**Reference: https://github.com/docker/machine/issues/4441**

**Here is the solution**

**https://github.com/docker/machine/releases/download/v0.15.0/docker-machine-Windows-x86\_64.exe**

**Save the downloaded file to your existing directory containing docker-machine.exe.**

For my system this is the location for docker-machine.exe  
/c/Program Files/Docker/Docker/Resources/bin/docker-machine.exe

**Backup the old file and replace it file with the new one.**

cp docker-machine.exe docker-machine.014.exe

**Rename the downloaded filename to docker-machine.exe**

mv docker-machine-Windows-x86\_64.exe docker-machine.exe

**Build Instructions**

1. Create virtual switch in Hyper-V manager named myswitch
2. Request Docker to create a VM named myvm1

docker-machine create -d hyperv --hyperv-virtual-switch "myswitch" myvm1

**Results**

docker-machine create -d hyperv --hyperv-virtual-switch "myswitch" myvm1

Running pre-create checks...

(myvm1) Image cache directory does not exist, creating it at C:\Users\Trey Brister\.docker\machine\cache...

(myvm1) No default Boot2Docker ISO found locally, downloading the latest release...

(myvm1) Latest release for github.com/boot2docker/boot2docker is v18.05.0-ce

(myvm1) Downloading C:\Users\Trey Brister\.docker\machine\cache\boot2docker.iso from https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso...

(myvm1) 0%....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%

Creating machine...

(myvm1) Copying C:\Users\Trey Brister\.docker\machine\cache\boot2docker.iso to C:\Users\Trey Brister\.docker\machine\machines\myvm1\boot2docker.iso...

(myvm1) Creating SSH key...

(myvm1) Creating VM...

(myvm1) Using switch "myswitch"

(myvm1) Creating VHD

(myvm1) Starting VM...

(myvm1) Waiting for host to start...

Waiting for machine to be running, this may take a few minutes...

Detecting operating system of created instance...

Waiting for SSH to be available...

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 your Docker Client to the Docker Engine running on this virtual machine, run: C:\Users\rmallaya\bin\docker-machine.exe env myvm1

-------------------------------------------------------------

**C:\Users\rmallaya\bin\docker-machine.exe env myvm1**

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

$Env:DOCKER\_HOST = "tcp://192.168.0.102:2376"

$Env:DOCKER\_CERT\_PATH = "C:\Users\rmallaya\.docker\machine\machines\myvm1"

$Env:DOCKER\_MACHINE\_NAME = "myvm1"

$Env:COMPOSE\_CONVERT\_WINDOWS\_PATHS = "true"

# Run this command to configure your shell:

# & "C:\Users\rmallaya\bin\docker-machine.exe" env myvm1 | Invoke-Expression