

Pkgng, ports and poudriere

Этот доклад НЕ является

- `man pkg`
- `man poudriere`
- `cat /usr/ports/README`

Этот доклад про:

- Преимущества pkg перед pkg_*
- Неочевидные проблемы особенности ports
- Poudriere = пороховая бочка в пакетостроении

1. pkg

- Pkg aka pkgng = попытка обновить пакетный менеджер для FreeBSD
- Зачем – подробно на <http://wiki.freebsd.org/PkgPrimer>

Сравнение

Function	legacy pkg_*	pkgng
get list of installed packages	pkg_info	pkg info
get basic information about a package	pkg_info pkgname-pkgversion	pkg info pkgname OR pkg info category/name OR pkg info pkgname-pkgversion
get detailed information about a package	N/A	pkg info -f pkgname OR pkg info -f category/name OR pkg info -f pkgname-pkgversion
search for a remote package	N/A	pkg search mypackage OR pkg search category/name OR pkg search pkgname-pkgversion
remove unused packages install as a dependency	N/A	pkg autoremove

Сравнение

Function	legacy pkg_*	pkgng
create a remote repository	N/A	pkg repo /path/to/a/directory/containing/packages
manipulate packages in a jail	N/A	pkg -j <name_or_id_of_a_jail> <command>
show statistic about installed packages	N/A	pkg stat
checking for missing dependency (and try to fix them)	N/A	pkg check -d

Мало? Вот ещё:

- Txz = лучше компрессия, меньше трафика
- YAML Manifest = читаемо
- Arch = в пакете записано, “чей” он
- Options = известно как собран пакет
- SQLite DB = чёткая структура хранения мета информации (в том числе обратные зависимости)

Это ещё не всё

- At least one of the packages provided by the repository is not compatible with your abi:
freebsd:10:x86:32
- Подключение неверного репозитория видно сразу = экономия трафика и времен
- Multirepo?... Multirepo!!!
- Pkg = no dependencies at all
- Pkg = next generation. Just live with that.

/usr/ports

- Одно дерево на всех
- Нет 100% гарантий что всё дерево
СОВМЕСТИМО
- Release early, release often = Broke often
- Obsolete versions
- Lacking of human testing
- Compiled = working

/usr/ports

- Патчи не всегда охотно принимает upstream
- Новый софт кто-то должен портировать
- 3rd party зависимости не ускоряют процесс

“Срез” портов

- Новая концепция, о которой думают
- Теги/Бранчи для портов
- Больше усилий на поддержание актуальности
- Больше возможностей построить гарантированно рабочий “набор” пакетов
- Неочевидные плюсы и минусы

poudriere

- Почему не tinderbox?
- Медленно
- From scratch – new vision, new approach
- `wc -l /usr/local/share/poudriere/*` = 2889 total
- Clean, understandable shell code
- All states/info stored in zfs properties = no DB
- Can work with `pkg_/pkg`

Poudriere under hood

poudriere jail [parameters] [options]

- Parameters:

- -c -- create a jail

....

Poudriere under hood

- j jailname -- Specifies the jailname
- v version -- Specifies which version of FreeBSD we want in jail
- a arch -- Indicates architecture of the jail: i386 or amd64
(Default: same as host)
- f fs -- FS name (tank/jails/myjail)
- M mountpoint -- mountpoint
- m method -- when used with -c forces the method to use by default
"ftp", could also be "svn", "svn+http", "svn+ssh",
"csup" please note that with svn and csup the world
will be built. note that building from sources can use
src.conf and jail-src.conf from localbase/etc/poudriere.d
- t version -- version to upgrade to

Poudriere under hood

poudriere ports [parameters] [options]

- Parameters:
- `-c` `-- create a portstree`
-

Poudriere under hood

- F -- when used with -c, only create the needed ZFS filesystems and directories, but do not populate them.
- p name -- specifies the name of the portstree we work on . If not specified, work on a portstree called "default".
- f fs -- FS name (tank/jails/myjail)
- M mountpoint -- mountpoint
- m method -- when used with -c, specify the method used to update the tree by default it is portsnap, possible usage are "csup", "portsnap", "svn", "svn+http", "svn+ssh"

Poudriere bulk

Parameters:

-f file -- Give the list of ports to build

Options:

-c -- Clean the previous built binary packages

-D -- Debug mode, display more information

-t -- Add some testings to package building

-s -- Skip sanity

-J n -- Run n jobs in parallel

-j name -- Run only on the given jail

-p tree -- Specify on which ports tree the bulk will be done

-w -- Save WRKDIR on failed builds

Poudriere life cycle

- Create @clean snapshot of jail
- Remove obsolete packages
- Remove packages with missing dependencies
- Calculate build order
- Check if options changed = some ports required to be rebuild
- ...

Poudriere life cycle

- ...
- Zfs clone snapshot for each dependency build
- Build all in parallel (bulk -J 5, as example)
- Create packages
- Deinstall
- Run `zfs diff -FH @prebuild ${JAILFS} = PROFIT!!!`
- Build pkgng repo

Poudriere running example

====>> Mounting system devices for t-10-i386

/etc/resolv.conf -> /zpl/poudriere/jails/t-10-i386/etc/resolv.conf

====>> Starting jail t-10-i386

====>> Mounting ports filesystems for t-10-i386

====>> Populating LOCALBASE

====>> Cleaning up old logs

====>> Calculating ports order and dependencies

====>> Sanity checking the repository

====>> Deleting stale symlinks

====>> Cleaning the build queue

====>> Starting using 4 builders

====>> [01] Starting build of graphics/xpdf

====>> [02] Starting build of sysutils/fusefs-ntfs

====>> [03] Starting build of x11/xrandr

====>> [04] Starting build of graphics/mupdf

Poudriere cookies

- HTML statuses
- Repo creation
- Manual ports upgrade/patch
- Easy to tweak (small clean codebase)

Questions?

- ?