Docker For Developers

Chris Tankersley



Docker for Developers

by Chris Tankersley

22



Table of Contents

Preface	VII	
Assumptions	VII	
Style Conventions	VIII	
Containers	1	
A Basic Container	2	
Beyond Basic Containers	3	
Along Comes Docker	3	
Why We Should Care as Developers	4	
Getting Started	7	
Installing Docker	8	
Running Our First Container	14	
How Containers Work	15	
	Preface Assumptions Style Conventions Containers A Basic Container Beyond Basic Containers Along Comes Docker Why We Should Care as Developers Getting Started Installing Docker Running Our First Container How Containers Work	

Chapter 3.	Working With Containers	17
	Images	17
	Controlling Containers	18
	Container Data	20
	Networking	23
Chapter 4.	Containerizing Your Application	27
	Getting PHP to Run	28
	Getting a Database	28
	Linking PHP to the MySQL Server	29
	Getting a Web Server	30
	Testing the application	30
	Thinking About Architecture	36
Chapter 5.	Creating Custom Containers	37
	Dockerfiles	38
	Building a Custom Image	42
Chapter 6.	Docker Tools	43
	Docker Machine	43
	Docker Swarm	45
	Docker Compose	48
Chapter 7.	Command Cheatsheets	51
	Images	51
	Containers	51
	docker-machine	52
	docker-compose	52

Getting Started

Chapter

Before we begin using Docker, we are going to need to install it. There will be a few caveats that we are going to discuss as we go through the installation because, unless you are on Linux, we're going to need some extra software to utilize Docker. This will create some extra issues down the road, but rest assured I'll keep you abreast of the more disasterous pitfalls that you may encounter, or various issues that might arise on non-Linux systems.

The installation is normally fairly easy no matter what OS you are going to use, so let's get cracking. We're going to install Docker 1.6. I'll go over some basic installation, but you can always refer to <u>https://docs.docker.com/installation/</u> for anything special or other Operating Systems if you aren't using Windows, OSX, or Ubuntu.

Throughout this book, I'm going to be using Ubuntu for all of the examples because it not only gives us a full operating system to work with as a host, but it's also very easy to set up. There are smaller Linux distributions that are designed for running Docker, but we are more worried about development at this stage. Since we're using containers it doesn't really matter what the host OS is.

If you are running Windows or OSX I would also highly recommend setting up an Ubuntu virtual machine instead of using the installation instructions later in this chaper for those operating systems. The reason for this is because on non-Linux systems we will need to utilize a virtual machine to provide a way for our Docker containers, which rely on Linux subsystems, to function. OSX and Windows have a tool called Docker Toolbox which will set all of this up for you however. It is more seamless than the older boot2docker system, which was a third-party

Getting Started

system that set up a VM and provided commands for working with the virtual machines. Docker Toolbox installation is detailed below.

Installing Docker

Ubuntu

Since Ubuntu ships with Long Term Release releases, I would recommend installing Ubuntu 14.04 and using that. The following instructions should work just fine for 14.10 or 15.04 as well. Ubuntu does have Docker in it's repositories but it is generally out of date pretty quickly so we're going to use an apt repository from Docker that will keep us up-to-date. Head on over to <u>http://www.ubuntu.com/download/server</u> and download the 14.04 LTS Server ISO and install it like normal. If you'd like a GUI, grab the desktop version. Either one will work. I'll be using the Desktop version with the intent to deploy to Ubuntu 14.04 Server.

If you've never installed Ubuntu before, Ubuntu provides a quick tutorial on what to do. Server instructions can be found at <u>http://www.ubuntu.com/download/server/install-ubuntu-server</u> and Desktop instructions can be found at <u>http://www.ubuntu.com/download/desktop/install-ubuntu-desktop</u>.

There's a few commands we can run to set up the Docker repository. Open up a terminal and install Docker:

```
$ sudo -i
$ wget -q0- https://get.docker.com/ | sh
$ usermod -a -G docker [username]
$ exit
$ sg docker
```

Line 1 switches us to the root user to make things easier. Lines 2 runs a script that adds the repository to Ubuntu, updates apt, and install Docker for us. Line 3 sets up your user to use Docker so that we do not have to be root all of the time, so replace [Username] with your actual user you will use.

We can make sure that the Docker engine is working by running docker -v to see what version we are at:

```
$ docker -v
Docker version 1.9.0, build 76d6bc9
```

To make sure that the container system is working, we can run a small contaier.

```
$ docker run --rm hello-world
```

Ubuntu is all set up!

Windows 10 with Hyper-V

In 2016 Docker formally released a beta of Docker that runs on Windows, if your version of Windows includes Hyper-V. This includes Windows 10 Professional, Enterprise, and Education. If you have Home, you will need to upgrade to Windows 10 Professional to be able to use Hyper-V.

INSTALLING DOCKER

Head on over to the Docker Products^[1] and download the package for Windows. The Docker Beta for includes Docker Engine, Compose, Machine, and Swarm all ready to be installed, and the installer will also enable Hyper-V.

Launch the Installer. Accept the install agreement and let Docker install. Once the installer is done, make sure the 'Launch Docker' option is selected and finish the installation. That's it! See Figure 2-1.

Docker will start itself up. If Hyper-V is not installed, it will prompt you to install Hyper-V and then restart the PC (Figure 2-2). This may take a few additional moments, and your PC may restart a few times.

Once that is all finished you will have a 'Docker for Windows' icon on your desktop, and a Docker icon



in your notification tray. As long as it is not red, you should be able to power up a Powershell window and run docker -v to make sure everything is working.

\$ docker -v
Docker version 1.12.0-rc2, build 906eacd, experimental

[1] Docker Products: <u>https://www.docker.com/products/docker#windows</u>

Getting Started

Out of Memory Issues

On my test laptop, which has 4GB of RAM, I had to lower the VM memory usage all the way down to 1024 MB before it would run. I am not 100% sure why exactly, as I had more than 2 GB of available RAM at startup. If you get this error, right-click on the Docker icon in your notification tray, select 'Settings', and then 'Advanced.' Lower the Memory slider until you are able to start Docker.

Windows 7/8/8.1

Docker and Microsoft have released a version of Docker that runs under Hyper-V, but only for users of Windows 10 Professional or higher. For users of older versions, or lower versions of Windows 10, you will still need to download the Docker Toolbox^[2], which will set everything up for us. The Toolbox includes the Docker client, Docker Machine, Docker Compose, Kitematic, and VirtualBox. It does not come with Docker Swarm.

Start up the installer. When you get to Figure 2-3 you can install VirtualBox and git if

	Setup - Docker Toolk	юх — чи
lect Components Which components should	be installed?	
Select the components yo install. Click Next when yo	u want to install; dear the com u are ready to continue.	ponents you do not want to
Full installation		Ŷ
Docker Client for Wine	lows	20.6 MB
Docker Machine for W	indows	154.3 MB
Docker Compose for V	Windows	5.4 MB
VirtualBox	PHILE A	81.6 MB
CB for Windows	(Alpha)	135.1 PIS
C) on the transmiss		6.01.0 F.10
Current selection requires	at least 428.5 MB of disk space	æ.

needed. I've already got them installed so I'll be skipping them but feel free to select those if needed. You should be good with the rest of the default options that the installer provides.

Since this changes your PATH, you will probably want to reboot Windows once it is all finished.

Once everything is all finished, there will be two new icons on your desktop or in your Start menu. Open up "Docker Quickstart Terminal." At this time Powershell and CMD support are somewhat lacking, so this terminal will be the best way to work with Docker. Once the terminal is up, run docker -v to see if you get a version number back.

FIGURE 2-4

\$ docker -v Docker version 1.9.0, build 76d6bc9

[2] Docker Toolbox: <u>https://www.docker.com/products/docker#windows</u>

Index

A

Alphine Linux, 4 Amazon EC2, 44 Ansible, 30 Apache, 4, 19, 36 application complicated, 22 containers, 49 demo, 28 foreground, 40 full-stack, 48 real, 32 test, 30

В

base boxes, 17 Basic chroot, 3 Boot2Docker running, 47 system, 7 BSD Jails, 3

C

CentOS images, 15, 17 chroot, 2–3, 14 client, 2, 4 command-line, 4 docker-swarm, 47 CMD, 38–41 default, 41 support, 10 commands

adduser, 14 eval, 44 execute, 15 init, 33 migrate, 33 new, 38 normal, 14 ps, 15 volume, 20 composer, 28, 36-37, 41 configuration, 24-25, 30, 33, 49 easy-to-deploy, 48 management, 4, 30 single file, 48 special, 36 containers 5.6-fpm, 28 busybox, 22 composer/composer, 41 custom, 31, 36-38, 40, 42 database, 29 deploy, 38 intermediate, 42 IP address, 41 linking, 25 named, 24 new, 15, 48 php-fpm, 41 single, 15, 27, 36 stopped, 19-20 testphp, 29 total, 49

Index

D

daemon, 39 mode, 19 data, sharing, 23 database, 23, 27-29, 32, 36, 48 config, 29 engines, 36 migrations, 32-33 servers, 22, 32, 36-37 data volumes, 20-23, 29, 37 existing, 49 normal, 20 persistant, 22 Debian, 39 default.conf, 24 dependencies, 28, 46 deployments, 32, 37, 45 Digital Ocean, 44 docker client, 4 command line interface(CLI), 11-12, 21 commands, 48 compose, 4, 10, 48-49 default, 19 installed, 43 ps, 19, 22, 24-25, 47, 51 running, 7-8, 13 virtual machine, 24 winpty, 11 docker4devs, 23 docker-compose, 49-50, 52 Dockerfile, 18, 38-42, 49 sample, 40 Docker Machine, 4, 10, 43-45 new, 52

Docker Quickstart Terminal, 10, 13 Docker Registry, 18 Docker Toolbox, 7, 21 installation, 8 Docker Tools, 43–44, 46, 48, 50 driver, 11, 44, 52 digitalocean, 44 virtualbox, 13

Ε

ENTRYPOINT, 38, 41 environments, 1, 13, 33, 48

F

files

docker-compose.yml, 48–49 etc/hosts, 49 local, 40 log, 42 phinx.yml, 33 remote, 40 single, 30 FreeBSD, 3

G

GUI, 8

Н

host computer, 20 file system, 21 remote, 44 HyperV, 44

IBM, 3 images available, 17-18 base, 15, 18, 38-39 composer, 28 custom, 31-32, 39, 42 download, 45 existing, 39 generated, 40 pre-built, 14 single, 17 stock, 27 swarm, 45 installation, 4, 7, 12 basic, 7, 14 normal Ubuntu, 14 IP address, 24-26, 30, 47 external, 47 new, 24

Κ

Kitematic, 10–11, 21 KVM, 3

L

layers, 18, 21–23, 40, 42 multiple, 38 Linux, 2–3, 7 containers, 3 distributions, 7 kernel, 3, 15 LXC, 3

Μ

machine multiple, 1, 45 new, 44 pre-configured, 43 provisioning, 43 running, 52 single, 45 stopped, 52 Microsoft, Azure, 44 migration scripts, 33 mounting directories, 13 multiple processes, running, 36 MySQL, 29, 33, 48-49 command line client, 29 container, 37 image, 22 layer, 42 PDO extensions, 32

Ν

name, 19 random, 19 specific, 49 network connections, 11 internal, 16 nginx, 19, 24–25, 30–31, 36–37, 39–40, 42, 47–49 configuration, 24, 30, 49 image, 17, 42 package, 39 nginx container, 24–26, 30, 32, 36, 49 generic, 42 new, 32 non-Linux systems, 7 Index

0

OpenVZ, 3 orchestrate, 4 orchestration system, 50 OSX, 2, 7, 12, 20, 28

Ρ

Parrallels, 3 Persisting Data, 21 Phinx, 32–33 PHP container, 23-25, 28-29, 31, 36 extensions, 31 image, 18, 31 multiple versions, 4-5 and nginx containers, 32 PDO extension, 32 running, 37 port exposed, 24-26, 41 external, 47 forwarding, 16 map, 26 Powershell, 11, 21 processes containerizing, 5 php-fpm, 24 separate, 2 production environment, 21, 30 Puppet, 4, 30

Q

Quickstart, 11 Terminal, 13, 21

R

registry, 51 private, 18 repository, 8, 18, 27, 42 apt, 8

S

server, 4, 8, 24, 30 single-process-per-container, 27 Solaris, 3 swarm, 45 master, 45–46 nodes, 50

Т

tar files, 40

U

ubuntu image, 15 installing, 8

V

Vagrant, 1, 3–4, 17, 27 Virtualbox, 1, 10–11, 44–47 virtual machines running, 2 single, 27 traditional, 16 Virtuozzo, 3 VMWare, 44 Workstation, 1 VOLUME instruction, 42 volumes cross-host, 50 hosted, 28 host-mounted, 20 link, 22 shared, 21

W

Windows, 7–8, 12, 20–21, 24, 26, 28, 47 Mounts, 21 Powershell, 21 users, 21, 44