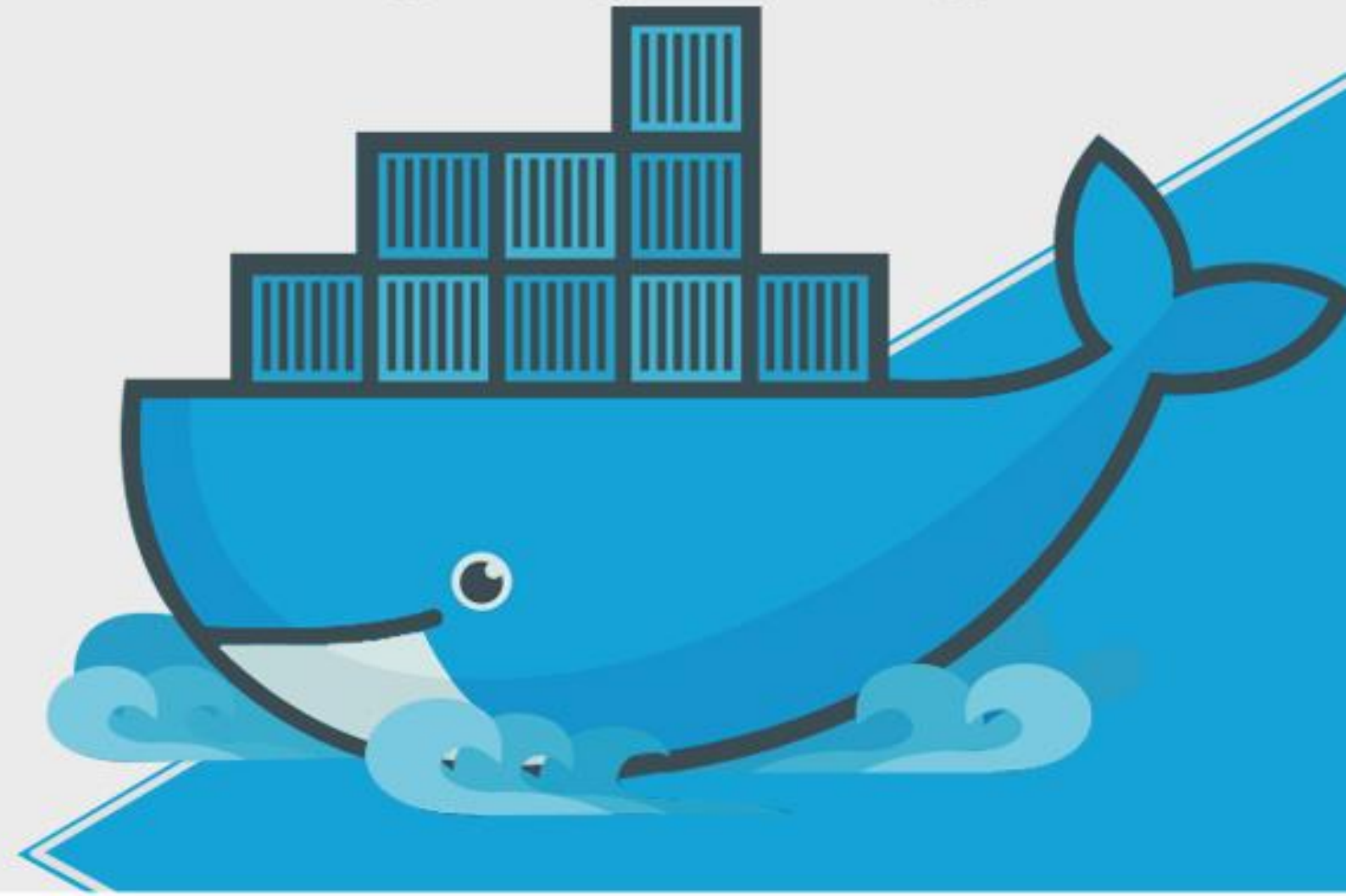
Docker



tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia https://twitter.com/tutorialspoint

About the Tutorial

This tutorial explains the various aspects of the Docker Container service. Starting with the basics of Docker which focuses on the installation and configuration of Docker, it gradually moves on to advanced topics such as Networking and Registries. The last few chapters of this tutorial cover the development aspects of Docker and how you can get up and running on the development environments using Docker Containers.

Audience

This tutorial is meant for those who are interested in learning Docker as a container service. This product has spread like wildfire across the industry and is really making an impact on the development of new generation applications. So anyone who is interested in learning all the aspects of Docker should go through this tutorial.

Prerequisites

The prerequisite is that the readers should be familiar with the basic concepts of Windows and the various programs that are already available on the Windows operating system. In addition, it would help if the readers have some exposure to Linux.

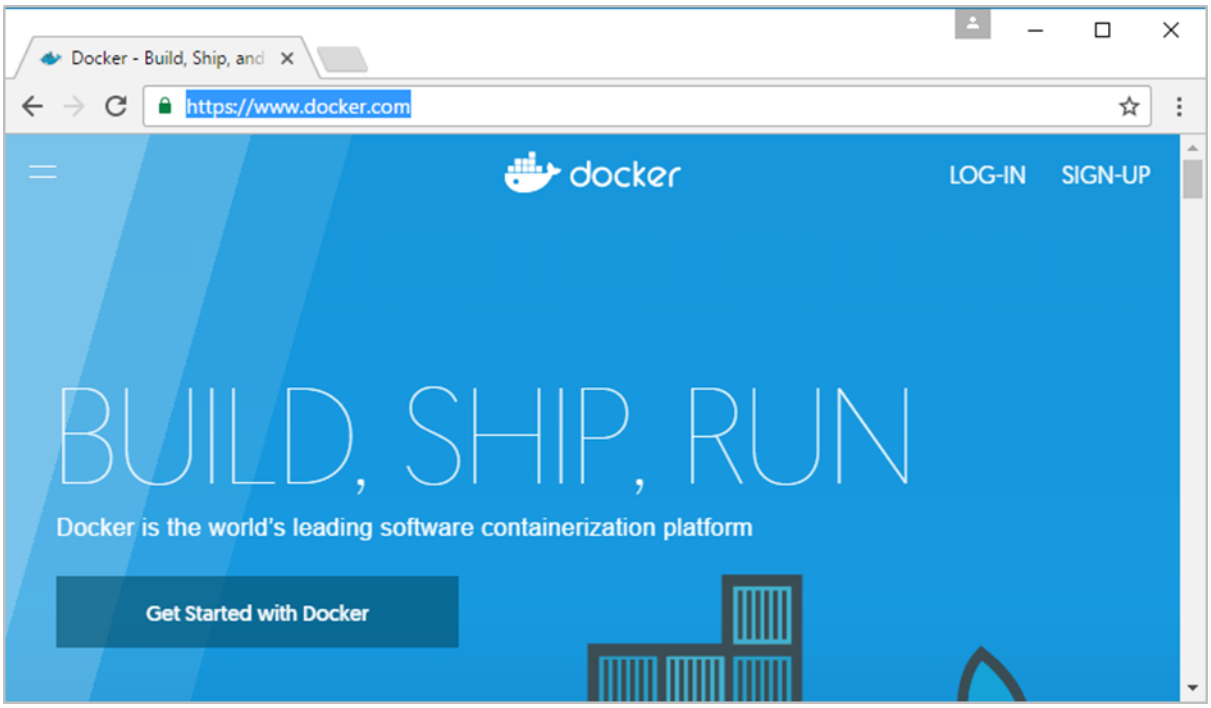
Copyright & Disclaimer

Table of Contents

1. DOCKER – OVERVIEW

Docker is a container management service. The keywords of Docker are **develop**, **ship** and **run** anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

The initial release of Docker was in March 2013 and since then, it has become the buzzword for modern world development, especially in the face of Agile-based projects.



Features of Docker

- Docker has the ability to reduce the size of development by providing a smaller footprint of the operating system via containers.
- With containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
- You can deploy Docker containers anywhere, on any physical and virtual machines and even on the cloud.
- Since Docker containers are pretty lightweight, they are very easily scalable.

Components of Docker

Docker has the following components

- **Docker for Mac** – It allows one to run Docker containers on the Mac OS.
- **Docker for Linux** - It allows one to run Docker containers on the Linux OS.
- **Docker for Windows** - It allows one to run Docker containers on the Windows OS.
- **Docker Engine** – It is used for building Docker images and creating Docker containers.
- **Docker Hub** – This is the registry which is used to host various Docker images.
- **Docker Compose** – This is used to define applications using multiple Docker containers.

We will discuss all these components in detail in the subsequent chapters.

The official site for Docker is https://www.docker.com/ The site has all information and documentation about the Docker software. It also has the download links for various operating systems.

2. DOCKER – INSTALLING DOCKER ON LINUX

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 screenshot shows a simple Ubuntu server which has been installed on Oracle 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 Linux kernel 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 the following 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 root access.
- **update** - The **update** option is used ensure that all packages are updated on the Linux system.

Return Value

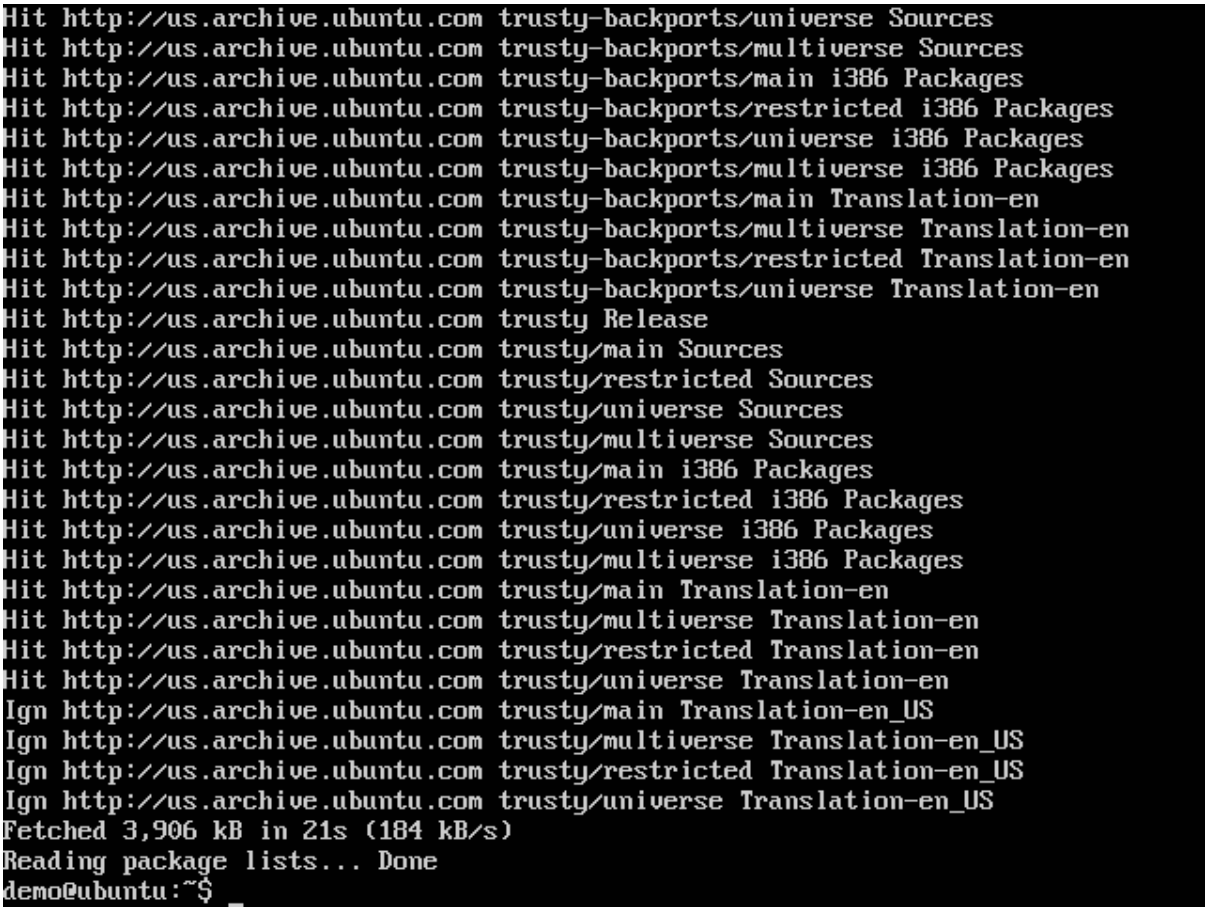
None

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 work with 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
```

```
demo@ubuntudemo:~$ sudo apt-get install apt-transport-https ca-certificates
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  apt-transport-https ca-certificates
2 upgraded, 0 newly installed, 0 to remove and 105 not upgraded.
Need to get 215 kB of archives.
After this operation, 8,192 B disk space will be freed.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main apt-transport-ht
ps amd64 1.0.1ubuntu2.15 [25.0 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main ca-certificates
ll 20160104ubuntu0.14.04.1 [190 kB]
Fetched 215 kB in 1s (152 kB/s)
Preconfiguring packages ...
(Reading database ... 57694 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.0.1ubuntu2.15_amd64.deb ...
Unpacking apt-transport-https (1.0.1ubuntu2.15) over (1.0.1ubuntu2.11) ...
Preparing to unpack .../ca-certificates_20160104ubuntu0.14.04.1_all.deb ...
Unpacking ca-certificates (20160104ubuntu0.14.04.1) over (20141019ubuntu0.14.04
1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up apt-transport-https (1.0.1ubuntu2.15) ...
Setting up ca-certificates (20160104ubuntu0.14.04.1) ...
Processing triggers for ca-certificates (20160104ubuntu0.14.04.1) ...
Updating certificates in /etc/ssl/certs... 19 added, 19 removed; done.
Running hooks in /etc/ca-certificates/update.d....done.
demo@ubuntudemo:~$
```

Step 4: The next step is to add the new GPG key. This key is required to ensure that all data 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
```

```
demo@ubuntudemo:~$ sudo apt-key adv \ --keyserver hkp://ha.pool.sks-keyservers.
et:80 \ --recv-keys 58118E89F3A912897C070ADBF76221572C52609D
Executing: gpg --ignore-time-conflict --no-options --no-default-keyring --homed
r /tmp/tmp.KcaZ3WlmGt --no-auto-check-trustdb --trust-model always --keyring /e
c/apt/trusted.gpg --primary-keyring /etc/apt/trusted.gpg  --keyserver hkp://ha.
ool.sks-keyservers.net:80  --recv-keys 58118E89F3A912897C070ADBF76221572C52609D
gpg: requesting key 2C52609D from hkp server ha.pool.sks-keyservers.net
gpg: key 2C52609D: public key "Docker Release Tool (releasedocker) <docker@dock
r.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
demo@ubuntudemo:~$
```

Step 5: Next, depending on the version of Ubuntu you have, you will need to add the relevant 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

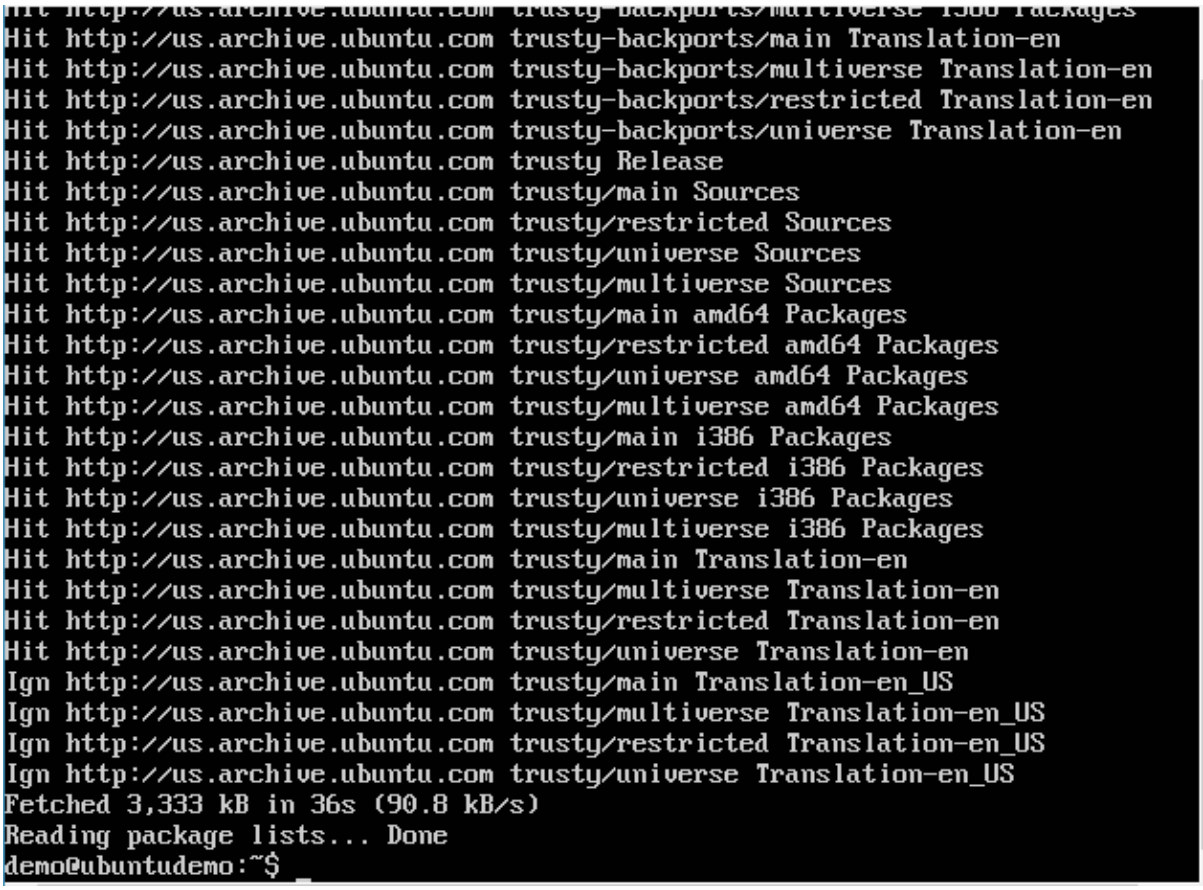
Since our OS is Ubuntu 14.04, we will use the Repository name as "deb https://apt.dockerproject.org/repo ubuntu-trusty main"

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
```

```
demo@ubuntudemo:~$ echo "deb https://apt.dockerproject.org/repo ubuntu-trusty
in" | sudo tee /etc/apt/sources.list.d/docker.list
deb https://apt.dockerproject.org/repo ubuntu-trusty main
demo@ubuntudemo:~$ _
```

Step 6: Next, we issue the **apt-get update command** to update the packages on the Ubuntu 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/

```
     1.8.2-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.8.1-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.8.0-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.7.1-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.7.0-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.6.2-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.6.1-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.6.0-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
     1.5.0-0~trusty 0
        500 https://apt.dockerproject.org/repo/ ubuntu-trusty/main amd64 Packa
s
demo@ubuntudemo:~$
```

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

```
Hit http://us.archive.ubuntu.com trusty-backports/main Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty-backports/universe Translation-en
Hit http://us.archive.ubuntu.com trusty Release
Hit http://us.archive.ubuntu.com trusty/main Sources
Hit http://us.archive.ubuntu.com trusty/restricted Sources
Hit http://us.archive.ubuntu.com trusty/universe Sources
Hit http://us.archive.ubuntu.com trusty/multiverse Sources
Hit http://us.archive.ubuntu.com trusty/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty/main i386 Packages
Hit http://us.archive.ubuntu.com trusty/restricted i386 Packages
Hit http://us.archive.ubuntu.com trusty/universe i386 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse i386 Packages
Hit http://us.archive.ubuntu.com trusty/main Translation-en
Hit http://us.archive.ubuntu.com trusty/multiverse Translation-en
Hit http://us.archive.ubuntu.com trusty/restricted Translation-en
Hit http://us.archive.ubuntu.com trusty/universe Translation-en
Ign http://us.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us.archive.ubuntu.com trusty/multiverse Translation-en_US
Ign http://us.archive.ubuntu.com trusty/restricted Translation-en_US
Ign http://us.archive.ubuntu.com trusty/universe Translation-en_US
Fetched 30.2 kB in 15s (1,980 B/s)
Reading package lists... Done
demo@ubuntudemo:~$
```

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
```

```
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
Found linux image: /boot/vmlinuz-3.13.0-105-generic
Found initrd image: /boot/initrd.img-3.13.0-105-generic
Found memtest86+ image: /memtest86+.elf
Found memtest86+ image: /memtest86+.bin
done
Setting up linux-image-extra-3.13.0-105-generic (3.13.0-105.152) ...
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 3.13.0-105-generi
/boot/vmlinuz-3.13.0-105-generic
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 3.13.0-105-generic
boot/vmlinuz-3.13.0-105-generic
update-initramfs: Generating /boot/initrd.img-3.13.0-105-generic
run-parts: executing /etc/kernel/postinst.d/update-notifier 3.13.0-105-generic
boot/vmlinuz-3.13.0-105-generic
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 3.13.0-105-generic
oot/vmlinuz-3.13.0-105-generic
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.2.0-27-generic
Found initrd image: /boot/initrd.img-4.2.0-27-generic
Found linux image: /boot/vmlinuz-3.13.0-105-generic
Found initrd image: /boot/initrd.img-3.13.0-105-generic
Found memtest86+ image: /memtest86+.elf
Found memtest86+ image: /memtest86+.bin
done
Setting up linux-image-generic (3.13.0.105.113) ...
Setting up linux-image-extra-virtual (3.13.0.105.113) ...
demo@ubuntudemo:~$
```

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.

```
Selecting previously unselected package liberror-perl.
Preparing to unpack .../liberror-perl_0.17-1.1_all.deb ...
Unpacking liberror-perl (0.17-1.1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a1.9.1-1ubuntu0.3_all.deb ...
Unpacking git-man (1:1.9.1-1ubuntu0.3) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.3_amd64.deb ...
Unpacking git (1:1.9.1-1ubuntu0.3) ...
Selecting previously unselected package cgroup-lite.
Preparing to unpack .../cgroup-lite_1.9_all.deb ...
Unpacking cgroup-lite (1.9) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on next reboot
Setting up libltdl7:amd64 (2.4.2-1.7ubuntu1) ...
Setting up libsystemd-journal0:amd64 (204-5ubuntu20.20) ...
Setting up aufs-tools (1:3.2+20130722-1.1) ...
Setting up docker-engine (1.12.3-0~trusty) ...
docker start/running, process 22612
Setting up liberror-perl (0.17-1.1) ...
Setting up git-man (1:1.9.1-1ubuntu0.3) ...
Setting up git (1:1.9.1-1ubuntu0.3) ...
Setting up cgroup-lite (1.9) ...
cgroup-lite start/running
Processing triggers for libc-bin (2.19-0ubuntu6.7) ...
Processing triggers for ureadahead (0.100.0-16) ...
demo@ubuntudemo:~$
```

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 version installed.

Return Value

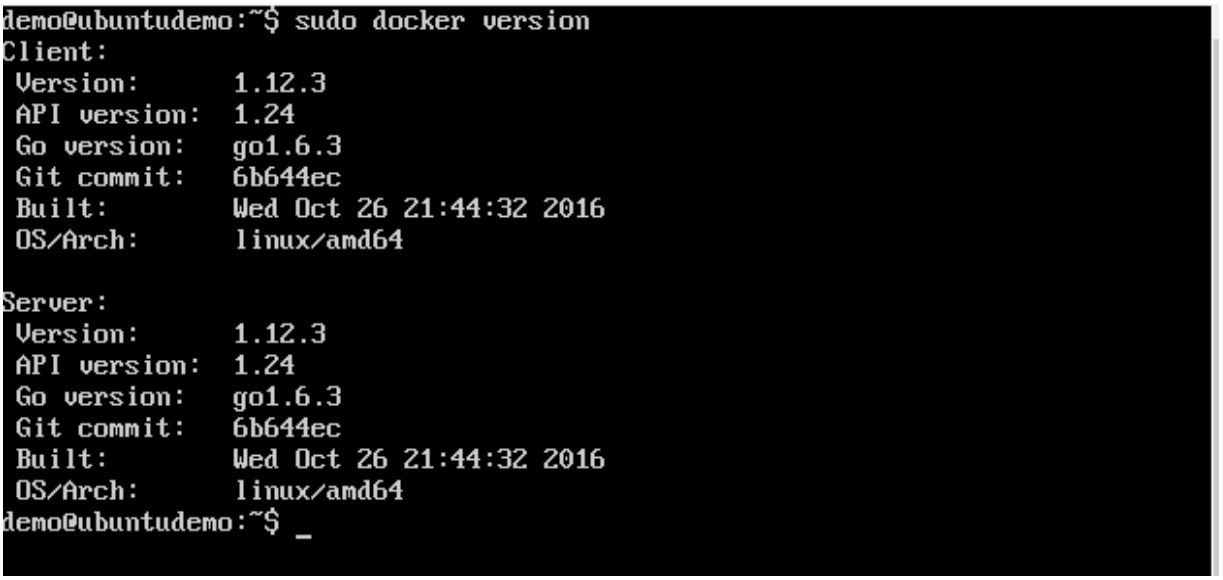
The output will provide the various details of the Docker version installed on the system.

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 detailed information 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:

```
 Backing Filesystem: extfs
 Dirs: 0
 Dirperm1 Supported: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge null host overlay
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Security Options: apparmor
Kernel Version: 4.2.0-27-generic
Operating System: Ubuntu 14.04.4 LTS
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 993.1 MiB
Name: ubuntudemo
ID: ECDA:IFR3:ZCQJ:FNXL:APJR:BT6Y:JJ75:FUE6:DNP5:PD7B:AOAD:YVB4
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
WARNING: No swap limit support
Insecure Registries:
 127.0.0.0/8
demo@ubuntudemo:~$ _
```

Docker for Windows

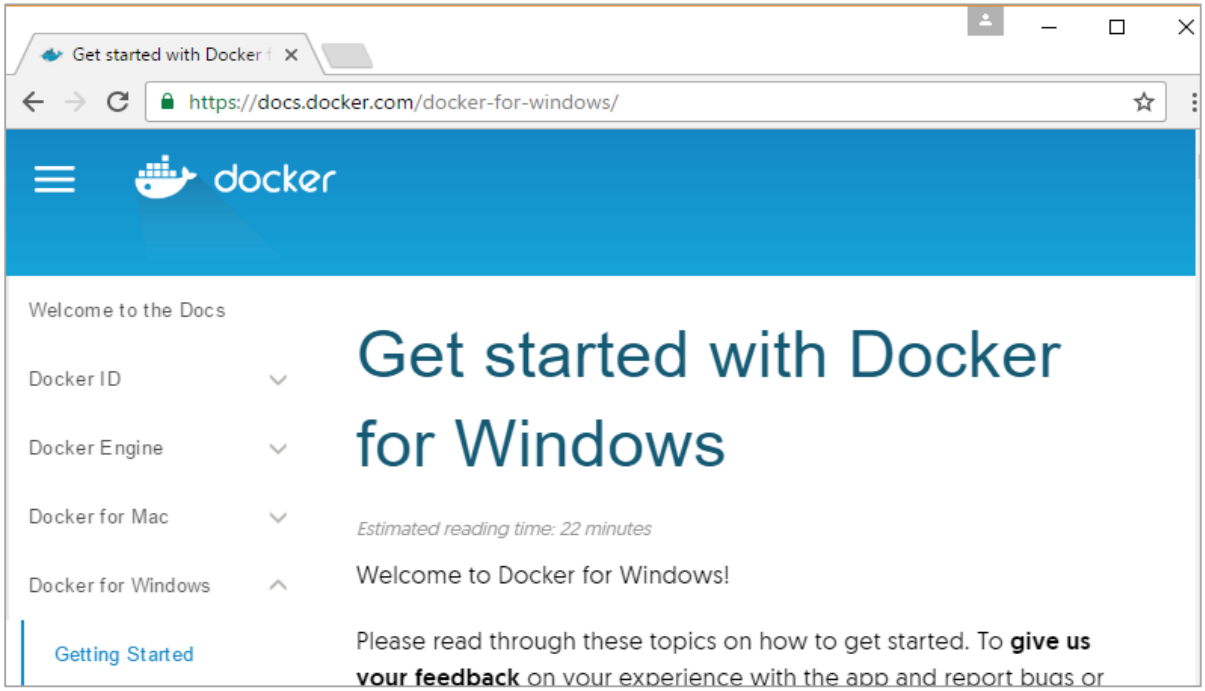
Docker has out-of-the-box support for Windows, but you need to have the following configuration in order to install Docker for Windows.

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/



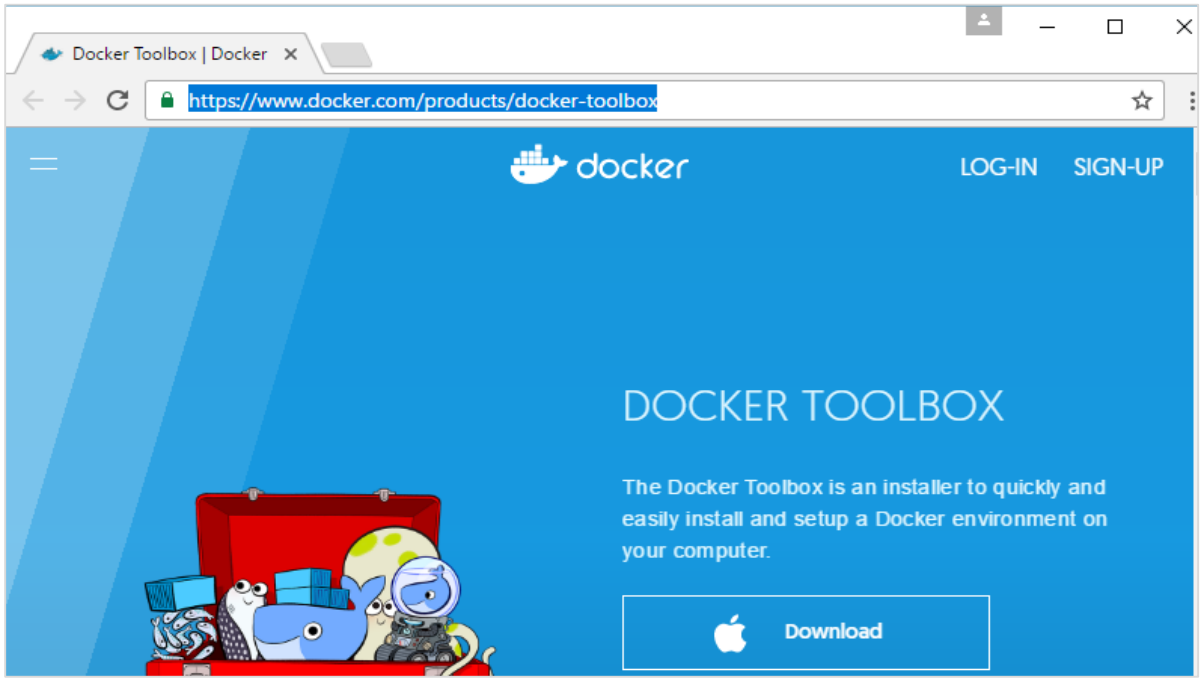
Docker ToolBox

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.

You can download Docker ToolBox from: https://www.docker.com/products/docker-toolbox



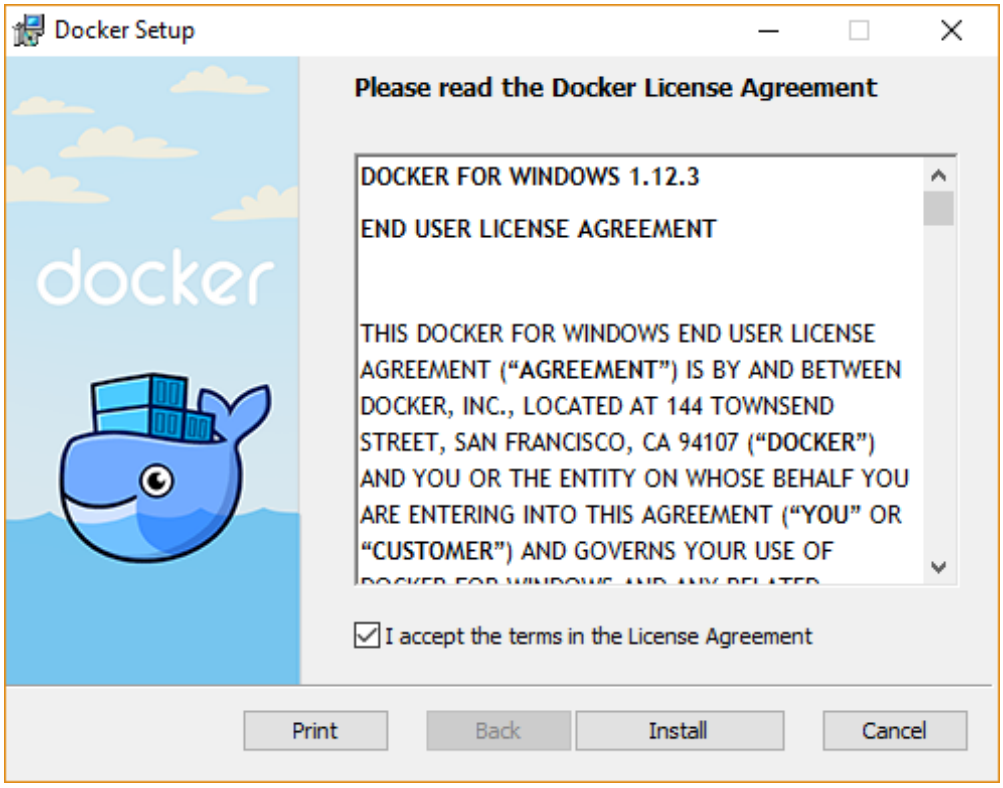
3. DOCKER – INSTALLATION

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 with the installation.



End of ebook preview
If you liked what you saw…
Buy it from our store @ **https://store.tutorialspoint**