Run your first container –

learn use Docker,to search for Docker images on Docker hub, and to run your images and connect to the applications inside them.

setup verification

Verify that you have the following software installed:

1. A 64-bit OS

2. [Docker Toolbox](https://www.docker.com/toolbox)

Docker Machine will create a virtual machine for us where we can run our docker containers, since you cannot run Docker containers on bare OS X or Windows. This part isn't needed if you're running a Linux desktop.

Let's set up our first container host!

```

$ docker-machine create --driver virtualbox default

Creating VirtualBox VM...

Creating SSH key...

Starting VirtualBox VM...

Starting VM...

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

```

You now have a container host ready to use, let's connect to it.

```

$ docker-machine env default

export DOCKER\_TLS\_VERIFY="1"

export DOCKER\_HOST="tcp://192.168.99.101:2376"

export DOCKER\_CERT\_PATH="/Users/user/.docker/machine/machines/default"

export DOCKER\_MACHINE\_NAME="default"

# Run this command to configure your shell:

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

```

Now copy & paste either the `export` lines, or run the `eval` command. Both will make sure your shell is correctly configured to connect to the correct container host.

## Using Docker

Now that you have a container host there are several things you can do. If you just run the command `docker` you'll get a list of all the commands you can issue.

First off we want to have a look at the Docker installation information, so run:

```

$ docker info

```

This should give you an output similar to this:

```

$ docker info

Containers: 0

Images: 0

Server Version: 1.9.0

Storage Driver: aufs

Root Dir: /var/lib/docker/aufs

Backing Filesystem: extfs

Dirs: 0

Dirperm1 Supported: true

Execution Driver: native-0.2

Logging Driver: json-file

Kernel Version: 3.19.0-30-generic

Operating System: Ubuntu 14.04.3 LTS

CPUs: 1

Total Memory: 991.6 MiB

Name: student012a

ID: 4PQN:XIMG:AMDW:R3UV:T7PX:ZSVJ:6XGC:CBXS:J3KF:4WZX:QXPK:HZBT

WARNING: No swap limit support

Labels:

provider=amazonec2

```

Here we can see how many images we have downloaded, how many containers we have running, and more info like version of Docker etc.

The second command we'll look at is searching for images that we find interesting.

The [Docker hub](https://hub.docker.com) currently has approximately \*\*75000\*\* images available for everyone to use freely.

Try a search for Sensu, a [really cool monitoring solution](http://sensuapp.org):

```

$ docker search sensu

```

to download an image using the `docker pull` command:

```

$ docker pull ubuntu

```

You'll see that Docker starts downloading a Docker image, this specific one is a base Ubuntu image. We'll try other images with more applications later.

When download is complete, your "Hello World" with Docker command looks something like this:

```

$ docker run ubuntu echo hello world

```

You can run other commands or invoke a bash shell like this:

```

$ docker run -i -t ubuntu /bin/bash

```

This will start the container, giving it its own file system, networking etc. and start up a shell for us to rummage around inside the container.

Notice how the prompt should resemble something similar to `root@87f4bb2a8aec:/#` now. This means you have an interactive session inside of the container.

To exit the container, there are two options. `CTRL-d` exits the shell and shuts down the container. If you want to keep the container running but detach from it, do `CTRL-p CTRL-q`. You can then re-attach to it by running `docker attach UNIQUEID`. Following this enter `CTRL-d` or `exit`.

We will run tomcat which a Java based application server. To do this we use the following command.

```

$ docker run -d -p 8888:8080 tomcat:8.0

```

The image will be downloaded if required and tomcat will be run in a \*deamon\* mode.

Now run the following command.

```

$ docker ps

```

Which will yield an output that looks something like below (Partial output).

```

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

9599164804bb tomcat:8.0 "catalina.sh run" 4 seconds ago Up 3 seconds 0.0.0.0:8888->8080/tcp mudit