

pkgsrc bulk-builds für illumos SmartOS

Thomas Merkel tm@core.io

2020-12-15

Agenda

- whoami
- Was ist pkgsrc?
- Vorteile von pkgsrc
- Verwendung von pkgbuild Zonen
 - pkgbuild Struktur
 - Konfiguration
 - Paket erstellen
- Eigene Bulk Builds erstellen
- Fazit und Fragen

- Server Ninja / Partner bei [SkyLime GmbH](#)
- Hosting und Cloud Lösungen auf SmartOS unter [reco-systems](#)
- Unix und Linux Consultant seit ~15 Jahren
- Fokus auf DevOps, Automatisierung, Infrastruktur und “Cloud”

Was ist pkgsrc

- Paketverwaltung für UNIX-artige Betriebssysteme
- Unterstützt 23 unterschiedliche Betriebssysteme
 - AIX, BSD, Solaris, MacOS, ...
- Standardpaketverwaltung für NetBSD und SmartOS
- Installation via Quellcode
- **Möglichkeit zur Erstellung von Binärpaketen**
- Release jedes Quartal
- 22.500 Pakete

```
.
├── archivers
│   ├── 9e
│   │   └── patches
...
├── sysutils
│   ├── 855resolution
│   │   └── patches
│   ├── bacula
│   │   ├── files
│   │   └── patches
...

```

Vorteile von pkgsrc

- Unterstützt viele Betriebssysteme und Plattformen
- Verwendung von Makefiles
- Einfaches Tooling durch Macros und BSD Make
- Bootstrap ohne root-Rechte möglich

Nachteile von pkgsrc

- CVS als Versionsverwaltung
- Lange Integrationszeit für patches für pkgsrc mk/*

Verwendung von pkgbuild Zonen

- **pkgbuild** Skripte und Tooling von Joyent um relativ einfach pkgsrc-Pakete zu erstellen
- **pkgbuild Zone** SmartOS Zone die bereits das Tooling und Bootstrap Daten von pkgsrc beinhaltet

```
[gz] $ imgadm import 1f538e62-11e9-11eb-b7d2-338b1de5418c
```

Erstellung eines pkgbuild.json mit der image_uuid

```
[gz] $ vmadm create -f pkgbuild.json
```


Verwendung von pkgbuild Zonen

pkgbuild Struktur

/data

|-- bulklog

| |-- 2020Q3

|-- chroot

| |-- dev-2020Q3-x86_64

|-- distfiles

| |-- binutils-2.34.tar.bz2

...

-- packages

| |-- SmartOS

| |-- 2020Q3

|-- pkgbuild

| |-- conf

| | |-- 2020Q3-tools

| | |-- 2020Q3-x86_64

| | |-- upstream-trunk

| |-- include

| |-- pkglist

Verwendung von pkgbuild Zonen

Konfiguration

- Wichtigsten Konfigurationen befinden sich unter `/data/pkgbuild/conf`
- Jeder Ordner entspricht einer Version, OS und Architektur, nur bei illumos fehlt das OS
 - `2020Q3-x86_64`
 - `netbsd-trunk-amd64`
- Mit `.local`-Dateien können einfach Konfigurationsparameter überschrieben werden

Verwendung von pkgbuild Zonen

Paket erstellen

- Verwendung von Sandboxen und bestehenden Bootstrap Kits
- Falls keine Signierten Pakete erstellt werden sollen keine Konfigurationsanpassung nötig
- Konfiguration von pkgsrc Paket Optionen unter `/data/pkgbuild/conf/PKGBUILDVERSION/mk.conf.local`
 - `PKG_OPTIONS.curl= libssh2 http2`
 - ...

```
[pkgbuild] $ run-sandbox 2020Q3-x86_64
```

```
[2020Q3-x86_64] $ cd /data/pkgsrc/www/nginx
```

```
[2020Q3-x86_64] $ bmake package
```

Eigene Bulk Builds erstellen

Erstellung eigener Pakete im bestehenden Joyent PREFIX (/opt/local unter SmartOS)

- Konfigurationsdatei `pbulk.conf` bzw. `pbulk.conf.local`
 - Eigene Paketliste via `limited_list=` angeben
 - `build_chroots` und `scan_chroots` je nach Leistung anpassen
- Nutzung von SSH Verbindungen zu localhost: `ssh-keygen; cat /root/.ssh/id_rsa.pub > /root/.ssh/authorized_keys; ssh root@127.0.0.1`

```
[pkgbuild] $ export PKGBUILD=2020Q3-x86_64  
[pkgbuild] $ run-jenkins-build
```

- Reports und Logs: `/data/bulklog/2020Q3/x86_64`
- Pakete: `/data/packages/SmartOS/2020Q3/x86_64`

Bulk Builds im eigenen PREFIX

Ugly Hacking: Erstellung eigener Pakete in /opt/extra anstatt in /opt/local.

```
[pkgbuild] # cd /data/pkgbuild/conf  
[pkgbuild] # cp -r 2020Q3-tools 2020Q3-extra
```

Mit gsed einfach alles umbenennen:

```
[pkgbuild] # cd 2020Q3-extra  
[pkgbuild] # for f in *; do gsed -i -e 's/tools/extra/g' $f; done  
[pkgbuild] # gsed -i -e 's/extextra/exttools/g' mk.conf  
[pkgbuild] # gsed \  
    's/CHECK_WRKREF=.* /CHECK_WRKREF=\t\t\ttools home/g' \  
    mk.conf
```

Erstellen eigener Paketliste /data/pkgbuild/pkglist/trunk-extra:

```
[pkgbuild] # echo "chat/irssi" \  
    > /data/pkgbuild/pkglist/trunk-extra
```

Bulk Builds im eigenen PREFIX

pbulk Kit muss nun für den neuen Prefix erstellt werden:

```
[pkgbuild] # mk-pbulk-bootstrap 2020Q3-extra
```

Nun kann das Bulk Building starten:

```
[pkgbuild] # export PKGBUILD=2020Q3-extra  
[pkgbuild] # run-jenkins-build
```

Bootstrap Kit erstellen:

```
[pkgbuild] # export PKGBUILD=2020Q3-extra  
[pkgbuild] # run-bootstrap
```

Mitmachen im [Work-In-Progress](#) (git-)Repository

- [Dokumentation](#)
- [pkgsrc.se Übersicht](#)
- [#pkgsrc](#) auf <irc://chat.freenode.net>