docker-machine create

*Estimated reading time: 9 minutes*

Create a machine. Requires the --driver flag to indicate which provider (VirtualBox, DigitalOcean, AWS, etc.) the machine should be created on, and an argument to indicate the name of the created machine.

**Looking for the full list of available drivers?**

For a full list of drivers that work with docker-machine create and information on how to use them, see [Machine drivers](https://docs.docker.com.zh.xy2401.com/v17.09/machine/drivers/).

Example

Here is an example of using the --virtualbox driver to create a machine called dev.

$ docker-machine create --driver virtualbox dev

Creating CA: /home/username/.docker/machine/certs/ca.pem

Creating client certificate: /home/username/.docker/machine/certs/cert.pem

Image cache does not exist, creating it at /home/username/.docker/machine/cache...

No default boot2docker iso found locally, downloading the latest release...

Downloading https://github.com/boot2docker/boot2docker/releases/download/v1.6.2/boot2docker.iso to /home/username/.docker/machine/cache/boot2docker.iso...

Creating VirtualBox VM...

Creating SSH key...

Starting VirtualBox VM...

Starting VM...

To see how to connect Docker to this machine, run: docker-machine env dev

Accessing driver-specific flags in the help text

The docker-machine create command has some flags which are applicable to all drivers. These largely control aspects of Machine’s provisioning process (including the creation of Docker Swarm containers) that the user may wish to customize.

$ docker-machine create

Docker Machine Version: 0.5.0 (45e3688)

Usage: docker-machine create [OPTIONS] [arg...]

Create a machine.

Run 'docker-machine create --driver name' to include the create flags for that driver in the help text.

Options:

--driver, -d "none" Driver to create machine with.

--engine-install-url "https://get.docker.com" Custom URL to use for engine installation [$MACHINE\_DOCKER\_INSTALL\_URL]

--engine-opt [--engine-opt option --engine-opt option] Specify arbitrary flags to include with the created engine in the form flag=value

--engine-insecure-registry [--engine-insecure-registry option --engine-insecure-registry option] Specify insecure registries to allow with the created engine

--engine-registry-mirror [--engine-registry-mirror option --engine-registry-mirror option] Specify registry mirrors to use [$ENGINE\_REGISTRY\_MIRROR]

--engine-label [--engine-label option --engine-label option] Specify labels for the created engine

--engine-storage-driver Specify a storage driver to use with the engine

--engine-env [--engine-env option --engine-env option] Specify environment variables to set in the engine

--swarm Configure Machine with Swarm

--swarm-image "swarm:latest" Specify Docker image to use for Swarm [$MACHINE\_SWARM\_IMAGE]

--swarm-master Configure Machine to be a Swarm master

--swarm-discovery Discovery service to use with Swarm

--swarm-strategy "spread" Define a default scheduling strategy for Swarm

--swarm-opt [--swarm-opt option --swarm-opt option] Define arbitrary flags for swarm

--swarm-host "tcp://0.0.0.0:3376" ip/socket to listen on for Swarm master

--swarm-addr addr to advertise for Swarm (default: detect and use the machine IP)

--swarm-experimental Enable Swarm experimental features

Additionally, drivers can specify flags that Machine can accept as part of their plugin code. These allow users to customize the provider-specific parameters of the created machine, such as size (--amazonec2-instance-type m1.medium), geographical region (--amazonec2-region us-west-1), and so on.

To see the provider-specific flags, simply pass a value for --driver when invoking the create help text.

$ docker-machine create --driver virtualbox --help

Usage: docker-machine create [OPTIONS] [arg...]

Create a machine.

Run 'docker-machine create --driver name' to include the create flags for that driver in the help text.

Options:

--driver, -d "none" Driver to create machine with.

--engine-env [--engine-env option --engine-env option] Specify environment variables to set in the engine

--engine-insecure-registry [--engine-insecure-registry option --engine-insecure-registry option] Specify insecure registries to allow with the created engine

--engine-install-url "https://get.docker.com" Custom URL to use for engine installation [$MACHINE\_DOCKER\_INSTALL\_URL]

--engine-label [--engine-label option --engine-label option] Specify labels for the created engine

--engine-opt [--engine-opt option --engine-opt option] Specify arbitrary flags to include with the created engine in the form flag=value

--engine-registry-mirror [--engine-registry-mirror option --engine-registry-mirror option] Specify registry mirrors to use [$ENGINE\_REGISTRY\_MIRROR]

--engine-storage-driver Specify a storage driver to use with the engine

--swarm Configure Machine with Swarm

--swarm-addr addr to advertise for Swarm (default: detect and use the machine IP)

--swarm-discovery Discovery service to use with Swarm

--swarm-experimental Enable Swarm experimental features

--swarm-host "tcp://0.0.0.0:3376" ip/socket to listen on for Swarm master

--swarm-image "swarm:latest" Specify Docker image to use for Swarm [$MACHINE\_SWARM\_IMAGE]

--swarm-master Configure Machine to be a Swarm master

--swarm-opt [--swarm-opt option --swarm-opt option] Define arbitrary flags for swarm

--swarm-strategy "spread" Define a default scheduling strategy for Swarm

--virtualbox-boot2docker-url The URL of the boot2docker image. Defaults to the latest available version [$VIRTUALBOX\_BOOT2DOCKER\_URL]

--virtualbox-cpu-count "1" number of CPUs for the machine (-1 to use the number of CPUs available) [$VIRTUALBOX\_CPU\_COUNT]

--virtualbox-disk-size "20000" Size of disk for host in MB [$VIRTUALBOX\_DISK\_SIZE]

--virtualbox-host-dns-resolver Use the host DNS resolver [$VIRTUALBOX\_HOST\_DNS\_RESOLVER]

--virtualbox-dns-proxy Proxy all DNS requests to the host [$VIRTUALBOX\_DNS\_PROXY]

--virtualbox-hostonly-cidr "192.168.99.1/24" Specify the Host Only CIDR [$VIRTUALBOX\_HOSTONLY\_CIDR]

--virtualbox-hostonly-nicpromisc "deny" Specify the Host Only Network Adapter Promiscuous Mode [$VIRTUALBOX\_HOSTONLY\_NIC\_PROMISC]

--virtualbox-hostonly-nictype "82540EM" Specify the Host Only Network Adapter Type [$VIRTUALBOX\_HOSTONLY\_NIC\_TYPE]

--virtualbox-import-boot2docker-vm The name of a Boot2Docker VM to import

--virtualbox-memory "1024" Size of memory for host in MB [$VIRTUALBOX\_MEMORY\_SIZE]

--virtualbox-no-share Disable the mount of your home directory

You may notice that some flags specify environment variables that they are associated with as well (located to the far left hand side of the row). If these environment variables are set when docker-machine create is invoked, Docker Machine will use them for the default value of the flag.

Specifying configuration options for the created Docker engine

As part of the process of creation, Docker Machine installs Docker and configures it with some sensible defaults. For instance, it allows connection from the outside world over TCP with TLS-based encryption and defaults to AUFS as the [storage driver](https://docs.docker.com.zh.xy2401.com/v17.09/engine/reference/commandline/dockerd/#daemon-storage-driver-option) when available.

There are several cases where the user might want to set options for the created Docker engine (also known as the Docker *daemon*) themselves. For example, they may want to allow connection to a [registry](https://docs.docker.com.zh.xy2401.com/v17.09/registry/) that they are running themselves using the --insecure-registry flag for the daemon. Docker Machine supports the configuration of such options for the created engines via the create command flags which begin with --engine.

Note that Docker Machine simply sets the configured parameters on the daemon and does not set up any of the “dependencies” for you. For instance, if you specify that the created daemon should use btrfs as a storage driver, you still must ensure that the proper dependencies are installed, the BTRFS filesystem has been created, and so on.

The following is an example usage:

$ docker-machine create -d virtualbox \

--engine-label foo=bar \

--engine-label spam=eggs \

--engine-storage-driver overlay \

--engine-insecure-registry registry.myco.com \

foobarmachine

This will create a virtual machine running locally in Virtualbox which uses the overlay storage backend, has the key-value pairs foo=bar and spam=eggs as labels on the engine, and allows pushing / pulling from the insecure registry located at registry.myco.com. You can verify much of this by inspecting the output of docker info:

$ eval $(docker-machine env foobarmachine)

$ docker info

Containers: 0

Images: 0

Storage Driver: overlay

...

Name: foobarmachine

...

Labels:

foo=bar

spam=eggs

provider=virtualbox

The supported flags are as follows:

* --engine-insecure-registry: Specify [insecure registries](https://docs.docker.com.zh.xy2401.com/v17.09/engine/reference/commandline/cli/#insecure-registries) to allow with the created engine
* --engine-registry-mirror: Specify [registry mirrors](https://docs.docker.com.zh.xy2401.com/v17.09/registry/recipes/mirror/) to use
* --engine-label: Specify [labels](https://docs.docker.com.zh.xy2401.com/v17.09/engine/userguide/labels-custom-metadata/#daemon-labels) for the created engine
* --engine-storage-driver: Specify a [storage driver](https://docs.docker.com.zh.xy2401.com/v17.09/engine/reference/commandline/cli/#daemon-storage-driver-option) to use with the engine

If the engine supports specifying the flag multiple times (such as with --label), then so does Docker Machine.

In addition to this subset of daemon flags which are directly supported, Docker Machine also supports an additional flag, --engine-opt, which can be used to specify arbitrary daemon options with the syntax --engine-opt flagname=value. For example, to specify that the daemon should use 8.8.8.8 as the DNS server for all containers, and always use the syslog [log driver](https://docs.docker.com.zh.xy2401.com/v17.09/engine/reference/run/#logging-drivers-log-driver) you could run the following create command:

$ docker-machine create -d virtualbox \

--engine-opt dns=8.8.8.8 \

--engine-opt log-driver=syslog \

gdns

Additionally, Docker Machine supports a flag, --engine-env, which can be used to specify arbitrary environment variables to be set within the engine with the syntax --engine-env name=value. For example, to specify that the engine should use example.com as the proxy server, you could run the following create command:

$ docker-machine create -d virtualbox \

--engine-env HTTP\_PROXY=http://example.com:8080 \

--engine-env HTTPS\_PROXY=https://example.com:8080 \

--engine-env NO\_PROXY=example2.com \

proxbox

Specifying Docker Swarm options for the created machine

In addition to being able to configure Docker Engine options as listed above, you can use Machine to specify how the created Swarm master should be configured. There is a --swarm-strategy flag, which you can use to specify the [scheduling strategy](https://docs.docker.com.zh.xy2401.com/v17.09/swarm/scheduler/strategy/) which Docker Swarm should use (Machine defaults to the spread strategy). There is also a general purpose --swarm-opt option which works similar to the aforementioned --engine-opt option, except that it specifies options for the swarm manage command (used to boot a master node) instead of the base command. You can use this to configure features that power users might be interested in, such as configuring the heartbeat interval or Swarm’s willingness to over-commit resources. There is also the --swarm-experimental flag, that allows you to access [experimental features](https://github.com/docker/swarm/tree/master/experimental) in Docker Swarm.

If you’re not sure how to configure these options, it is best to not specify configuration at all. Docker Machine will choose sensible defaults for you and you won’t have to worry about it.

Example create:

$ docker-machine create -d virtualbox \

--swarm \

--swarm-master \

--swarm-discovery token://<token> \

--swarm-strategy binpack \

--swarm-opt heartbeat=5s \

upbeat

This will set the swarm scheduling strategy to “binpack” (pack in containers as tightly as possible per host instead of spreading them out), and the “heartbeat” interval to 5 seconds.

Pre-create check

Since many drivers require a certain set of conditions to be in place before they can successfully perform a create (e.g. VirtualBox should be installed, or the provided API credentials should be valid), Docker Machine has a “pre-create check” which is specified at the driver level.

If this pre-create check succeeds, Docker Machine will proceed with the creation as normal. If the pre-create check fails, the Docker Machine process will exit with status code 3 to indicate that the source of the non-zero exit was the pre-create check failing.