

Docker



In the next section, we will see how to check for the version of Docker that was installed.

**Docker Version**

To see the version of Docker running, you can issue the following command:

**Syntax**

docker version

**Options**

* **version** –It is used to ensure the Docker command returns the Docker versioninstalled.

**Return Value**

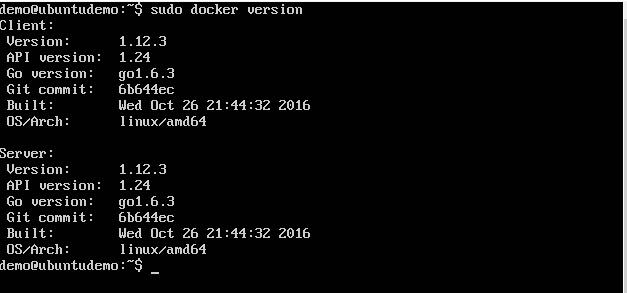
The output will provide the various details of the Docker version installed on the system.

**Example**

sudo docker version

**Output**

When we run the above program, we will get the following result:



**Docker Info**

To see more information on the Docker running on the system, you can issue the following command:

**Syntax**

docker info

**Options**

* **info** –It is used to ensure that the Docker command returns the detailedinformation on the Docker service installed.

**Return Value**

The output will provide the various details of the Docker installed on the system such as

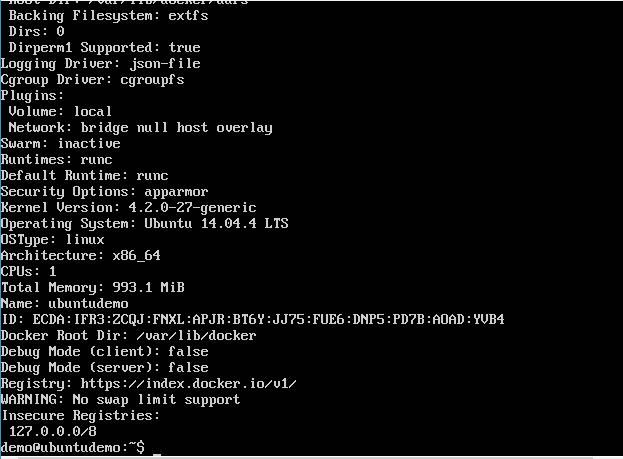
* Number of containers
* Number of images
* The storage driver used by Docker
* The root directory used by Docker
* The execution driver used by Docker

**Example**

sudo docker info

**Output**

When we run the above command, we will get the following result:



**Docker for Windows**



Docker has out-of-the-box support for Windows, but you need to have the following configuration in order to install Docker for Windows.

**System Requirements**

**Windows OS Windows 10 64 bit**

**Memory 2 GB RAM (recommended)**

You can download Docker for Windows from: <https://docs.docker.com/docker-for-windows/>



**Docker ToolBox**



Docker ToolBox has been designed for older versions of Windows, such as Windows 8.1 and Windows 7. You need to have the following configuration in order to install Docker for Windows.

**System Requirements**

Windows OS Windows 7 , 8, 8.1

Memory 2 GB RAM (recommended)

Virtualization This should be enabled.

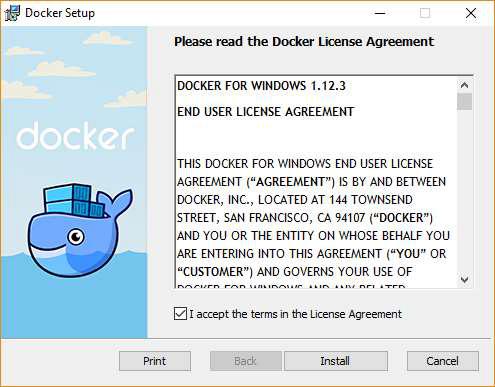
Let’s go through the installation of each product.

**Docker for Windows**



Once the installer has been downloaded, double-click it to start the installer and then follow the steps given below.

**Step 1**: Click on the Agreement terms and then the Install button to proceed ahead withthe installation.



**Step 2**: Once complete, click the Finish button to complete the installation.



**Docker ToolBox**

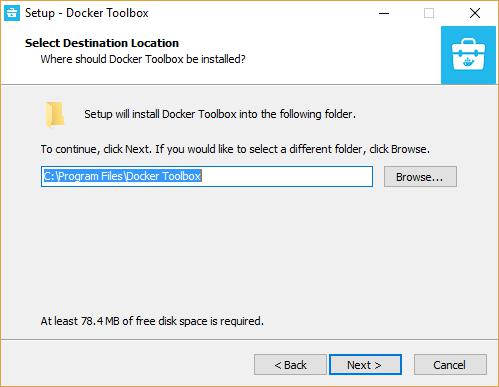


Once the installer has been downloaded, double-click it to start the installer and then follow the steps given below.

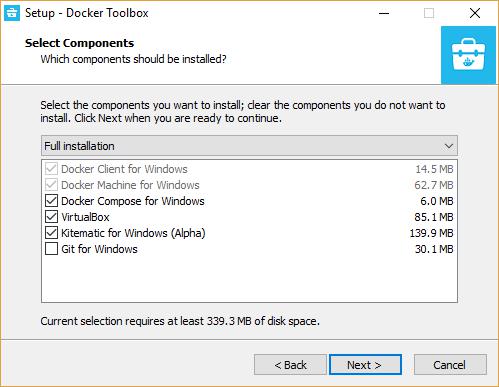
**Step 1**: Click the Next button on the start screen.



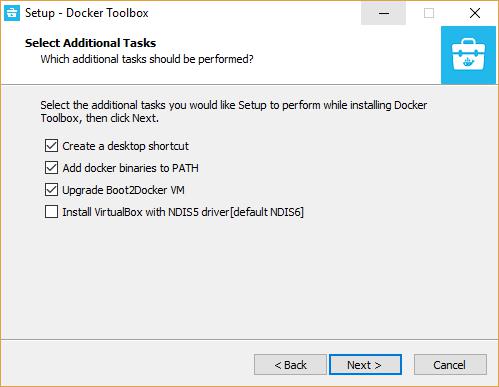
**Step 2**: Keep the default location on the next screen and click the Next button.



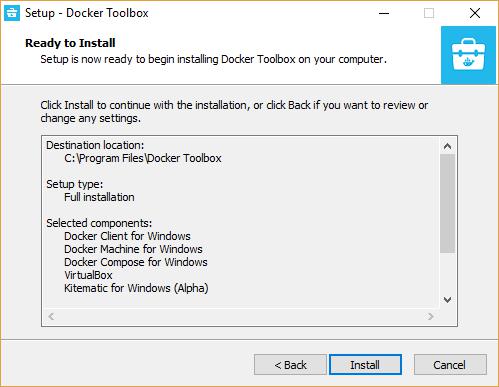
**Step 3**: Keep the default components and click the Next button to proceed.



**Step 4**: Keep the Additional Tasks as they are and then click the Next button.



**Step 5**: On the final screen, click the Install button.



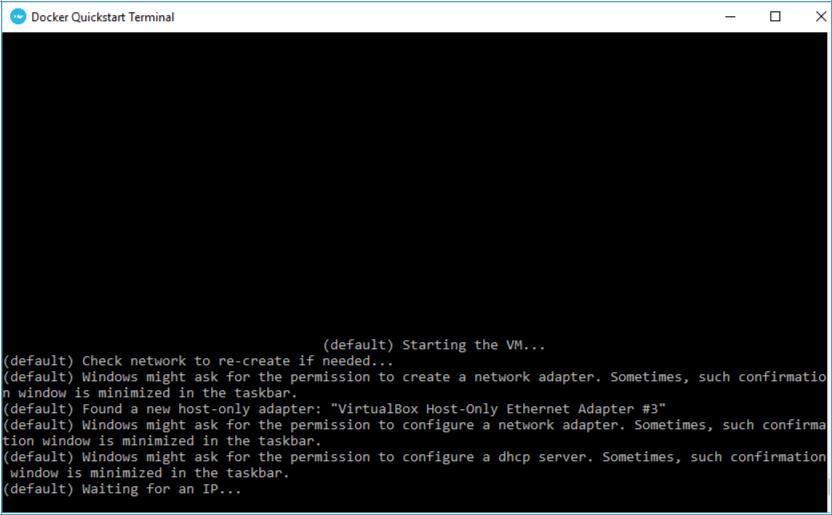
**Working with Docker Toolbox**



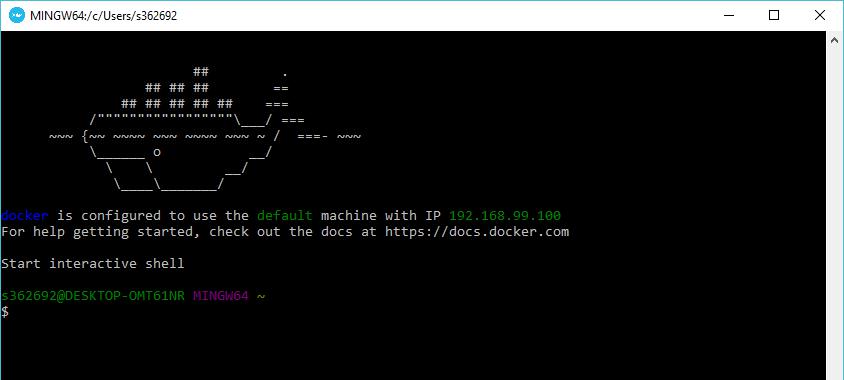
Let’s now look at how Docker Toolbox can be used to work with Docker containers on Windows. The first step is to launch the Docker Toolbox application for which the shortcut is created on the desktop when the installation of Docker toolbox is carried out.



Next, you will see the configuration being carried out when Docker toolbox is launched.



Once done, you will see Docker configured and launched. You will get an interactive shell for Docker.



To test that Docker runs properly, we can use the Docker **run command** to download and run a simple **HelloWorld Docker container**.

The working of the Docker **run command** is given below:

docker run

This command is used to run a command in a Docker container.

**Syntax**

docker run image

**Options**

* Image – This is the name of the image which is used to run the container.

**Return Value**

The output will run the command in the desired container.

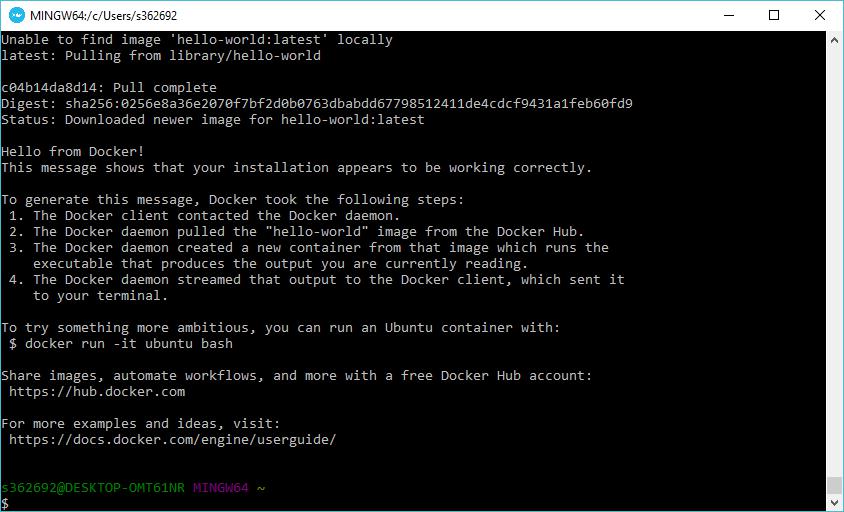
**Example**

sudo docker run hello-world

This command will download the **hello-world** image, if it is not already present, and run the **hello-world** as a container.

**Output**

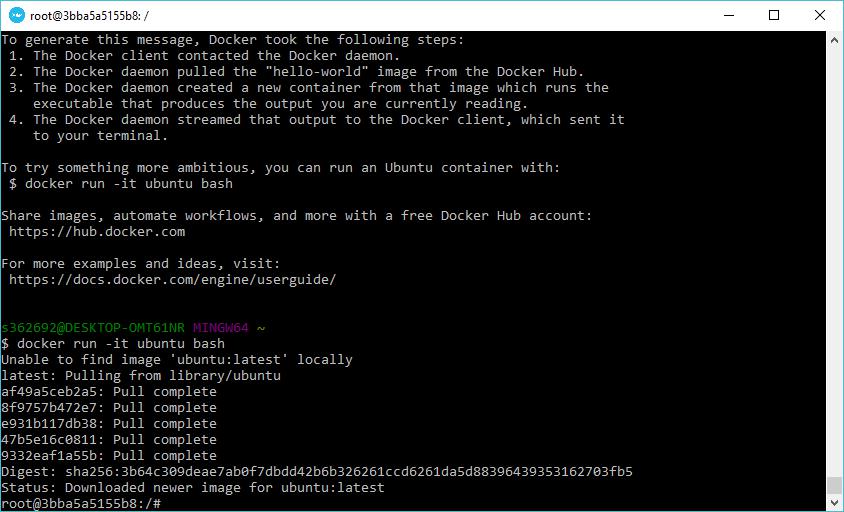
When we run the above command, we will get the following result:



If you want to run the Ubuntu OS on Windows, you can download the Ubuntu Image using the following command:

Docker run –it Ubuntu bash

Here you are telling Docker to run the command in the interactive mode via the **–it** option.



In the output you can see that the Ubuntu image is downloaded and run and then you will be logged in as a root user in the Ubuntu container.