Get started with Docker Machine and a local VM

*Estimated reading time: 13 minutes*

Let’s take a look at using docker-machine to create, use and manage a Docker host inside of a local virtual machine.

Prerequisite Information

With the advent of [Docker for Mac](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-mac/) and [Docker for Windows](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-windows/) as replacements for [Docker Toolbox](https://docs.docker.com.zh.xy2401.com/v17.09/toolbox/overview/), we recommend that you use these for your primary Docker workflows. You can use these applications to run Docker natively on your local system without using Docker Machine at all. (See [Docker for Mac vs. Docker Toolbox](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-mac/docker-toolbox/) for an explanation on the Mac side.)

For now, however, if you want to create *multiple* local machines, you still need Docker Machine to create and manage machines for multi-node experimentation. Both Docker for Mac and Docker for Windows include the newest version of Docker Machine, so when you install either of these, you get docker-machine.

The new solutions come with their own native virtualization solutions rather than Oracle VirtualBox, so keep the following considerations in mind when using Machine to create local VMs.

* **Docker for Mac** - You can use docker-machine create with the virtualbox driver to create additional local machines.
* **Docker for Windows** - You can use docker-machine create with the hyperv driver to create additional local machines.

IF YOU ARE USING DOCKER FOR WINDOWS

Docker for Windows uses [Microsoft Hyper-V](https://msdn.microsoft.com/en-us/virtualization/hyperv_on_windows/windows_welcome) for virtualization, and Hyper-V is not compatible with Oracle VirtualBox. Therefore, you cannot run the two solutions simultaneously. But you can still use docker-machine to create more local VMs by using the Microsoft Hyper-V driver.

The prerequisites are:

* Have Docker for Windows installed, and running (which requires that virtualization and Hyper-V are enabled, as described in [What to know before you install Docker for Windows](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-windows/install/#what-to-know-before-you-install)).
* Set up the Hyper-V driver to use an external virtual network switch See the [Docker Machine driver for Microsoft Hyper-V](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/hyper-v/) topic, which includes an [example](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/hyper-v/#example) of how to do this.

IF YOU ARE USING DOCKER FOR MAC

Docker for Mac uses [HyperKit](https://github.com/docker/HyperKit/), a lightweight macOS virtualization solution built on top of the [Hypervisor.framework](https://developer.apple.com/reference/hypervisor) in macOS 10.10 Yosemite and higher.

Currently, there is no docker-machine create driver for HyperKit, so you will use virtualbox driver to create local machines. (See the [Docker Machine driver for Oracle VirtualBox](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/virtualbox/).) Note that you can run both HyperKit and Oracle VirtualBox on the same system. To learn more, see [Docker for Mac vs. Docker Toolbox](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-mac/docker-toolbox/).

* Make sure you have [the latest VirtualBox](https://www.virtualbox.org/wiki/Downloads) correctly installed on your system (either as part of an earlier Toolbox install, or manual install).

IF YOU ARE USING DOCKER TOOLBOX

Docker for Mac and Docker for Windows both require newer versions of their respective operating systems, so users with older OS versions must use Docker Toolbox.

* If you are using Docker Toolbox on either Mac or an older version Windows system (without Hyper-V), you will use the virtualbox driver to create a local machine based on Oracle [VirtualBox](https://www.virtualbox.org/" \t "_blank). (See the [Docker Machine driver for Oracle VirtualBox](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/virtualbox/).)
* If you are using Docker Toolbox on a Windows system that has Hyper-V but cannot run Docker for Windows (for example Windows 8 Pro), you must use the hyperv driver to create local machines. (See the [Docker Machine driver for Microsoft Hyper-V](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/hyper-v/).)
* Make sure you have [the latest VirtualBox](https://www.virtualbox.org/wiki/Downloads) correctly installed on your system. If you used [Toolbox](https://www.docker.com/products/docker-toolbox) or [Docker for Windows](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-windows/) to install Docker Machine, VirtualBox is automatically installed.
* If you used the Quickstart Terminal to launch your first machine and set your terminal environment to point to it, a default machine was automatically created. If this is the case, you can still follow along with these steps, but create another machine and name it something other than “default” (e.g., staging or sandbox).

Use Machine to run Docker containers

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.

The examples here show how to create and start a machine, run Docker commands, and work with containers.

Create a machine

1. Open a command shell or terminal window.

These command examples shows a Bash shell. For a different shell, such as C Shell, the same commands are the same except where noted.

1. Use docker-machine ls to list available machines.

In this example, no machines have been created yet.

$ docker-machine ls

NAME ACTIVE DRIVER STATE URL SWARM DOCKER ERRORS

1. Create a machine.

Run the docker-machine create command, pass the appropriate driver to the --driver flag and provide a machine name. If this is your first machine, name it default as shown in the example. If you already have a “default” machine, choose another name for this new machine.

* + If you are using Toolbox on Mac, Toolbox on older Windows systems without Hyper-V, or Docker for Mac, use virtualbox as the driver, as shown in this example. (The Docker Machine VirtualBox driver reference is [here](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/virtualbox/).) (See [prerequisites](https://docs.docker.com.zh.xy2401.com/v17.09/machine/get-started/#prerequisite-information) above to learn more.)
  + On Docker for Windows systems that support Hyper-V, use the hyperv driver as shown in the [Docker Machine Microsoft Hyper-V driver reference](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/hyper-v/) and follow the [example](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/hyper-v/#example), which shows how to use an external network switch and provides the flags for the full command. (See [prerequisites](https://docs.docker.com.zh.xy2401.com/v17.09/machine/get-started/#prerequisite-information) above to learn more.)
  + $ docker-machine create --driver virtualbox default
  + Running pre-create checks...
  + Creating machine...
  + (staging) Copying /Users/ripley/.docker/machine/cache/boot2docker.iso to /Users/ripley/.docker/machine/machines/default/boot2docker.iso...
  + (staging) Creating VirtualBox VM...
  + (staging) Creating SSH key...
  + (staging) Starting the VM...
  + (staging) Waiting for an IP...
  + Waiting for machine to be running, this may take a few minutes...
  + Machine is running, waiting for SSH to be available...
  + Detecting operating system of created instance...
  + Detecting the provisioner...
  + Provisioning with boot2docker...
  + Copying certs to the local machine directory...
  + Copying certs to the remote machine...
  + Setting Docker configuration on the remote daemon...
  + Checking connection to Docker...
  + Docker is up and running!
  + To see how to connect Docker to this machine, run: docker-machine env default

This command downloads a lightweight Linux distribution ([boot2docker](https://github.com/boot2docker/boot2docker)) with the Docker daemon installed, and creates and starts a VirtualBox VM with Docker running.

1. List available machines again to see your newly minted machine.
2. $ docker-machine ls
3. NAME ACTIVE DRIVER STATE URL SWARM DOCKER ERRORS
4. default \* virtualbox Running tcp://192.168.99.187:2376 v1.9.1
5. Get the environment commands for your new VM.

As noted in the output of the docker-machine create command, you need to tell Docker to talk to the new machine. You can do this with the docker-machine env command.

$ docker-machine env default

export DOCKER\_TLS\_VERIFY="1"

export DOCKER\_HOST="tcp://172.16.62.130:2376"

export DOCKER\_CERT\_PATH="/Users/<yourusername>/.docker/machine/machines/default"

export DOCKER\_MACHINE\_NAME="default"

# Run this command to configure your shell:

# eval "$(docker-machine env default)"

1. Connect your shell to the new machine.
2. $ eval "$(docker-machine env default)"

**Note**: If you are using fish, or a Windows shell such as Powershell/cmd.exe the above method will not work as described. Instead, see [the env command’s documentation](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/env/) to learn how to set the environment variables for your shell.

This sets environment variables for the current shell that the Docker client will read which specify the TLS settings. You need to do this each time you open a new shell or restart your machine. (See also, how to [unset environment variables in the current shell](https://docs.docker.com.zh.xy2401.com/v17.09/machine/get-started/#unset-environment-variables-in-the-current-shell).)

You can now run Docker commands on this host.

Run containers and experiment with Machine commands

Run a container with docker run to verify your set up.

1. Use docker run to download and run busybox with a simple ‘echo’ command.
2. $ docker run busybox echo hello world
3. Unable to find image 'busybox' locally
4. Pulling repository busybox
5. e72ac664f4f0: Download complete
6. 511136ea3c5a: Download complete
7. df7546f9f060: Download complete
8. e433a6c5b276: Download complete
9. hello world
10. Get the host IP address.

Any exposed ports are available on the Docker host’s IP address, which you can get using the docker-machine ip command:

$ docker-machine ip default

192.168.99.100

1. Run a [Nginx](https://www.nginx.com/) webserver in a container with the following command:
2. $ docker run -d -p 8000:80 nginx

When the image is finished pulling, you can hit the server at port 8000 on the IP address given to you by docker-machine ip. For instance:

$ curl $(docker-machine ip default):8000

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

body {

width: 35em;

margin: 0 auto;

font-family: Tahoma, Verdana, Arial, sans-serif;

}

</style>

</head>

<body>

<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and

working. Further configuration is required.</p>

<p>For online documentation and support please refer to

<a href="http://nginx.org/">nginx.org</a>.<br/>

Commercial support is available at

<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>

</body>

</html>

You can create and manage as many local VMs running Docker as you please; just run docker-machine create again. All created machines will appear in the output of docker-machine ls.

Start and stop machines

If you are finished using a host for the time being, you can stop it with docker-machine stop and later start it again with docker-machine start.

$ docker-machine stop default

$ docker-machine start default

Operate on machines without specifying the name

Some docker-machine commands will assume that the given operation should be run on a machine named default (if it exists) if no machine name is specified. Because using a local VM named default is such a common pattern, this allows you to save some typing on the most frequently used Machine commands.

For example:

$ docker-machine stop

Stopping "default"....

Machine "default" was stopped.

$ docker-machine start

Starting "default"...

(default) Waiting for an IP...

Machine "default" was started.

Started machines may have new IP addresses. You may need to re-run the `docker-machine env` command.

$ eval $(docker-machine env)

$ docker-machine ip

192.168.99.100

Commands that follow this style are:

- `docker-machine config`

- `docker-machine env`

- `docker-machine inspect`

- `docker-machine ip`

- `docker-machine kill`

- `docker-machine provision`

- `docker-machine regenerate-certs`

- `docker-machine restart`

- `docker-machine ssh`

- `docker-machine start`

- `docker-machine status`

- `docker-machine stop`

- `docker-machine upgrade`

- `docker-machine url`

For machines other than default, and commands other than those listed above, you must always specify the name explicitly as an argument.

Unset environment variables in the current shell

You might want to use the current shell to connect to a different Docker Engine. This would be the case if, for example, you are [running Docker for Mac concurrent with Docker Toolbox](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-mac/docker-toolbox/) and want to talk to two different Docker Engines, or running swarms on Docker Cloud and want to [switch between managing the swarm and using Docker hosts](https://docs.docker.com.zh.xy2401.com/v17.09/docker-cloud/cloud-swarm/connect-to-swarm/#switch-between-your-swarm-and-docker-hosts-in-the-same-shell). In both scenarios, you have the option to switch the environment for the current shell to talk to different Docker engines.

1. Run env|grep DOCKER to check whether DOCKER environment variables are set.
2. $ env | grep DOCKER
3. DOCKER\_HOST=tcp://192.168.99.100:2376
4. DOCKER\_MACHINE\_NAME=default
5. DOCKER\_TLS\_VERIFY=1
6. DOCKER\_CERT\_PATH=/Users/victoriabialas/.docker/machine/machines/default

If it returns output (as shown in the example), you can unset the DOCKER environment variables.

1. Use one of two methods to unset DOCKER environment variables in the current shell.
   * Run the unset command on the following DOCKER environment variables.
   * unset DOCKER\_TLS\_VERIFY
   * unset DOCKER\_CERT\_PATH
   * unset DOCKER\_MACHINE\_NAME
   * unset DOCKER\_HOST
   * Alternatively, run a shortcut command docker-machine env -u to show the command you need to run to unset all DOCKER variables:
   * $ docker-machine env -u
   * unset DOCKER\_TLS\_VERIFY
   * unset DOCKER\_HOST
   * unset DOCKER\_CERT\_PATH
   * unset DOCKER\_MACHINE\_NAME
   * # Run this command to configure your shell:
   * # eval $(docker-machine env -u)

Run eval $(docker-machine env -u) to unset all DOCKER variables in the current shell.

1. Now, after running either of the above commands, this command should return no output.
2. $ env | grep DOCKER

If you are running Docker for Mac, you can run Docker commands to talk to the Docker Engine installed with that app.

If you are running swarms on Docker Cloud, you can re-run the export command you used to connect to the swarm.

Since [Docker for Windows is incompatible with Toolbox](https://docs.docker.com.zh.xy2401.com/v17.09/docker-for-windows/install/#what-to-know-before-you-install), this scenario isn’t applicable because Docker for Windows uses the Docker Engine and Docker Machine that come with it.

Start local machines on startup

In order to ensure that the Docker client is automatically configured at the start of each shell session, some users like to embed eval $(docker-machine env default) in their shell profiles (e.g., the ~/.bash\_profile file). However, this fails if the default machine is not running. If desired, you can configure your system to start the default machine automatically.

Here is an example of how to configure this on macOS.

Create a file called com.docker.machine.default.plist under ~/Library/LaunchAgents with the following content:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<key>EnvironmentVariables</key>

<dict>

<key>PATH</key>

<string>/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin</string>

</dict>

<key>Label</key>

<string>com.docker.machine.default</string>

<key>ProgramArguments</key>

<array>

<string>/usr/local/bin/docker-machine</string>

<string>start</string>

<string>default</string>

</array>

<key>RunAtLoad</key>

<true/>

</dict>

</plist>

You can change the default string above to make this LaunchAgent start any machine(s) you desire.

Where to go next

* Provision multiple Docker hosts [on your cloud provider](https://docs.docker.com.zh.xy2401.com/v17.09/machine/get-started-cloud/)
* [Understand Machine concepts](https://docs.docker.com.zh.xy2401.com/v17.09/machine/concepts/)
* [Docker Machine list of reference pages for all supported drivers](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/)
* [Docker Machine driver for Oracle VirtualBox](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/virtualbox/)
* [Docker Machine driver for Microsoft Hyper-V](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/hyper-v/)
* [docker-machine command line reference](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/)

# Docker Machine command-line reference

*Estimated reading time: 1 minute*

* [active](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/active/)
* [config](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/config/)
* [create](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/create/)
* [env](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/env/)
* [help](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/help/)
* [inspect](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/inspect/)
* [ip](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/ip/)
* [kill](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/kill/)
* [ls](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/ls/)
* [mount](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/mount/)
* [provision](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/provision/)
* [regenerate-certs](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/regenerate-certs/)
* [restart](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/restart/)
* [rm](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/rm/)
* [scp](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/scp/)
* [ssh](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/ssh/)
* [start](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/start/)
* [status](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/status/)
* [stop](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/stop/)
* [upgrade](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/upgrade/)
* [url](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/url/)