Hewlett Packard
Enterprise

Continuous Packaging is also mandatory for DevOps

Bruno Cornec
HP EMEA EG Presales Strategist
WW Linux Community Lead, HP Open Source Profession

v 3.2 – August 2016

EMEA Customers Solution Innovation Center

Making the new style of IT a reality

- 15+ years of success, world wide programs, including **Cloud** Center of Excellence, **Big Data** Center of Excellence, **Open Source** Solutions Initiative, RISC to **HP Intel Architecture** Migrations, **NVF** Center of Excellence, EMEA **Networking** Customer Visit Center and more

- Complete IT (400+ systems, 3000+ network ports, 500+ TB storage)
- Portfolio of 40+ ready to demo solutions with access to our ecosystem of Partners
- Complete test & validation environment
- Strategic partnership with **Intel**, 15-year long standing collaboration
- Strategic partnership with **Red Hat** 7-year collaboration (OSSI)

- A unique proof point in the industry with a proven service offering

CoE

PoC

Live demo

Workshop

Mission: Accelerate the adoption of new and innovative solutions by creating simple and rewarding end-to-end customer experiences that benefit our customers and partners, in a compelling and engaging collaborative environment.

...more information available at http://www.hpintelco.net

Hewlett Packard Enterprise

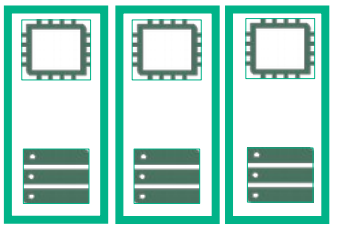
Introducing myself

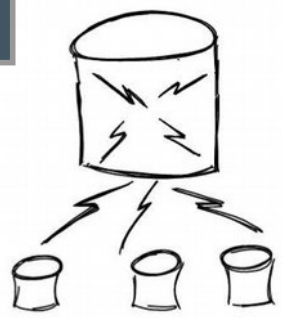
- Software engineering and Unices since 1988:
  - Mostly Configuration Management Systems (CMS), Build systems, quality tools, on multiple commercial Unix systems
  - Discovered Open Source & Linux (OSL) & made first contributions in 1993
  - Full time on OSL since 1995, first as HP reseller then @HP
- 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 (6000+), Concerts, Photography

DevOps approach:
A continuous delivery pipeline

Change in
- Executable code
- Configuration
- Infra / environment
- Data
- Monitoring
- ...

Version controlled
Peer reviewed
Automated tests (lots)
Gates (automated / manual)
Automated deployment
Continuous Packaging
Feedback loop (monitoring)
Codified Operation Readiness





Continuous Delivery Pipeline

git

ANSIBLE

Zuul Nodepool

Nexus

Project-Builder.org

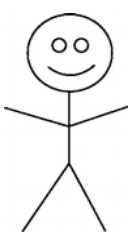
Puppet

Chef

openstack™

HPE Helion

docker



Infrastructure as code

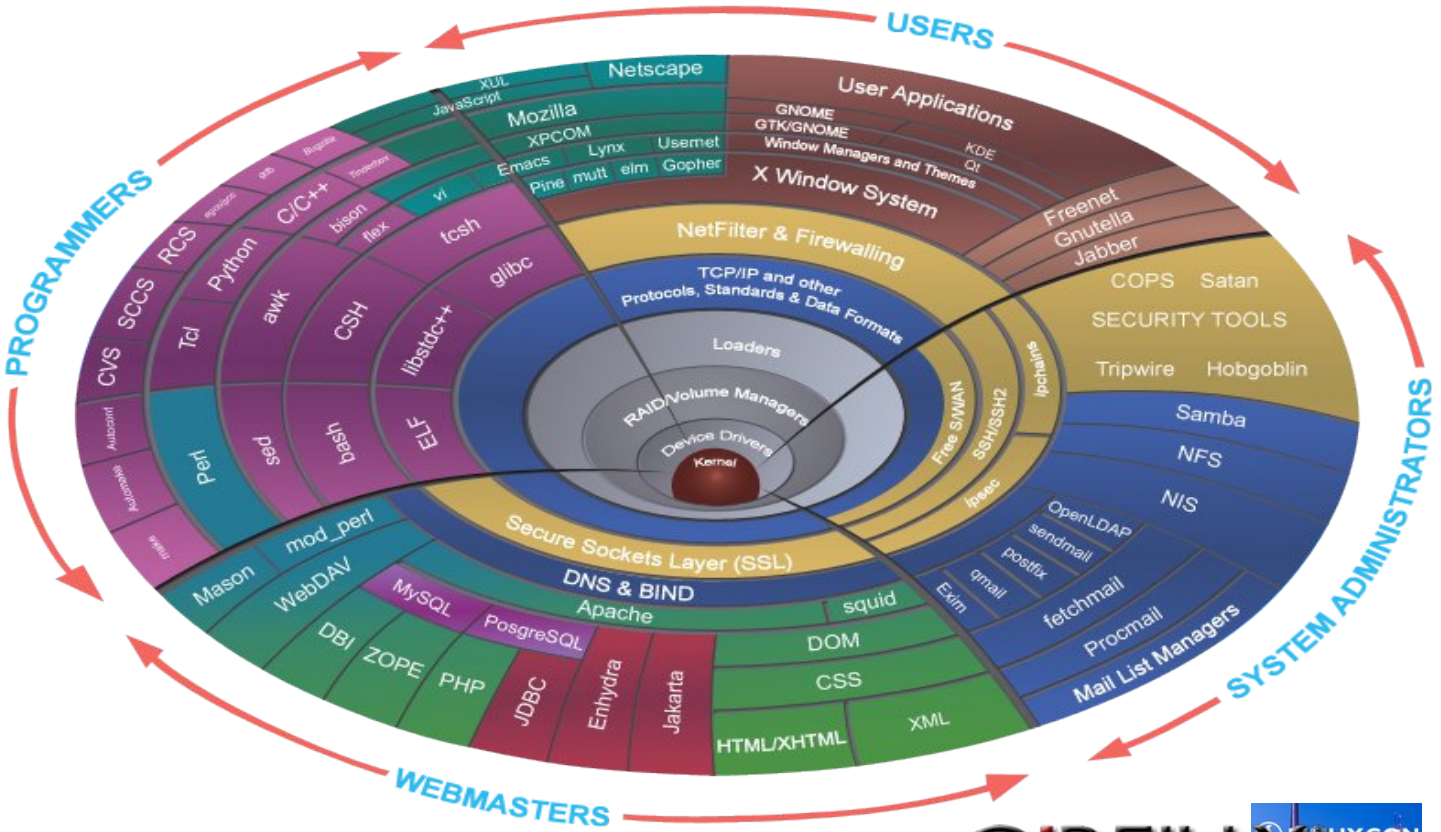
Hewlett Packard Enterprise

LINUXCON NORTH AMERICA

Basics on packaging

Linux Distribution : a project in itself

- **Coherent packages set** (1k-30k) taken from upstream projects
- **Package Manager**
- Management tools
- **Installation program**
- Startup scripts
- Specific tools
- **Functional updates**
- **Security updates**
- Community driven or Commercial (HW certification, LTS, support)



Linux Distributions Time line

- 1984 – GNU/FSF Project - R. Stallman
- 1991 - Linux
  L. Torvalds
- 1992 – SLS – Peter Mc Donald
- 1993 – Slackware – Patrick Volkerding
- 1993 – Debian Package
- 1993 – Red Hat Linux
  Marc Ewing
- 1996 – SuSE & Yast – Florian La Roche
- 1996 – Debian GNU/Linux
  Ian Murdock
- 1997 – Red Hat Package Manager
  Erik Troan & Mark Ewing
- 1998 – Advanced Packaging Tool
  Brian White
- 2001 – SLES
- 2002 – RHEL
- 2003 – Fedora
  Warren Togami
- 2003 – YUM - Seth Vidal
- 2004 – Ubuntu
  Mark Shtuttleworth
- 2005 – OpenSUSE
- 2010 – Mageia
  Anne Nicolas
- 2015 – DNF








Some definitions

- **Software Package:**
  - Application stored with its metadata and build receipt in an archive format.
  - Provides dependency information at build and install time

- **Package format:**
  - Linux provides multiple format: rpm, deb, tgz, ipkg,...
  - Open, based on tool like cpio.
  - Associated with distribution families.

- **Package manager:**
  - Automates packages installation, upgrade, configuration, and removal in a consistent manner.
  - Manages package dependencies to install easily from top of tree

- **Package repository:**
  - Storage location from which software packages are retrieved for installation/update
  - Manages repository metadata, including dependencies

- **Continuous Packaging:**
  - Every software component is managed using software packages
  - Package build is done on the fly, as the software is developed

Why using packages ?

- **tar.gz** format advantages:
  - Relatively easy to handle
  - Stable
  - Only 2 tools needed (tar and gzip)
  - Can integrate some files for metadata

- **tar.gz** format drawbacks :
  - No repository management
  - No dependency management
  - No easy update mechanism
  - No signature support
  - Limited checksum support
  - No package database

Why using packages ?

- **RPM/deb** format advantages
  - Stable
  - Binary and source formats available w/ multiarch support
  - Native support for LSB/FHS
  - Provides metadata, build procedure, patches and upstream content
  - Manages installation, upgrade, removal
  - Signature/Checksum support and verification
  - Deployment server availability – Scripted methods
  - Baseline support
  - RPM places everything in the .spec file and supports subpackages
  - RPM Package database available to query metadata

- **RPM/deb** format drawbacks
  - Require appropriate tools but in distro
  - Portability across OSes

Building packages and repositories (rpm world)

- **rpmbuild**
  - Build src.rpm and arch.rpm packages from the SPEC file
  - Take in account config files, cron jobs, init scripts, log rotation, shell config
  - *rpmbuild -ba pkg.spec*
  - **DO NOT BUILD AS root**
- **Signing RPMs**
  - Ensures authenticity of the provider and package integrity
  - Requires GPG configuration and macros in $HOME/.rpmmacros
  - *rpm --addsign pkg.src.rpm pkg.arch.rpm*
- **rpmlint**
  - Check rpm package common errors wrt distribution policies
  - *rpmlint pkg.src.rpm pkg.arch.rpm*
- **createrepo**
  - Separate command (not a yum option)
  - Create a yum repository from a directory with packages
  - *createrepo* .

Packaging best practices

- **Have a working installation procedure**
  - Use configure if possible or language build tools (setup.py, Makefile.PL, …)
  - Have a file based install working targeting /usr/local
  - Use variables for all target directories (/etc, /usr/share, /usr/bin, /usr/lib, …)
  - Script the build and install phases if necessary to share between various tools
- **Know your package deliverables**
  - Know what you want to deliver and where, what is optional. Package accordingly
  - Know your external dependencies
  - No source components should have their own separated packages
  - Avoid including other projects code inline. Use dependency or re-package separately if fork needed.
  - Generate repositories to ease your consumers' life
  - Decide on a license
- **A delivery is a tree of packages**
  - Increasing number of packages reduces build time on the long run
  - Increase complexity at package definition, not at install time, thanks to package managers
  - Decide whether you want meta-packages (prj-all, prj-net, prj-tape, prj-data, …)

Basics on project-builder.org

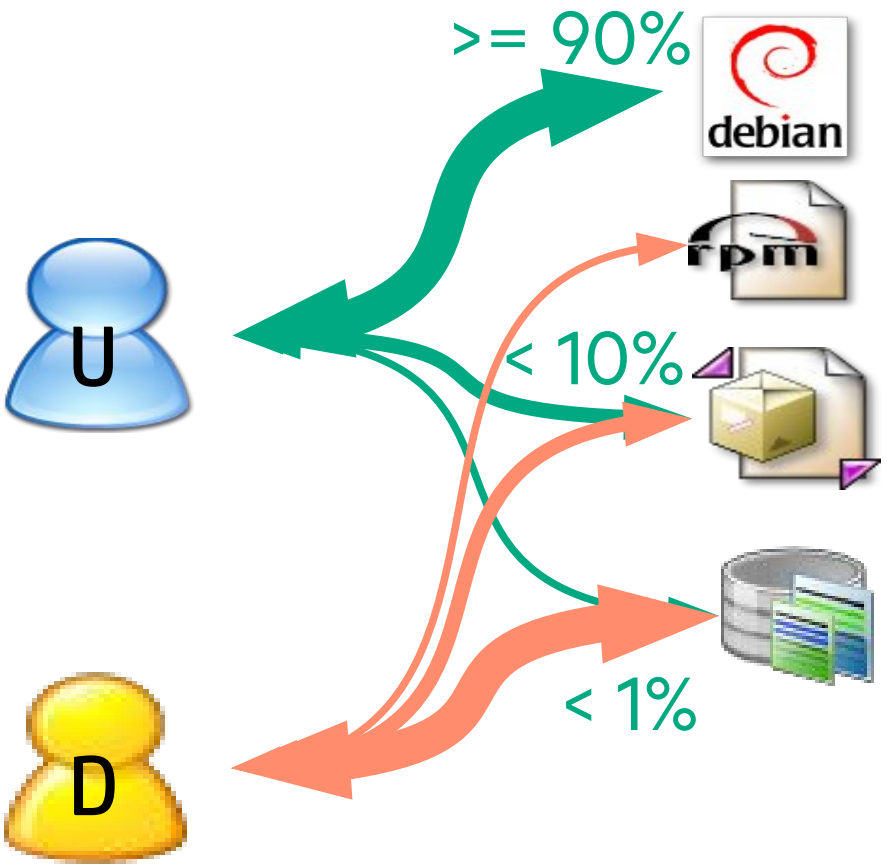
What is project-builder.org's goal ?

Make upstream projects life easier with regards to packaging their software

What is project-builder.org's itch ?

Make my life easier with regards to packaging my software

Users / Sysadmins want packaged software

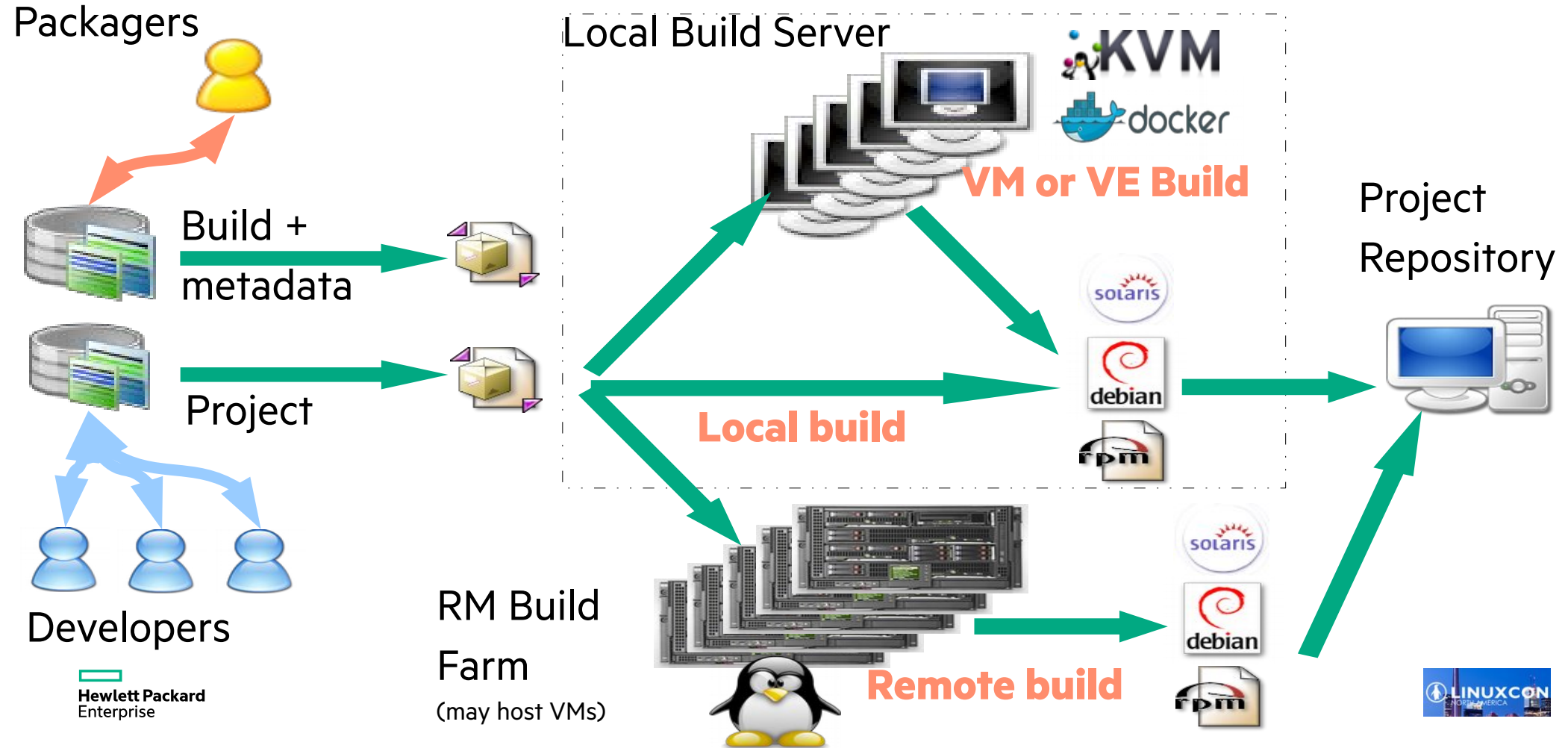


- Ease of use: GUI, CLI
- Distribution compliance
- Integration with deployment tools
- Lag between SW and Pkg availability
- Test of alpha/beta/VCS SW
- Too many:
  - distributions,
  - versions,
  - package formats,
  - tools to generate packages
  - tools to manage repositories

Upstream benefits from Continuous packaging

- Packaging should be a **project concern** as well as coding, testing, installing, …. especially for smaller projects
- Packaging as your **only** way of **delivery** (not a dream)
  Minimal overhead, maximum benefit:
  - Consistancy and reproduceability for devs and users
  - Distribution & deployment server integration,
  - Improved deployment without risk of screwing up the system
- Packaging as a **marketing** activity for the upstream project. Easy way to extend your user base, and improve your community relationship and is a “competitive advantage”.
- New mantra: “**Package early, package always**”
- THE SOLUTION IS INDEED **CONTINUOUS PACKAGING** (whatever the tool)

Continuous Packaging Architecture



Goals

- Project-builder main goal: help **package continuously** being **agnostic**
  - **VCS agnostic:** no VCS but guys it's 21st century now, SVN, CVS, Mercurial, GIT and GIT/SVN, SVK....
  - **OS agnostic**: Linux: RPM, deb, ebuild, slack based, ... 150+ distro tuples made and counting – repositories for yum, urpmi, apt. Solaris pkg. HP-UX sd in roadmap
  - **Build environment agnostic**: local, VM (QEMU, KVM...), VE (chroot, Docker, rpmbootstrap, rinse, mock, debootstrap...), RM (build farm)
  - **No project impact:** preserves the md5sum of the delivered upstream sources. Can be completely external to the upstream project.
  - Avoids duplication of code and metadata
  - THE SOLUTION IS INDEED **CONTINUOUS PACKAGING** (with project-builder.org !)

DEMONSTRATION

Goodies

- **Project-builder provides additional goodies**
  - Easy VMs/VEs/RMs installation/setup. Updates not mandatory.
  - Macro system with perl variables to avoid duplication of metadata
  - Skeleton generation to help starting
  - Manages package delivery up to your repository (ssh based) with repository management (yum, apt, urpmi)
  - Integration of tests in the process
  - Manages patches/additional sources when not upstream
  - Checks validity of packages built (lintian, rpmlint)
  - Easy creation of new versions for upstream management
  - Manages website delivery, announces on mailing-lists

Docker containers Management

- **Phase 0** (img tag: d-v-a) - newve
  - Local Image Creation (debootstrap/rpmbotstrap)
  - Reuse Image from Docker Trusted or Local Registry
- **Phase 1** (img tag: d-v-a-pb) - (sbx2|)setupve
  - adds pb tools and build account
- **Phase 2** (img tag: d-v-a-pb-proj) - prepve
  - adds project build dependencies
- **Phase 3** (container) - build2ve
  - Build project in the Docker container


Minimal OS Image

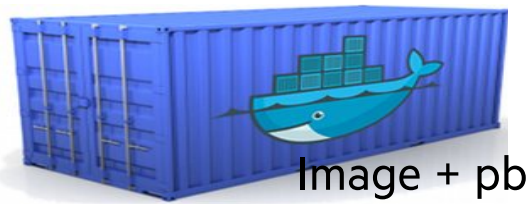


Image + pb tools


Image Proj1 deps


Image Proj2 deps


Proj1 container


Proj2 container

DEMONSTRATION

To be done

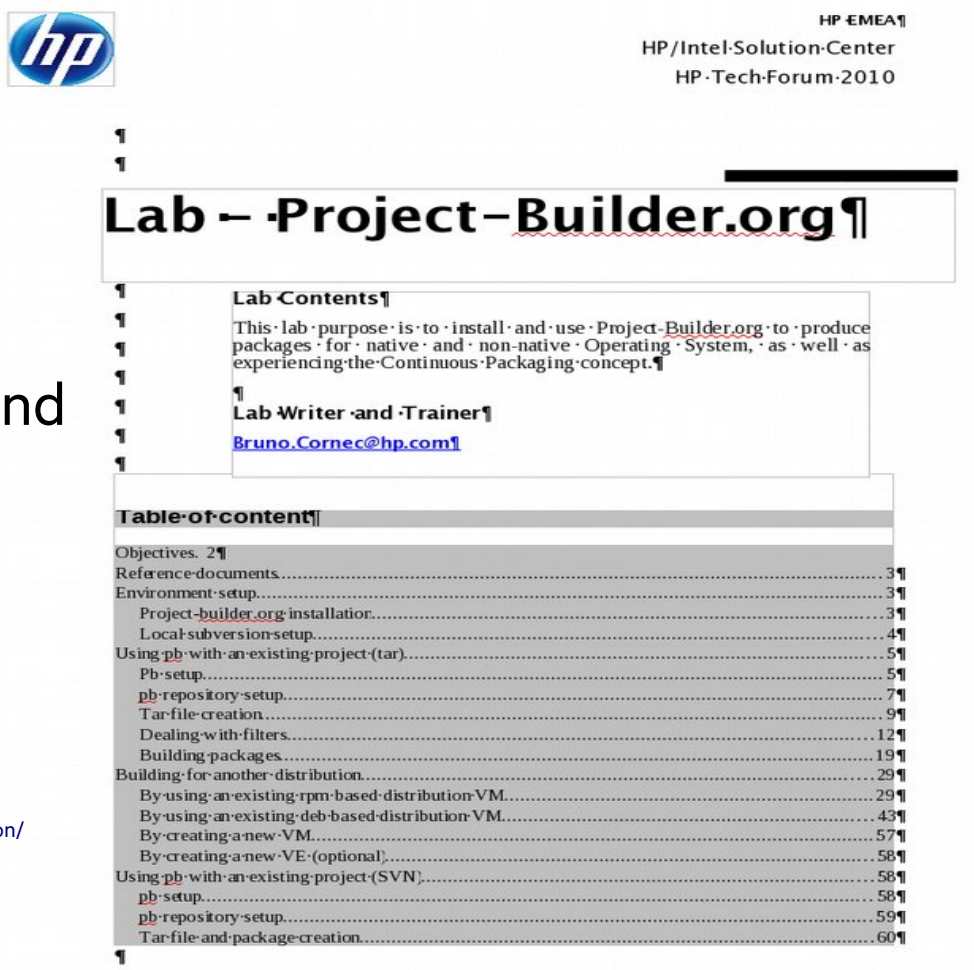
- Support for libvirt, virsh, openstack API, LXC (?)
- Other CMS (Bazaar, ...) only when/if needed
- Other VMs (VMWare, Xen, ...) only when/if needed
- Multiple delivery means – usage of ansible ?
- Look at interactions with Buildbot
- Config-Model for configuration file management
- REST API and Web Interface would be nice.

  Any contributor ?

Learn Project-Builder.org

- Start with the **Lab** (63 pages) at
  http://www.project-builder.org
- Use **man** (pb, pb.conf + 8 ProjectBuilder::* man pages)
- Use the **mailing-lists** pb-announce and pb-devel at http://www.mondorescue.org/sympa
- Use **examples** from
  http://trac.project-builder.org/browser/projects/
- Use **blog article** for Docker from

https://brunocornec.wordpress.com/2015/11/25/project-builder-org-0-12-7-is-out-and-0-13-1-will-be-soon/



HP EMEA
HP/Intel Solution Center
HP Tech Forum 2010

Lab – Project-Builder.org

Lab Contents

This lab purpose is to install and use Project-Builder.org to produce packages for native and non-native Operating System, as well as experiencing the Continuous Packaging concept.

Lab Writer and Trainer

Bruno.Cornec@hp.com

Table of content

Web Resources

Project-Builder.org Web sites:

- http://www.project-builder.org
- http://trac.project-builder.org

Projects using project-builder.org:

- http://www.project-builder.org (of course :-)
- http://www.mondorescue.org
- http://www.fossology.org
- http://www.linuxcoe.org

Related tools

Project-Builder.org is mostly suited for upstream projects wanting to package their applications

Distributions provide each their build tools

- SuSE: Open Build Service (Multi distro, Web based, BaaS)
- Fedora (Koji)
- Mandriva/Mageia (Youri, mgarepo...)

Other complementary tools:

- Buildbot
- Parallel::ForkManager, DBI, DB::SQLite, File::MimeInfo, Mail::Sendmail

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