

# 15 Quick Docker Tips

Lucas Carlson

Share  
+

**Brian Morearty recently gave a lightning talk (<https://speakerdeck.com/bmorearty/15-docker-tips-in-5-minutes>) at Twitter HQ about Docker. You should really check out his slides, they are fun and have a lot of character. He also runs a hands-on training course for Docker.**

But as a reference, here are the 15 tips he suggests:

## 1. Getting the id of the last-run container.

- You could do this:

```
$ ID=$(docker run ubuntu echo hello world)
hello world
$ docker commit $ID hello world
fd08a884dc79
```

- But you have to keep assigning IDs. Try this instead:

```
$ alias dl='docker ps -l -q'
$ docker run ubuntu echo hello world
hello world
$ dl
1904cf045887
$ docker commit 'dl' hello world
fd08a884dc79
```

## 2. Why you always want a Dockerfile, even if

you install everything in the shell.

- Docker commit commands can get unwieldy:

```
$ docker run -i -t ubuntu bash
root@db0c3978abf8:/# apt-get install postgresql -y
root@db0c3978abf8:/# exit
$ docker commit -run='{"Cmd":["postgres", "-too -many -opts"]}' `dl` postgres # ugly long
command, easier to do within a Dockerfile
5061b88e4cf0
```

- Instead, make a wee little Dockerfile that is FROM the image you made interactively
- There you can set CMD, ENTRYPOINT, VOLUME, etc.

### 3. Su-su-sudo

- Here is how to run Docker without root:

```
$ sudo groupadd docker
$ sudo gpasswd -a myusername docker
$ sudo service docker restart
$ exit
```

### 4. Take out the garbage

- Here is how to delete all stopped containers (useful if you need to re-run named containers):

```
$ docker rm $(docker ps -a -q)
```

- Tries, but conveniently fails to delete still-running containers.

### 5. jq – the gangsta way to parse docker inspect's output

- Here is how to grab the IP address in one line of unix:

```
$ docker inspect `dl` | grep IPAddress | cut -d '"' -f 4
172.17.0.52
$ docker inspect `dl` | jq -r '[0].NetworkSettings.IPAddress'
172.17.0.52
```

## 6. What environment variables does an image have?

- This is especially nice when using `docker run --link` to connect containers:

```
$ docker run ubuntu env
HOME=/
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
container=lxc
HOSTNAME=5e1560b7f757
RUN vs. CMD instruction
```

## 7. Here is the difference between RUN and CMD in the Dockerfile:

```
FROM an_existing_image
MAINTAINER my_email_address
\# docker build will execute these commands:
RUN apt-get update
RUN apt-get install softwares
\# docker run will run this command by default:
CMD ["softwares"]
```

## 8. CMD vs ENTRYPOINT instruction

- CMD arguments can be over-ridden:

```
cat Dockerfile
FROM ubuntu
CMD ["echo"]
$ docker run imagename echo hello
hello
```

- ENTRYPOINT arguments can NOT be over-ridden:

```
cat Dockerfile
FROM ubuntu
ENTRYPOINT ["echo"]
$ docker run imagename echo hello
echo hello
```

## 9. Does a Docker container have its own IP address?

- Yes. It is like a process with an internal IP address

```
$ ip -4 -o addr show eth0
2: eth0 inet 162.243.139.222/24
$ docker run ubuntu ip -4 -o addr show eth0
149: eth0 inet 172.17.0.43/16
```

## 10. Docker's architecture

- Docker has a thin CLI client that has binaries for Linux and OS X, and a REST server daemon with binaries for Linux over UNIX socket

```
$ nc -U /var/run/docker.sock
GET /images/json HTTP/1.1
HTTP/1.1 200 OK
Content-Type: application/json
Date: Tue, 05 Nov 2013 23:18:09 GMT
Transfer-Encoding: chunked
16aa
[{"Repository":"postgres","Tag":"..."
```

## 11. Graph the dependencies among your images

- Here is how to generate an image dependency diagram

```
$ docker images -viz | dot -Tpng -o docker.png
```

- Then to see it, run this command and go to

```
$ python -m SimpleHTTPServer
```

## 12. Where does Docker store everything?

- In /var/lib/docker... images are stored in the graph folder

```
$ sudo -i
\# cd /var/lib/docker
\# ls -F
containers/ graph/ repositories volumes/
```

## 13. Docker source code: Go, Go, Go, Grabote

- docker/docker.go – the CLI
- – the REST API router
- – the Docker server

## 14. RUN some daemons, exit the container. What happens?

- When you docker run -i -t it will override the RUN you have in your Dockerfile

```
$ cat Dockerfile
FROM ubuntu
RUN pg_ctl start
$ docker run -i -t postgresimage bash
root@08d363f57161:/# ps aux
\# No postgres daemon running
```

## 15. Letting containers talk to each other

```
$ docker run -d -name mysql ctlc/mysql
2202255dc8813401b8139bd9d6e7950a989507183a68ae08d9078d5700eddd2b
$ docker run --link mysql:db ubuntu env
HOME=/
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=715d0f260b66
DB_PORT=tcp://172.17.0.2:3306
DB_PORT_3306_TCP=tcp://172.17.0.2:3306
DB_PORT_3306_TCP_ADDR=172.17.0.2
DB_PORT_3306_TCP_PORT=3306
DB_PORT_3306_TCP_PROTO=tcp
DB_NAME=/insane_archimedes/db
```

Thanks to Brian Morearty who has a hands-on training course for Docker.

(**Edit:** Brian removed this online course (<http://ilikestuffblog.com/2014/12/12/last-chance-to-get-the-hands-on-with-docker-course/>) because it was too out-of-date, created in Jan 2014 and using Docker 0.7.6)

**Share**

**RELATED POSTS****Reliable Container Discovery for Cloud-Hosted SOA**

Reading Time: about 7 minutes. Service-oriented architecture (SOA) is a method of deploying and run...  
(/developers/blog/post/containers-service-discovery-soa-ansible)

**How to Deploy Drupal with Docker Containers**

Reading Time: about 5 minutes. Drupal is a powerful, popular content management system and web appli...  
(/developers/blog/post/drupal-php-docker-containers)

Provide your email to receive the  
Developer Center Newsletter.

**SUBMIT**

Start your free CenturyLink Cloud trial  
today

**GET STARTED ([HTTPS://WWW.CTL.IO/FREE-TRIAL/](https://www.ctl.io/free-trial/))**

© 2020 CenturyLink. All Rights Reserved.