Docker
Master the execution environment of your applications

Aurélien Dumez
Inria Bordeaux - Sud-Ouest

Tuesday, March 24th 2015
Content

1. The bad parts
2. Overview
3. Internals
4. Docker@Inria-BSO
5. Let’s play
6. Conclusion
The bad parts
The bad parts

- 09/09/14 [ Is Docker ready for production? Feedbacks of a 2 weeks hands on ]
- 02/07/15 [ Lets review.. Docker (again) ]
- 03/06/15 [ Docker, la plateforme à la mode (linuxfr) ]
Overview
- Automates the deployment of applications into containers
- Starts as an internal project within dotCloud (now Docker, Inc.)
- Is not a virtualization tool (not a new Vsomething)
- Does not turn a bad programmer (or sysadmin) into a good one

Remember the keyword: **container**
Open-source (Apache 2.0 license)

Written in Go (mainly because C/Java/Python/Other sucks)

Heavily relies on Linux kernel’s isolation features: cgroups, namespaces, capabilities...

First public release: March 2013

Good release rate:

- 2014: 1.1 (Jul), 1.2 (Aug), 1.3 (Oct), 1.4 (Dec)
- 2015: 1.5 (Feb), 1.6 (due: Mar)
- Portable deployment across machines: simpler dependencies
- Application-centric: one container per task (not a full VM)
- Automatic build: Dockerfile
- Versioning: git-like versions tracking
- Component re-use: share images between containers
- Sharing: registry / Docker Hub
- 64-bit GNU/Linux distribution
- Kernel: 3.10+ / older with some patches (CentOS 6.5+)
- Storage backend: Device Mapper, AUFS, VFS, BTRFS
  [See RedHat developer blog for more details]
- Preferably one or more security layer: SELinux, AppArmor...
  [J. Petazzoni: Containers & Docker: How Secure Are They?]
- Successfully tested on:
  - Debian GNU/Linux 8
  - CentOS 6.6 (+EPEL Testing repository)
  - ArchLinux (btrfs)
Internals
- Client-Server application
- RESTful API
- **Base Image**: image without parent
- **Image**: read-only layer on top of another image
- **Container**: read-write layer on top of another image
Docker@Inria-BSO
First contact : June 2014
Into production : November 2014
OS : CentOS 6.6 (with EPEL Testing repository)

Web sites powered by Docker :
  - Forum « Poppy Project »
  - Forum « Dessine-moi un Robot »
  - Forum « OpenLab Flowers »

Other uses : web apps testing (Joomla !, WordPress...)

Let’s play
Let's play

cmd : docker version

Who are you?

```plaintext
~$ docker version
Client version: 1.3.3
Client API version: 1.15
Go version (client): go1.3.3
Git commit (client): d344625
OS/Arch (client): linux/amd64
Server version: 1.3.3
Server API version: 1.15
Go version (server): go1.3.3
Git commit (server): d344625
```
Let’s play

- Registry: public or private place to store docker’s images
- Docker Hub: public registry + web interface + REST API
- User account needed to publish images
- Pricing: free for unlimited public repos + 1 private repo

1. ~$ docker login
2. Username: dumez
3. Password:
4. Email: aurelien.dumez@
5. Account created. Please use the confirmation link we sent to your email to activate it.

Security concerns? You may run your own registry! Really...
Search for and download an image

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>VIRTUAL SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>centos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ansible/centos7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tutum/centos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>blalor/centos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

~$ docker images

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>VIRTUAL SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>centos</td>
<td>7</td>
<td>88f9454e60dd</td>
<td>13 days ago</td>
<td>210 MB</td>
</tr>
<tr>
<td>centos</td>
<td>centos7</td>
<td>88f9454e60dd</td>
<td>13 days ago</td>
<td>210 MB</td>
</tr>
<tr>
<td>centos</td>
<td>latest</td>
<td>88f9454e60dd</td>
<td>13 days ago</td>
<td>210 MB</td>
</tr>
</tbody>
</table>
Create and list containers

```bash
~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

dce6ed9c055c  centos:7  "/bin/bash"  2 minutes ago Exited (130) 3 seconds ago kickass_poincare

~$ docker run --ti centos:7 /bin/bash
[root@dce6ed9c055c ~]# cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/
BUG_REPORT_URL="https://bugs.centos.org/"
[root@dce6ed9c055c ~]# ps aux
USER   PID %CPU %MEM    VSZ   RSS TTY STAT START TIME COMMAND
root    1  0.0  0.0 11752 2924 ?  Ss 13:04 0:00 /bin/bash
root   19  0.0  0.0 19756 2260 ?  R+ 13:05 0:00 ps aux
[root@dce6ed9c055c ~]# exit
exit

~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

dce6ed9c055c  centos:7  "/bin/bash"  2 minutes ago Exited (130) 3 seconds ago kickass_poincare
```
Let's play

cmd: docker start

```bash
~$ docker start --ai dce6
[root@dce6d9c055c ~]# ping -c 2 linuxfr.org
PING linuxfr.org (88.191.250.176) 56(84) bytes of data.
64 bytes from prod. linuxfr.org (88.191.250.176): icmp_seq=1 ttl=49 time=16.5 ms
64 bytes from prod. linuxfr.org (88.191.250.176): icmp_seq=2 ttl=49 time=16.5 ms
--- linuxfr.org ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 16.567/16.572/16.578/0.128 ms
[root@dce6d9c055c ~]# exit
~$
```

...and yes, networking just works out of the box!
What happened in my container?

```
1 ~$ docker logs dce6
2 [root@dce6ed9c055c /]# cat /etc/os-release
3 NAME="CentOS Linux"
4 VERSION="7 (Core)"
5 ID="centos"
6 ID_LIKE="rhel fedora"
7 VERSION_ID="7"
8 PRETTY_NAME="CentOS Linux 7 (Core)"
9 ANSI_COLOR="0;31"
10 CPE_NAME="cpe:/o:centos:centos:7"
11 HOME_URL="https://www.centos.org/
12 BUG_REPORT_URL="https://bugs.centos.org/"
13
14 [root@dce6ed9c055c /]# ps aux
15 USER   PID %CPU %MEM  VSZ  RSS TTY STAT START  TIME COMMAND
16 root    1  0.0  0.0 11752 2924  ?  Ss  13:04  0:00 /bin/bash
17 root    19  0.0  0.0 19756 2260  ?  R+  13:05  0:00 ps aux
18 [root@dce6ed9c055c /]# exit
19 exit
20 [root@dce6ed9c055c /]# ping -c 2 linuxfr.org
21 PING linuxfr.org (88.191.250.176) 56(84) bytes of data.
22 64 bytes from prod. linuxfr.org (88.191.250.176): icmp_seq=1 ttl=49 time=16.5 ms
23 64 bytes from prod. linuxfr.org (88.191.250.176): icmp_seq=2 ttl=49 time=16.5 ms
24
25 ---- linuxfr.org ping statistics ----
26 2 packets transmitted, 2 received, 0% packet loss, time 1001ms
27 rtt min/avg/max/mdev = 16.567/16.572/16.578/0.128 ms
28 [root@dce6ed9c055c /]# exit
```
Let's play

cmd : docker run

Launch multiple containers in daemon mode

```
~$ docker run -d centos:7 /bin/cat /etc/hostname
b9d0952884d1dc88e3a4a666088e8c34f3e8c3e73dc0eb4695e0bf6e5

~$ docker run -d centos:7 /bin/cat /etc/hostname
7393f6df79b73b9a2a7800f7bd49b711cf117b687eb44985d56ff81ed45f05

~$ docker run -d centos:7 /bin/cat /etc/hostname
7c9011c9867a52810322c364831baa706572b15e5f2ff04af1eb1d17b876b07f

~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7c9011c9867a centos:7 "/bin/cat /etc/hostname" 2 seconds ago Exited (0) 2 seconds ago jolly_goodall
7393f6df79b7 centos:7 "/bin/cat /etc/hostname" 3 seconds ago Exited (0) 3 seconds ago insane_carson
b9d0952884d1 centos:7 "/bin/cat /etc/hostname" 5 seconds ago Exited (0) 4 seconds ago happy_feynman
dce6ed9c055c centos:7 "/bin/bash" 58 minutes ago Exited (0) 10 minutes ago kickass_poincare
```
Some housekeeping

<table>
<thead>
<tr>
<th>CONTAINER ID</th>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
<th>STATUS</th>
<th>PORTS</th>
<th>NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>7c9011c9867a</td>
<td>centos:7</td>
<td>&quot;/bin/cat /etc/hosts&quot;</td>
<td>2 seconds ago</td>
<td>Exited (0)</td>
<td>2 seconds ago</td>
<td>jolly_goodall</td>
</tr>
<tr>
<td>7393f6df79b7</td>
<td>centos:7</td>
<td>&quot;/bin/cat /etc/hosts&quot;</td>
<td>3 seconds ago</td>
<td>Exited (0)</td>
<td>3 seconds ago</td>
<td>insane_carson</td>
</tr>
<tr>
<td>b9d0952884d1</td>
<td>centos:7</td>
<td>&quot;/bin/cat /etc/hosts&quot;</td>
<td>5 seconds ago</td>
<td>Exited (0)</td>
<td>4 seconds ago</td>
<td>happy_feynman</td>
</tr>
<tr>
<td>dce6ed9c055c</td>
<td>centos:7</td>
<td>&quot;/bin/bash&quot;</td>
<td>58 minutes ago</td>
<td>Exited (0)</td>
<td>10 minutes ago</td>
<td>kickass_poincare</td>
</tr>
</tbody>
</table>

$ docker rm 7c 73 b9

$ docker ps -a

<table>
<thead>
<tr>
<th>CONTAINER ID</th>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
<th>STATUS</th>
<th>PORTS</th>
<th>NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>dce6ed9c055c</td>
<td>centos:7</td>
<td>&quot;/bin/bash&quot;</td>
<td>58 minutes ago</td>
<td>Exited (0)</td>
<td>10 minutes ago</td>
<td>kickass_poincare</td>
</tr>
</tbody>
</table>
Did you say: "Lego® bricks"?

1. `docker run --name some_mysql -e MYSQL_ROOT_PASSWORD=mysecretpassword -d --point -P mysql`
2. `docker run --name some_wordpress --link some_mysql:mysql -p 127.0.0.1:8080:80 -d wordpress`
No magic inside

```bash
~$ docker exec -ti 58 /bin/bash
root@5887fa436734:/var/www/html# env
[
HOSTNAME=5887fa436734
MYSQL_ENV_MYSQL_ROOT_PASSWORD=mysecretpassword
MYSQL_PORT_3306_TCP_PORT=3306
MYSQL_PORT_3306_TCP=tcp://172.17.0.2:3306
MYSQL_ENV_MYSQL_VERSION=5.6.23
MYSQL_NAME=/some_wordpress/mysql
MYSQL_PORT_3306_TCP_PROTO=tcp
MYSQL_PORT_3306_TCP_ADDR=172.17.0.2
MYSQL_ENV_MYSQL_MAJOR=5.6
MYSQL_PORT=tcp://172.17.0.2:3306
root@5887fa436734:/var/www/html# cat /etc/hosts
[
172.17.0.3 5887fa436734
172.17.0.2 mysql
```
Dockerfile

```bash
FROM centos:6
MAINTAINER Aurelien Dumez <aurelien.dumez@inria.fr>
RUN yum -y update; yum clean all
RUN yum -y install httpd; yum clean all
RUN mkdir -p /var/www/html
ADD index.html /var/www/html/
ADD start.sh /
EXPOSE 80

#CMD ["/usr/sbin/httpd", "-DFOREGROUND"]
CMD ["/bin/bash", "/start.sh"]
```

Build

```bash
~$ docker build --rm -t dumez/centos_httpd .
Sending build context to Docker daemon 4.608 kB
Sending build context to Docker daemon
Step 0 : FROM centos:6
    ---> f6808a3e4d9e
   [.................................]
Step 8 : CMD /bin/bash /start.sh
    ---> Running in fd917c365721
    ---> 80557bbfc1d0
Removing intermediate container fd917c365721
Successfully built 80557bbfc1d0
~$
```
Let's play

```bash
～$ docker history dumez/centos_httpd

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>CREATED</th>
<th>CREATED BY</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>80557bbfc1d0</td>
<td>23 hours ago</td>
<td>/bin/sh –c #(nop) CMD [/bin/bash /start.sh]</td>
<td>0 B</td>
</tr>
<tr>
<td>d79a897fa0a6</td>
<td>23 hours ago</td>
<td>/bin/sh –c #(nop) EXPOSE map[80/tcp:{]}</td>
<td>0 B</td>
</tr>
<tr>
<td>e271b2439c32</td>
<td>23 hours ago</td>
<td>/bin/sh –c #(nop) ADD file:f70778d9d3db9a13c0</td>
<td>63 B</td>
</tr>
<tr>
<td>55049b1cdb19</td>
<td>23 hours ago</td>
<td>/bin/sh –c #(nop) ADD file:302d28059438a7495f</td>
<td>156 B</td>
</tr>
<tr>
<td>a28e0f8428fe</td>
<td>23 hours ago</td>
<td>/bin/sh –c mkdir –p /var/www/html</td>
<td>0 B</td>
</tr>
<tr>
<td>437ee40ad1d5</td>
<td>23 hours ago</td>
<td>/bin/sh –c yum –y install httpd; yum clean all</td>
<td>29.95 MB</td>
</tr>
<tr>
<td>fef0a9e288f0</td>
<td>23 hours ago</td>
<td>/bin/sh –c yum –y update; yum clean all</td>
<td>13.42 MB</td>
</tr>
<tr>
<td>3764a81cd6e5</td>
<td>23 hours ago</td>
<td>/bin/sh –c #(nop) MAINTAINER Aurelien Dumez &lt;</td>
<td>0 B</td>
</tr>
<tr>
<td>f808a3e4d9e</td>
<td>2 weeks ago</td>
<td>/bin/sh –c #(nop) ADD file:2b2b26209d285cd1a9</td>
<td>202.6 MB</td>
</tr>
<tr>
<td>5b12ef8fd570</td>
<td>5 months ago</td>
<td>/bin/sh –c #(nop) MAINTAINER The CentOS Project</td>
<td>0 B</td>
</tr>
<tr>
<td>511136ea3c5a</td>
<td>21 months ago</td>
<td></td>
<td>0 B</td>
</tr>
</tbody>
</table>
```

Step by step
Let's play

Connect to a running container

```bash
~$ docker run -d -P dumez/centos_httpd
b71600b5b407286248e0cc0d0af2685b9770fc3a822885beab901e368dbe4068

~$ docker exec -ti b71 /bin/bash
[root@b71600b5b407 /]# ps aux
```

<table>
<thead>
<tr>
<th>USER</th>
<th>PID</th>
<th>%CPU</th>
<th>%MEM</th>
<th>VSZ</th>
<th>RSS</th>
<th>TTY</th>
<th>STAT</th>
<th>START</th>
<th>TIME COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>11360</td>
<td>2344</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /bin/bash/start.sh</td>
</tr>
<tr>
<td>root</td>
<td>9</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>6344</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>root</td>
<td>10</td>
<td>0.0</td>
<td>0.0</td>
<td>4164</td>
<td>456</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 tail -f /var/log/httpd/access_log</td>
</tr>
<tr>
<td>apache</td>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>12</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>13</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>14</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>15</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>16</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>17</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>apache</td>
<td>18</td>
<td>0.0</td>
<td>0.0</td>
<td>175336</td>
<td>3860</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /sbin/httpd</td>
</tr>
<tr>
<td>root</td>
<td>19</td>
<td>2.0</td>
<td>0.0</td>
<td>14784</td>
<td>3004</td>
<td>?</td>
<td>S</td>
<td>15:26</td>
<td>0:00 /bin/bash</td>
</tr>
<tr>
<td>root</td>
<td>32</td>
<td>0.0</td>
<td>0.0</td>
<td>16668</td>
<td>2140</td>
<td>?</td>
<td>R+</td>
<td>15:26</td>
<td>0:00 ps aux</td>
</tr>
</tbody>
</table>

Aurélien Dumez

Docker
Let's play cmd : docker logs

$ docker logs -f b7

172.17.42.1 — [19/Mar/2015:15:31:19 +0000] "GET / HTTP/1.1" 200 156 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.76 Safari/537.36"

172.17.42.1 — [19/Mar/2015:15:31:19 +0000] "GET /favicon.ico HTTP/1.1" 404 287 "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.76 Safari/537.36"

Hello Docker!
Conclusion
- Form your own opinion: use Docker
- Don’t stay alone: use the Hub
- Docker can really save your time (Discourse@Inria-BSO)
- Docker evolves quickly: stay tuned
Thanks ! Questions ?