



Packages on FreeBSD

Baptiste Daroussin
bapt@FreeBSD.org

BSDCan 2013
Ottawa, May 18th



Summary

- Pre-requisite
- Building your own packages
- Publish your packages
- Deploy your packages
- Packaging base
- Overview of main features of pkg 1.1



Pre-requisite

- Why building packages :
 - Have specialised packages (non default options)
 - Specific version (perl, mysql, ...)
 - Have custom packages
 - Have home made packages
 - Maintaining your own stable branch of packages



Building : Old time

- Pointyhat (portbuild)
 - Complex,
 - Hard to debug
 - Some design bugs
- Tinderbox :
 - Slow
 - Complex
 - Same design bugs as pointyhat
- Home made scripts :
 - Rarely real clean room (sandbox)
 - Often slow
 - Need maintainance



Poudrière



- Building package Factory
- Package testing
- Package building Q/A (can be very strict)
- OS stressing tool
- Scalability testing tool



Poudrière



- Clean room building
- Cross compilation
- Massively parallelized and tunable
- Deep integration with ZFS
- Also available without ZFS
- Incremental builds
- Work with portshaker for external ports trees
- Can package ports and (soon) base



Building: poudrière



- Head jails
 - `poudriere jails -c -j 10amd64 -v head -m svn`
- Head jails with custom patches
 - `poudriere jails -c -j 10amd64_broken -v head -m svn -p my-broken.patch`
- Release jails
 - `poudriere jails -c -j 91amd64 -v head -m svn`
- Cross jail
 - `poudriere jail -c -j 10mips64 -v head -m svn -a mips64`



Building: poudrière



- Creating a ports tree :
 - `poudriere ports -c`
- Creating apache package:
 - `poudriere bulk -j 10amd64 www/apache24`
- Build specialized sets
 - `poudriere bulk -j 10amd64 -f pkg.list -z myspecialbox`
- Build the whole ports tree:
 - `poudriere bulk -j 10amd64 -a`



Publishing packages

- Creating a repository:
 - `pkg repo /path/to/your/repo`
 - `pkg repo /path/to/your/repo signature_key`
- Package for autoseup
 - Pack a simple file (`/usr/local/etc/pkg/repos/woot.conf`)

```
packagesite: http://mycompany/app/${ABI}
mirror_type: http
key : /usr/local/pkg/keys/woot.key
enabled : yes
```



Deploying packages

- Setup a new repository:
 - `pkg add http://mycompany/myapprepo.txz`
 - Done
- Supported protocols :
 - `http(s)://`
 - `ftp://`
 - `ssh://`
 - `file://`



Packaging base

- Packaging:
 - `poudriere pkgbase -j 10amd64 -m svn -v head -s scriptdir`
 - `poudriere pkgbase -j 10amd64 -v head`
- Result:
 - `freebsd-base-10.0.20130517.txz`
 - `freebsd-kernel-10.0.20130517.txz`
 - `freebsd-kernel-debug-10.0.20130517.txz`
 - `freebsd-docs-10.0.20130517.txz`
 - `freebsd-games-10.0.20130517.txz`
 - `freebsd-lib32-10.0.20130517.txz`

-



Packing base

- Upgrade from 8.3 to 9.1
 - `freebsd-update -r 9.1-RELEASE upgrade install`
 - `reboot`
 - `freebsd-update install`
 - `pkg-static upgrade -f`
- Tomorrow (already doable now)
 - `pkg install -f freebsd-kernel freebsd-base`
 - `pkg-static upgrade -f`



pkg 1.1

- full multi-repository support
- incremental catalog update
- simplified public API
- massive performances improvements
- pkg lock/unlock
- pkg annotate
- ssh protocol support
- plugins



Plugins

- Simple API
- 2 kind:
 - New commands
 - Hooks
- Consistent and unified configuration file
- Example :
 - pkg-plugin-serve
 - pkg-plugin-zfs



pkg next

- GSoC :
 - Pluggable solver
 - Requires/Provide
 - Improved multi repository (priorities)
- Planned :
 - Skip files via pattern
 - Eliminate as much as possible scripts (cross-installation friendly)



Questions ?

<http://github.com/freebsd/pkg>
<https://fossil.etoilebsd.net/poudriere>