docker-machine env

*Estimated reading time: 3 minutes*

Set environment variables to dictate that docker should run a command against a particular machine.

$ docker-machine env --help

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

Display the commands to set up the environment for the Docker client

Description:

Argument is a machine name.

Options:

--swarm Display the Swarm config instead of the Docker daemon

--shell Force environment to be configured for a specified shell: [fish, cmd, powershell, tcsh], default is sh/bash

--unset, -u Unset variables instead of setting them

--no-proxy Add machine IP to NO\_PROXY environment variable

docker-machine env machinename will print out export commands which can be run in a subshell. Running docker-machine env -u will print unset commands which reverse this effect.

$ env | grep DOCKER

$ eval "$(docker-machine env dev)"

$ env | grep DOCKER

DOCKER\_HOST=tcp://192.168.99.101:2376

DOCKER\_CERT\_PATH=/Users/nathanleclaire/.docker/machines/.client

DOCKER\_TLS\_VERIFY=1

DOCKER\_MACHINE\_NAME=dev

$ # If you run a docker command, now it will run against that host.

$ eval "$(docker-machine env -u)"

$ env | grep DOCKER

$ # The environment variables have been unset.

The output described above is intended for the shells bash and zsh (if you’re not sure which shell you’re using, there’s a very good possibility that it’s bash). However, these are not the only shells which Docker Machine supports. Docker Machine detects the shells available in your environment and lists them. Docker supports bash, cmd, powershell, and emacs.

If you are using fish and the SHELL environment variable is correctly set to the path where fish is located, docker-machine env name will print out the values in the format which fish expects:

set -x DOCKER\_TLS\_VERIFY 1;

set -x DOCKER\_CERT\_PATH "/Users/nathanleclaire/.docker/machine/machines/overlay";

set -x DOCKER\_HOST tcp://192.168.99.102:2376;

set -x DOCKER\_MACHINE\_NAME overlay

# Run this command to configure your shell:

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

If you are on Windows and using either PowerShell or cmd.exe, docker-machine env Docker Machine should now detect your shell automatically. If the automatic detection does not work, you can still override it using the --shell flag for docker-machine env.

For PowerShell:

$ docker-machine.exe env --shell powershell dev

$Env:DOCKER\_TLS\_VERIFY = "1"

$Env:DOCKER\_HOST = "tcp://192.168.99.101:2376"

$Env:DOCKER\_CERT\_PATH = "C:\Users\captain\.docker\machine\machines\dev"

$Env:DOCKER\_MACHINE\_NAME = "dev"

# Run this command to configure your shell:

# docker-machine.exe env --shell=powershell dev | Invoke-Expression

For cmd.exe:

$ docker-machine.exe env --shell cmd dev

set DOCKER\_TLS\_VERIFY=1

set DOCKER\_HOST=tcp://192.168.99.101:2376

set DOCKER\_CERT\_PATH=C:\Users\captain\.docker\machine\machines\dev

set DOCKER\_MACHINE\_NAME=dev

# Run this command to configure your shell: copy and paste the above values into your command prompt

**Tip:** 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).

Excluding the created machine from proxies

The env command supports a --no-proxy flag which will ensure that the created machine’s IP address is added to the [NO\_PROXY/no\_proxy environment variable](https://wiki.archlinux.org/index.php/Proxy_settings).

This is useful when using docker-machine with a local VM provider (e.g. virtualbox or vmwarefusion) in network environments where an HTTP proxy is required for internet access.

$ docker-machine env --no-proxy default

export DOCKER\_TLS\_VERIFY="1"

export DOCKER\_HOST="tcp://192.168.99.104:2376"

export DOCKER\_CERT\_PATH="/Users/databus23/.docker/machine/certs"

export DOCKER\_MACHINE\_NAME="default"

export NO\_PROXY="192.168.99.104"

# Run this command to configure your shell:

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

You may also want to visit the [documentation on setting HTTP\_PROXY for the created daemon using the --engine-env flag for docker-machine create](https://docs.docker.com.zh.xy2401.com/v17.09/machine/reference/create/#specifying-configuration-options-for-the-created-docker-engine).