Install Docker on Linux (Centos 7)

### **Install with yum**

1. Add the yum repo, It is already configured on your machine

# sudo tee /etc/yum.repos.d/docker.repo <<-EOF

[dockerrepo]

name=Docker Repository

baseurl=https://yum.dockerproject.org/repo/main/centos/7

enabled=1

gpgcheck=1

gpgkey=https://yum.dockerproject.org/gpg

EOF

1. Install the Docker package

$ sudo yum **install** docker-**engine**

1. Setup HTTP Proxy
2. First, create a systemd drop-in directory for the docker service:

**mkdir** /etc/systemd/**system**/docker.service.d

1. Now create a file called

/etc/systemd/system/docker.service.d/http-proxy.conf

**[Service]**

Environment="HTTP\_PROXY=http://<your cts userid>:<your CTS password>@proxy.coginizant.com:6040/" "NO\_PROXY=localhost,127.0.0.1,docker-registry.somecorporation.com"

1. Start the Docker daemon.

$ sudo service docker **start**

systemctl enable docker

1. Verify docker is installed correctly by running a test image in a container.

sudo docker run hello-world

To create the docker group and add your user:

1. Log into Centos as a user with sudo privileges.
2. Create the docker group and add your user.

sudo usermod -aG docker your\_username

1. Log out and log back in.

This ensures your user is running with the correct permissions.

1. Verify your work by running docker without sudo.

$ docker **run** ubuntu /bin/echo hello

1. To ensure Docker starts when you boot your system, do the following:

$ sudo chkconfig docker on

### **Installation**

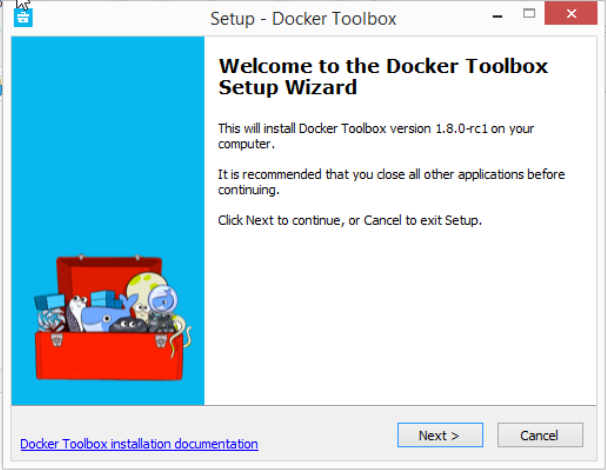
If you have VirtualBox running, you must shut it down before running the installer.

1. Download the Docker tool box from the below URL

https://www.docker.com/products/docker-toolbox

1. Click the installer link to download.
2. Install Docker Toolbox by double-clicking the installer.

The installer launches the “Setup - Docker Toolbox” dialog.



1. Press “Next” to install the toolbox.

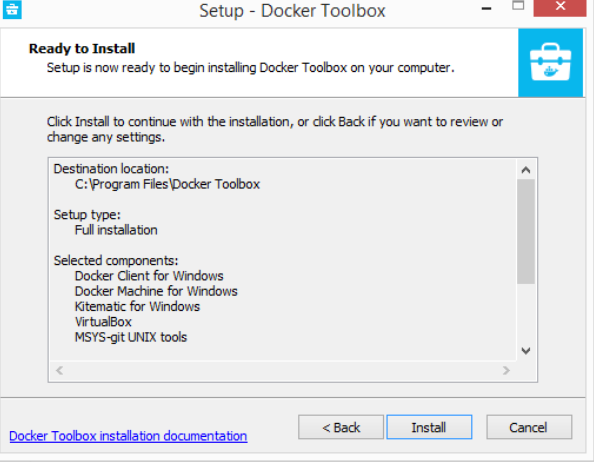
The installer presents you with options to customize the standard installation. By default, the standard Docker Toolbox installation:

* + installs executables for the Docker tools inC:\Program Files\Docker Toolbox
  + install VirtualBox; or updates any existing installation
  + adds a Docker Inc. folder to your program shortcuts
  + updates your PATH environment variable
  + adds desktop icons for the Docker Quickstart Terminal and Kitematic

This installation assumes the defaults are acceptable.

1. Press “Next” until you reach the “Ready to Install” page.

The system prompts you for your password.



1. Press “Install” to continue with the installation.

When it completes, the installer provides you with some information you can use to complete some common tasks.



1. Press “Finish” to exit.

## Running a Docker Container

To run a Docker container, you:

* create a new (or start an existing) Docker virtual machine
* switch your environment to your new VM
* use the docker client to create, load, and manage containers

Once you create a machine, you can reuse it as often as you like. Like any VirtualBox VM, it maintains its configuration between uses.

There are several ways to use the installed tools, from the Docker Quickstart Terminal or [from your shell](https://docs.docker.com/engine/installation/windows/#from-your-shell).

### **Using the Docker Quickstart Terminal**

1. Find the Docker Quickstart Terminal icon on your Desktop and double-click to launch it.

The application:

* + opens a terminal window
  + creates a default VM if it doesn’t exist, and starts the VM after
  + points the terminal environment to this VM

Once the launch completes, you can run docker commands.

1. Verify your setup succeeded by running the hello-world container.

$ docker run hello-world

Unable to find image 'hello-world:latest' locally

511136ea3c5a: Pull complete

31cbccb51277: Pull complete

e45a5af57b00: Pull complete

hello-world:latest: The image you are pulling has been verified.

Important: image verification **is** a tech preview feature **and** should **not** be

relied on to provide security.

Status: Downloaded newer image **for** hello-world:latest

Hello **from** Docker.

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.

2. The Docker daemon pulled the "hello-world" image **from** the Docker Hub.

(Assuming it was **not** already locally available.)

3. The Docker daemon created a new container **from** that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.

To **try** something more ambitious, you can run an Ubuntu container **with**:

$ docker run -it ubuntu bash

For more examples **and** ideas, visit:

http://docs.docker.com/userguide/

### **Using Docker from Windows Command Prompt (cmd.exe)**

Launch a Windows Command Prompt (cmd.exe).

The docker-machine command requires ssh.exe in your PATHenvironment variable. This .exe is in the MsysGit bin folder.

Add this to the %PATH% environment variable by running:

**set** **PATH**=%**PATH**%;"c:\Program Files (x86)\Git\bin"

Create a new Docker VM.

docker-machine **create** *--driver virtualbox my-default*

Creating VirtualBox VM...

Creating SSH **key**...

**Starting** VirtualBox VM...

**Starting** VM...

**To** see how **to** **connect** Docker **to** this machine, run: docker-machine env my-**default**

The command also creates a machine configuration in theC:\USERS\USERNAME\.docker\machine\machines directory. You only need to run the create command once. Then, you can usedocker-machine to start, stop, query, and otherwise manage the VM from the command line.

* List your available machines.

C:\Users\mary> docker-machine ls

NAME ACTIVE DRIVER STATE URL SWARM

my-**default** \* virtualbox Running tcp:*//192.168.99.101:2376*

If you have previously installed the deprecated Boot2Docker application or run the Docker Quickstart Terminal, you may have a dev VM as well.

* Get the environment commands for your new VM.

C:\Users\mary> docker-machine **env** --shell **cmd** my-default

* Connect your shell to the my-default machine.

C:\Users\mary> eval "$(docker-machine env my-default)"

* Run the hello-world container to verify your setup.

C:\Users\mary> docker **run** hello-world

### **Using Docker from PowerShell**

1. Launch a Windows PowerShell window.
2. Add ssh.exe to your PATH:

PS C:\Users\mary> $Env:Path = "${Env:Path};c:\Program Files (x86)\Git\bin"

1. Create a new Docker VM.

PS C:\Users\mary> docker-machine create --driver virtualbox my-**default**

1. List your available machines.

C:\Users\mary> docker-machine ls

NAME ACTIVE DRIVER STATE URL SWARM

my-**default** \* virtualbox Running tcp:*//192.168.99.101:2376*

1. Get the environment commands for your new VM.

C:\Users\mary> docker-machine env --shell powershell my-**default**

1. Connect your shell to the my-default machine.

C:\Users\mary> eval "$(docker-machine env my-default)"

1. Run the hello-world container to verify your setup.

C:\Users\mary> docker **run** hello-world