OpenDroneMap with Docker Toolbox on Windows

June 25, 2018

Install Docker Toolbox on Windows

https://docs.docker.com/toolbox/toolbox_install_windows/

Docker Toolbox provides a way to use Docker on Windows systems that do not meet minimal system requirements for the Docker for Windows app.

Get Docker Toolbox for Windows:

https://download.docker.com/win/stable/DockerToolbox.exe

What you get and how it works

Docker Toolbox includes the following Docker tools:

- Docker CLI client for running Docker Engine to create images and containers
- Docker Machine so you can run Docker Engine commands from Windows terminals
- Docker Compose for running the `docker-compose` command
- Kitematic, the Docker GUI
- the Docker QuickStart shell preconfigured for a Docker command-line environment
- Oracle VM VirtualBox

Because the Docker Engine daemon uses Linux-specific kernel features, you can’t run Docker Engine natively on Windows. Instead, you must use the Docker Machine command, `docker-machine`, to create and attach to a small Linux VM on your machine. This VM hosts Docker Engine for you on your Windows system.

Step 1: Check your version

To run Docker, your machine must have a 64-bit operating system running Windows 7 or higher. Additionally, you must make sure that virtualization is enabled on your machine.

If you have Windows 10 Professional consider using Docker for Windows instead of the Toolbox.

Step 2: Install Docker Toolbox

In this section, you install the Docker Toolbox software and several “helper” applications. The installation adds the following software to your machine:

- Docker Client for Windows
- Docker Toolbox management tool and ISO
- Oracle VM VirtualBox
- Git MSYS-git UNIX tools

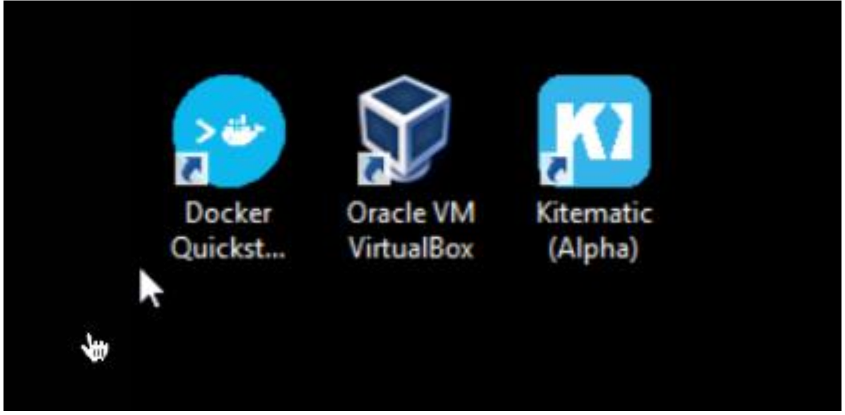
If you have a previous version of VirtualBox installed, do not reinstall it with the Docker Toolbox installer. When prompted, uncheck it.

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

Step 3: Verify your installation

The installer adds Docker Toolbox, VirtualBox, and Kitematic to your **Applications** folder. In this step, you start Docker Toolbox and run a simple Docker command.

1. On your Desktop, find the Docker QuickStart Terminal icon.



2. Click the Docker QuickStart icon to launch a pre-configured Docker Toolbox terminal.
   If the system displays a **User Account Control** prompt to allow VirtualBox to make changes to your computer. Choose **Yes**.
3. Make the terminal active by clicking your mouse next to the `$` prompt.
4.
5. Type the `docker run hello-world` command and press RETURN.
   The command does some work for you, if everything runs well, the command's output looks like this:

```
6.  $ docker run hello-world
7.  Unable to find image 'hello-world:latest' locally
8.  Pulling repository hello-world
9.  91c95931e552: Download complete
10.   a8219747be10: Download complete
11.   Status: Downloaded newer image for hello-world:latest
12.   Hello from Docker.
13.   This message shows that your installation appears to be working correctly.
14.
15.   To generate this message, Docker took the following steps:
16.    1. The Docker Engine CLI client contacted the Docker Engine daemon.
17.    2. The Docker Engine daemon pulled the "hello-world" image from the Docker Hub.
18.       (Assuming it was not already locally available.)
19.    3. The Docker Engine daemon created a new container from that image which runs
   the
```

```
      executable that produces the output you are currently reading.
   4. The Docker Engine daemon streamed that output to the Docker Engine CLI 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:
  https://docs.docker.com/userguide/
```

Optional: Add shared directories

By default, Toolbox only has access to the `C:\Users` directory and mounts it into the VMs at `/c/Users`.

Note: Within the VM path, `c` is lowercase and the `Users` is capitalized.

===

Once you have Docker installed you can install ODM for Docker as follows:

https://hub.docker.com/r/opendronemap/node-opendronemap/

Node-OpenDroneMap

node-OpenDroneMap is a Node.js App and REST API to access OpenDroneMap. It exposes an API which is used by WebODM.

Getting Started

We recommend that you setup node-OpenDroneMap using Docker.

- From the Docker Quickstart Terminal (Windows / OSX) or from the command line (Linux) type:
- `docker run -p 3000:3000 opendronemap/node-opendronemap`
- If you're on Windows/OSX, find the IP of your Docker machine by running this command from your Docker Quickstart Terminal:

```
docker-machine ip
```

Linux users can connect to 127.0.0.1.

- Open a Web Browser to `http://<yourDockerMachineIp>:3000`
- Load some images
- Press "Start Task"
- Go for a walk :)

If the computer running node-opendronemap is using an old or 32bit CPU, you need to compile OpenDroneMap from sources and setup node-opendronemap natively. You cannot use docker. Docker images work with CPUs with 64-bit extensions, MMX, SSE, SSE2, SSE3 and SSSE3 instruction set support or higher. Seeing a `Illegal instruction` error while processing images is an indication that your CPU is too old.

==

Change default vm settings

The default Virtual Box VM does not provide enough resources to give a good experience when building Yocto images. We recommend you create a new VM with at least 2 CPUs and 4GB of memory.

- Run **Docker Quickstart Terminal** and then run the following commands in that terminal.
- Remove the default vm

-
  ```
  docker-machine rm default
  ```
- Re-create the default vm
    - Choose the number of cpus with **--virtualbox-cpu-count**. For this example we'll use two.
    - Choose the amount of RAM: **--virtualbox-memory**. This is also based on the host hardware. However, choose at least 4GB.
    - Choose the amount of disk space: **--virtualbox-disk-size**. It is recommended that this be at least 50GB since building generates a lot* of output. In this example we'll choose 50GB.
    - Create vm with new settings

-
  ```
  docker-machine create -d virtualbox --virtualbox-cpu-count=2 --virtualbox-memory=4096 --virtualbox-disk-size=50000 default
  ```
- Restart docker

-
  ```
  docker-machine stop
  ```

-
  ```
  exit
  ```

Then start open a new Docker Quickstart Terminal.