



Hewlett Packard
Enterprise

Building Linux distribution packages with Docker

Bruno Cornec, HPE EG EMEA Open Source and Linux Strategist

February 2016

Introducing myself

- Software engineering and Unices since 1988:
 - Mostly Configuration Management Systems (CMS), Build systems, quality tools, on multiple commercial Unix systems
 - Discovered Open Source & Linux (OSL) & made first contributions in 1993
 - Full time on OSL since 1995, first as HP reseller then @HP
- Currently:
 - OSL Technology **Strategist**, EMEA EG Innovation Solution Center aka HP/Intel Solution Center, Grenoble
 - HP OSL **Advocate** and Converged Infrastructure **Ambassador**
 - WW Linux Community Lead for the HP **Open Source Profession**
 - POSS conference, OpenStack.fr and AFUL **board member**. Conferences at WW level at LinuxCon, Linux.conf.au, ...
 - MondoRescue, Project-Builder.org, UUWL and PUSK **Project Lead**
 - LinuxCOE, mrepo, tellico, rinse, fossology, collectl, Ironic **contributor**
 - FOSSBazaar/SPDX and OSL **Governance** enthusiast
 - Mandriva, Mageia, Fedora **packager**
- And also:
 - Amateur singer (Alto / Tenor), recorder player since 1976 and Choir **director** since 1987, CD collector (5000+), Concerts, Photography



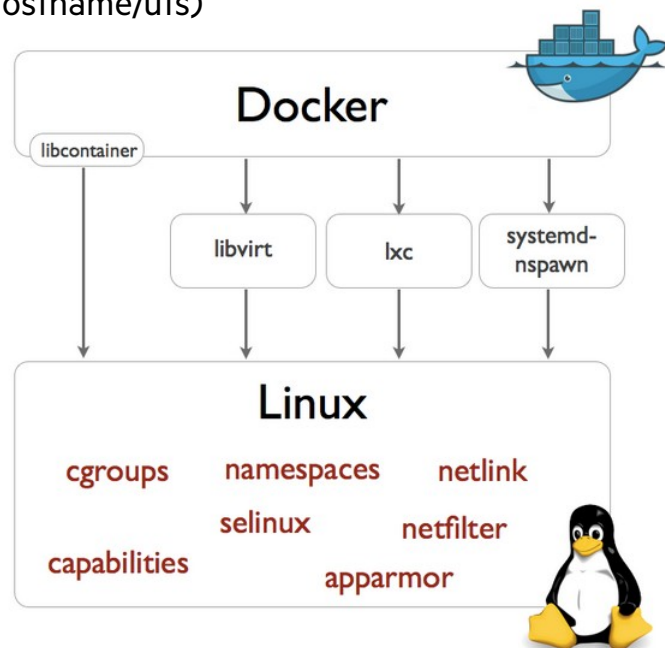
Context

Linux Containers

OS Virtualization (“LXC / Docker” containers)

OS Virt

- Applications in a zone are isolated from the others (chroot on steroids)
 - Security (as secure as your kernel)
 - Namespaces (root, UID, GID, network, PID, IPC, mount points, hostname/uts)
 - Cgroup Granularity
 - Single OS image manages HW accesses
 - Performance & transparency
neither emulation nor full virtualization
 - Netfilter for network isolation (NAT and port fwd)
 - AuFS overlays isolated file systems on top of a physical file system
- Other technos available
 - V-Server
 - Virtuozzo
 - OpenVZ

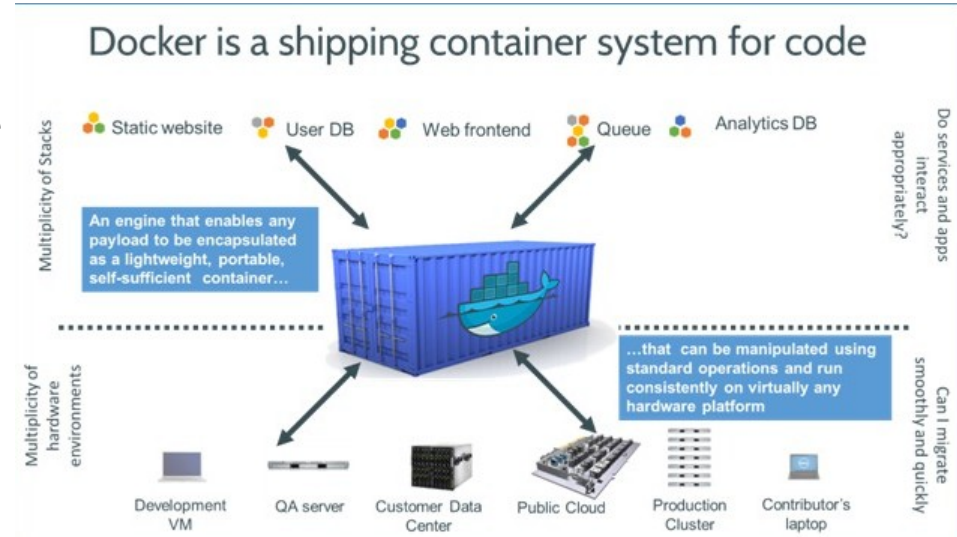




Docker In Brief

- **Vision:** Build, Ship, and Run Any App Anywhere
- **Both:** a company and an open source ecosystem (under the Apache license V2.0 since 2013)
- **Docker is a tool chain** written in Go that simplifies the creation and management of containers.
- **Paradigm shift:** “new” packaging model for deploying applications and services using Linux containers (Google has been doing this for years; 2 billion / week)
- **Containers:** A self-contained portable environment for packaging, deploying, and executing applications and services. Contains all execution dependencies for a service (single process -> single container)
- **Better resource utilization:** VMs virtualize the CPU, containers virtualize process(es) (lightweight)
- **Simplified Application Lifecycle Management:**
 - Build and configure once (integrated DevOps)
 - Deploy anywhere (laptop, server, cloud, ...)

Hewlett Packard Enterprise



• Value prop

- Reliable deployment
- More efficient deployment
- Reduced deployment latency
- More efficient use of system resources

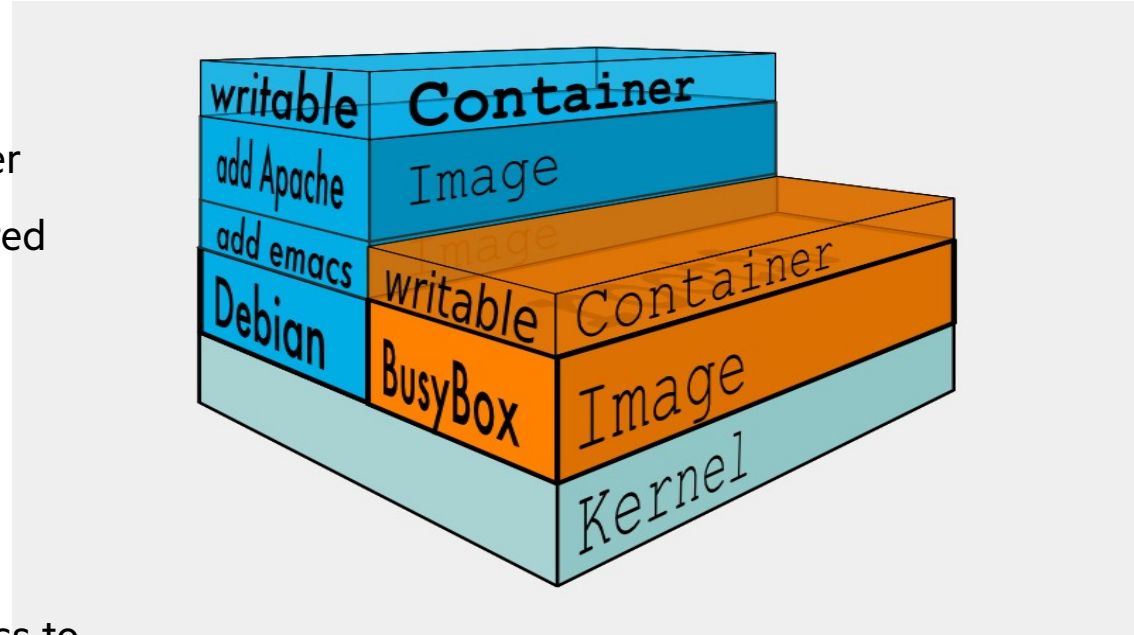
A layered approach

Images

- Local or remote reference content to initiate a container
- Multiple images can be layered adding content at each time using Copy on Write FS
- Cache to speed up repeated operations

Containers

- Last layer providing rw access to the cumulated set of images



1 application + its deps == 1 container

Why building distribution packages with Containers vs VMs?

- **Container like VMs brings isolation**

- No pollution of your running environment
- Easily scratch and redo if problems
- Easier refinement & automation of the build environment with the Docker file

- **Containers like VMs brings multi-distribution support**

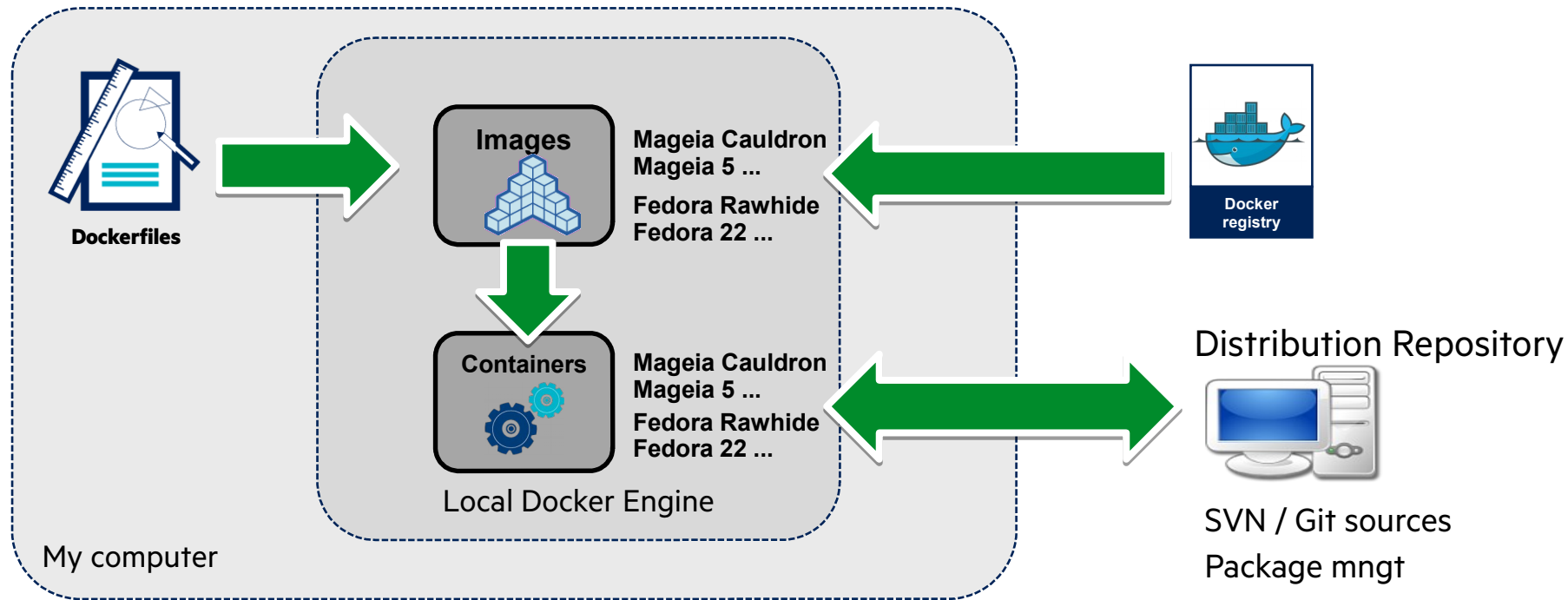
- Easy to build for another distribution than yours
- Useful also for your own distro: not everybody uses an unstable distro

- **Containers can use natively your home directory**

- Allow sharing of your package sources for local and in container build
- Allow sharing of your .rpmmacros, .rpmrc files, Mageia SSH keys or Fedora certs

 **VMs are mandatory is you need a different kernel**

Basic Docker workflow to build distribution packages



Building distribution images with a Dockerfile

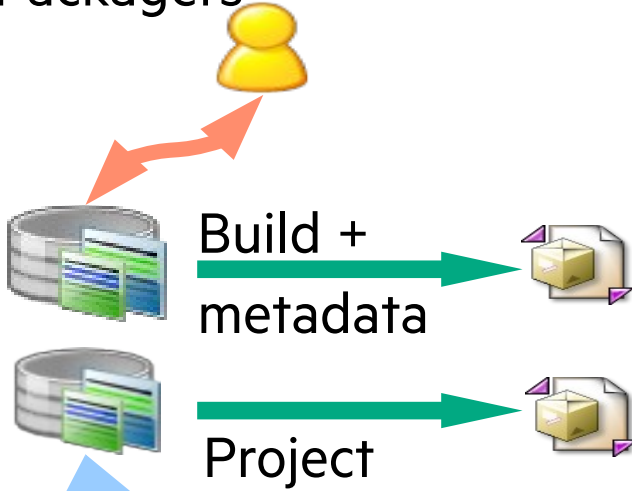
```
#!/bin/bash
mkdir -p /tmp/docker-mageia
cd /tmp/docker-mageia
RD="/home"
WD="$RD/bruno/Maison/bruno/prj/mageia/"
cat > Dockerfile << EOF
FROM juanluisbaptiste/mageia-cauldron:latest
MAINTAINER bcornec@mageia.org
RUN urpmi.update -a -c -f
RUN urpmi --auto --auto-select --no-recommends
RUN urpmi --auto bm subversion mgarepo colordiff sudo
RUN sed -i 's/users:x:.*$/users:x:501/' /etc/group
RUN useradd bruno -u 600 -g 501 -N -M -d $RD/bruno
RUN echo "bruno ALL=NOPASSWD:/sbin/urpmi" >> /etc/sudoers
WORKDIR $WD
USER bruno
CMD /bin/bash
EOF

docker build --file=Dockerfile -t pb:mageiabuild .
if [ $? -eq 0 ]; then
    docker run -v $RD/bruno:$RD/bruno -ti pb:mageiabuild
fi
```

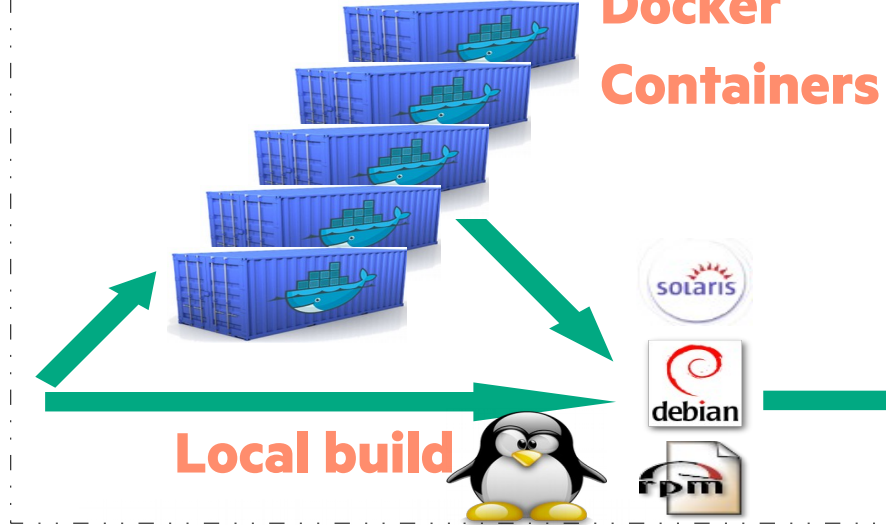
Show Time !

Continuous Packaging with Docker & project-builder.org

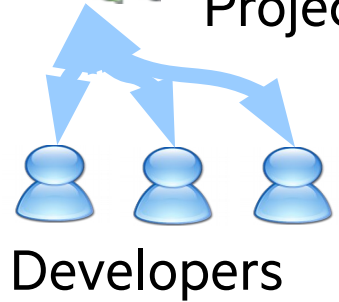
Packagers



Local Build Server



Project Repository



Bruno.Cornec@hpe.com

(Open Source and Linux Technology Strategist
at the HP/Intel Solution Center)

<http://downloads.linux.hpe.com/>



THANK YOU

Linus Torvalds, Richard Stallman, Eric Raymond,
Nat Makarevitch, René Cougnenc, Eric Dumas,
Rémy Card, Bdale Garbee, Bryan Gartner, Craig
Lamparter, Lee Mayes, Gallig Renaud, Andree
Leidenfrost, Phil Robb, Bob Gobeille, Martin
Michlmayr among others, for their work and
devotion to the Open Source Software cause...
and my family for their patience :-)

"Changes are never easy to make.
There is comfort and safety in tradition, but
change must come, no matter how painful
or expensive it may be."

Bill Hewlett