

AUR-Packages bauen und testen ohne ArchLinux im eigenen Docker-Container

Tübix

B1 Systems GmbH

2019-07-06



If you have any questions, comments or want to report errors in the training material please post them to doku@b1-systems.de.

The following names, products or services are registered trademarks:

Adaptec, AIX, Amazon, AMD, AMD Virtualization, AMD-V, AutoBuild, Cisco, DB2, Debian, Domino, the Gecko, IBM, Intel, Intel Logo, Intel Inside Logo, Intel Centrino Logo, iSeries, Java, JDBC, JVM, J2EE, Linux, Microsoft Windows, Microsoft Virtual PC, Nagios, Nagios Logo, Novell, N Logo, OpenStack, openSUSE, openSUSE Logo, Oracle, pSeries, PowerPath, QEMU, Qumranet, Qumranet Solid ICE, Red Hat, Red Hat Linux, Red Hat Enterprise Linux, Red Hat Shadowman Logo, Solaris, StarOffice, Sun, Sun Java, Sun Microsystems, SUSE, SUSE Linux, SUSE Linux Enterprise Server, Type Enforcement, Ubuntu, UNIX, VirtualBox, VMware, VMware Workstation, VMware Server, VMware ESX, YaST, Xen, Xen Logo, zSeries.

We want to point out that all soft and hardware names, trademarks and product names of the respective firms used in this manual remain property of their respective holders even if not mentioned in the list above or marked accordingly.

Inhaltsverzeichnis

0	Vorstellung B1 Systems	1
1	Vorwort	3
2	ArchLinux	5
3	AUR	7
4	Ein Paket beisteuern	10
5	PKGBUILD	14
6	Docker	18
7	Dockerfile	20
8	Aufgaben	23
9	Vielen Dank für Ihre Aufmerksamkeit	25

0 Vorstellung B1 Systems

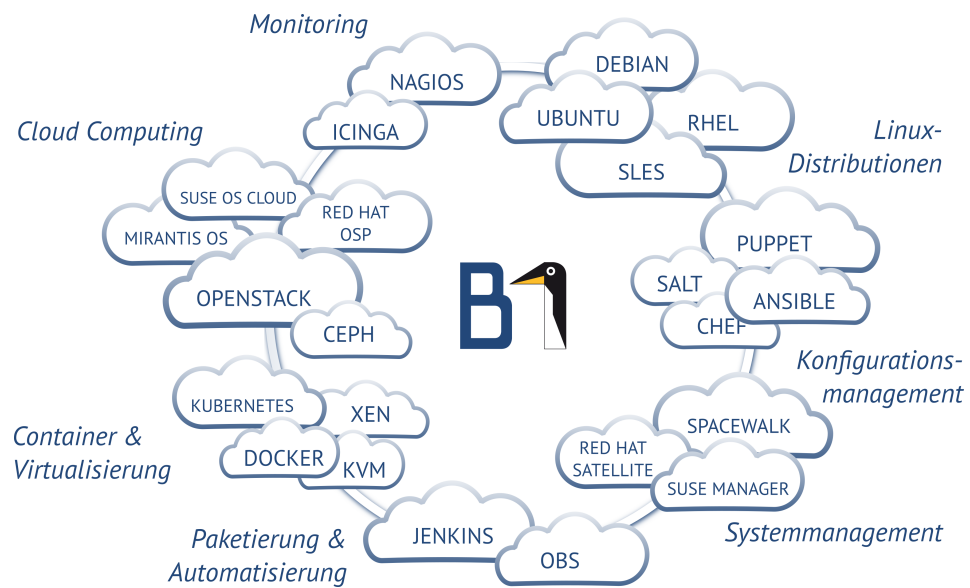


Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source-Themen
- national & international tätig
- über 100 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Leistungsangebot:
 - Beratung & Consulting
 - Support
 - Entwicklung
 - Training
 - Betrieb
 - Lösungen
- Standorte in Rockolding, Köln, Berlin & Dresden



Schwerpunkte



1 Vorwort



Vorwort/Disclaimer

B1 Systems GmbH

AUR-Packages bauen und testen ohne ArchLinux

4/24



Vorwort/Disclaimer

- Warum gerade ArchLinux?
 - ArchWiki
 - Lerneffekt
- Theorieteil am Anfang verkürzt
- Links verweisen auf englische Seiten
- Fragen -> Verständnisfragen sofort/am Ende der Folie

- Ist meine Lieblingsdistribution
- immer sehr aktuell
- für jeden geeignet, der sich wirklich mit Linux auseinander setzen möchte
- Das ArchWiki ist eine sehr gute Dokumentation
- Forum nutze ich eher weniger bis garnicht
- Man muss viel selbst machen, aber man weiß wie das dann geht
- Links verweisen auf englische Seiten - meist mehr Infos
- Theoretischer Anfänge für Grundlagen zum reinen Verständnis
- weiterführendes gern als Link *B1-Vortraege*
- Fragen - Verstaendnisfragen sofort/am Ende der Folie - Begriffe gern nachfragen
- Bitte nicht Siezen

2 ArchLinux






ArchLinux

- 2002 – Neuentwicklung aus Crux nach dem KISS-Prinzip
- KISS = Keep it simple stupid → *Mach es so einfach wie möglich*
- Rolling Release
- vergleichbar mit openSUSE Tumbleweed
- AUR = ArchLinux User Repository,
<https://aur.archlinux.org> – ähnlich wie PPAs in Ubuntu
- Pacman – Paket-Manager
- AUR-Helper – für einfache AUR-PKG Installation
- AUR-Helper: <https://github.com/Jguer/yay>
- PKGBUILD:
<https://wiki.archlinux.org/index.php/PKGBUILD>
- mehr im ArchLinux-Wiki unter
https://wiki.archlinux.org/index.php/Arch_Linux

- Gibts seit 2002 - war eine Neuentwicklung aus Crux nach dem KISS-Prinzip
- KISS steht fuer Keep it simple stupid zu Deutsch-> Mach es so einfach wie möglich
- Rolling Release Arch hat keine klassischen Versionen (OpenSuse 15/Fedora 29)
- als eher vergleichbar mit OpenSuse Tumbleweed
- AUR = ArchLinux User Repository -*ähnlich wie PPA's in Ubuntu*
- Pacman ist der Paket-Manager
- AUR-Helper -> Programme fuer einfache AUR-PKG Installation
- yay -> AUR-Helper
- PKGBUILD

3 AUR



AUR – ArchLinux User Repository

B1 Systems GmbH AUR-Packages bauen und testen ohne ArchLinux 8/24



AUR 1/2

- Übersicht, Suche & "Paketverwaltung"
- Pakete out-of-date flaggen/makieren
- Paketanfragen (requests) -*mit Begründung*
 - delete
 - merge
 - oprhan

- Übersicht, Suche & "Paketverwaltung"
 - Übersicht über verfügbare Pakete
 - Suche nach Paketen/Maintainern/Submitttern/Beschreibung
 - als Maintainer / angemeldeter User unterschiedliche Aktionen moeglich
- Pakete out-of-date flaggen/makieren
 - Upstream hat aktuellere Version als das AUR-pkg
- Paketanfragen (requests) - mit Begruendung - Freitextfeld
- Begruendung wird auch geprueft
 - delete
 - merge
 - oprhan




AUR 2/2

- Paketinformationen
 - Beschreibung
 - Maintainer (!= Submitter)
 - Upstream-URL
 - Licenses
 - Keywords
 - Quelle
 - Abhaengigkeiten
- Kommentare
- git clone URL

- Paketinformation
 - Beschreibung
 - Maintainer
 - Upstream-URL
 - Licenses
 - Keywords
 - Quelle
 - Abhaengigkeiten
- Kommentare
- git clone URL

4 Ein Paket beisteuern



Ein Paket beisteuern

B1 Systems GmbH

AUR-Packages bauen und testen ohne ArchLinux

11/24



Ein Paket beisteuern 1/3

- Arch package guidelines
- Package etiquette
- AUR submission guidelines
- Paketanfragen (requests) -*mit Begründung*
 - delete
 - merge
 - oprhan
- Vorher den Maintainer versuchen zu kontaktieren.
- ungefähr 2 Wochen warten
- Jede Anfrage wird auf die Requests-Liste geschickt und von TU bearbeite

- Arch package guidelines
 - grundlegende Infos wie man ein PKGBUILD
 - hier gibts auch ein Beispiel fuers PKGBUILD
- Package etiquette
- AUR submission guidelines
- Paketanfragen (requests) -*mit Begründung*
 - delete – Paket löschen
 - merge – 2 Pakete zusammenführen
 - oprhan – Maintainer ist "tod"
- Vorher den Maintainer versuchen zu kontaktieren.
- ungefähr 2 Wochen warten
- Jede Anfrage wird auf die Requests-Liste geschickt und von TU bearbeite



Ein Paket beisteuern 2/3

- Übersicht über verfügbare Pakete
- Suche nach Paketen/Maintainern/Submittern/Beschreibung
- als Maintainer / angemeldeter User unterschiedliche Aktionen möglich
- Pakete out-of-date flaggen/makieren
- Upstream hat aktuellere Version als das AUR-pkg
- Paketanfragen (requests) - mit Begründung - Freitextfeld
- Begründung wird auch geprüft

- Übersicht über verfügbare Pakete
- Suche nach Paketen/Maintainern/Submittern/Beschreibung
- als Maintainer / angemeldeter User unterschiedliche Aktionen möglich
- Pakete out-of-date flaggen/makieren
- Upstream hat aktuellere Version als das AUR-pkg
- Paketanfragen (requests) - mit Begründung - Freitextfeld
- Begründung wird auch geprüft

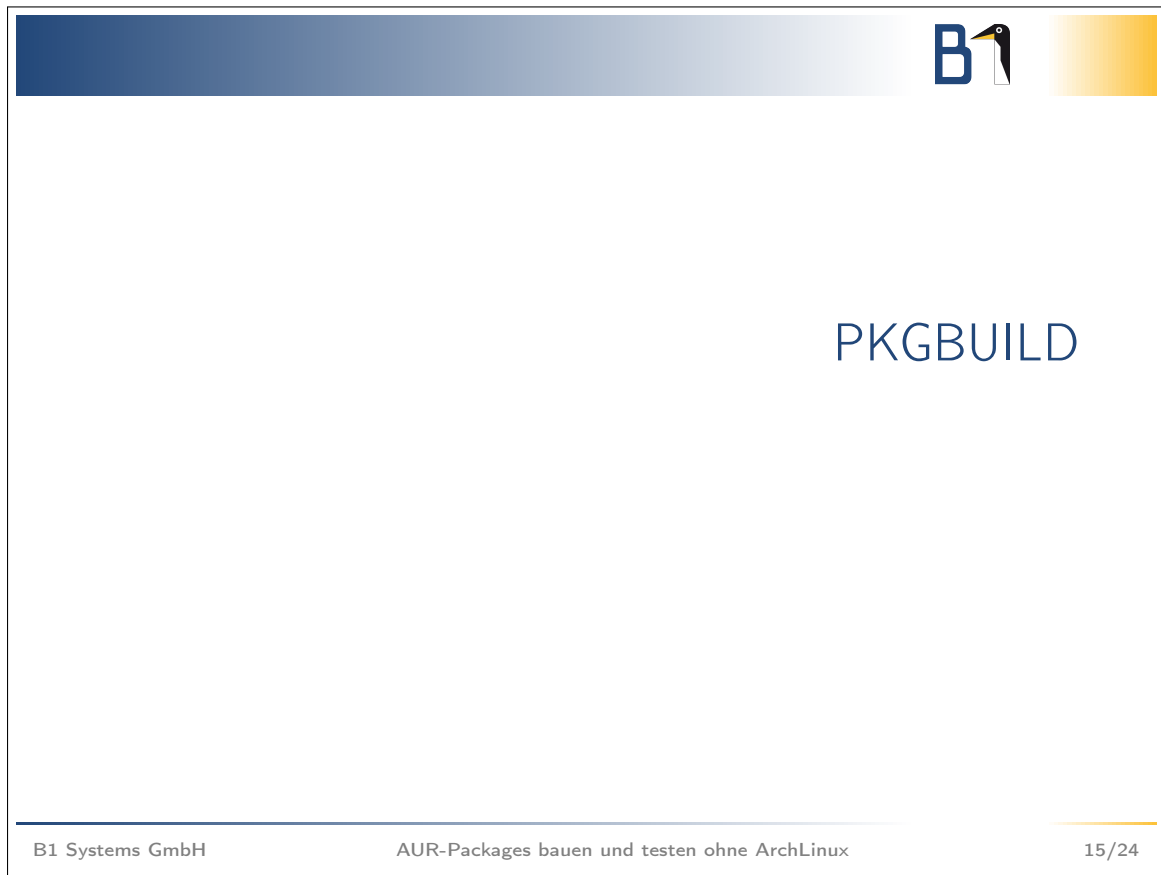


Ein Paket beisteuern 3/3

- Paketinformation
 - Beschreibung
 - Maintainer
 - Upstream-URL
 - Licenses
 - Keywords
 - Quelle
 - Abhängigkeiten
- Kommentare
- git clone URL

- Paketinformation
 - Beschreibung – Was macht diese Software? Was kann man damit machen?
 - Maintainer – Wer betreut das Paket?
 - Upstream-URL – Wo kommt das denn her?
 - Licenses – Unter welcher Lizenz steht das? – Später mehr
 - Keywords – SSEO"
 - Quelle – Was kommt mit dem Paket mit
 - Abhängigkeiten – Was braucht das Paket vorher/beim Bauen?
- Kommentare – Usercomments – Austausch/Diskussion
- git clone URL – einfacher runterladen"

5 PKGBUILD





PKGBUILD 1/2

- gute Infos im ArchWiki
<https://wiki.archlinux.org/index.php/PKGBUILD>
- makepkg im {base-devel}
- obligatorische Variablen:
 - pkgname – Packagename
 - pkgver – Packageversion
 - pkgrel – Packagerelease
 - arch – Architektur (i686, x86_64, any)
- empfohlene Variablen:
 - license (Warning im Build)
 - checksums (md5 oder sha256) - *gen by updpkgsums*
 - pkgdesc – Beschreibung

- gute Infos im ArchWiki
- makepkg im {base-devel}
- obligatorische Variablen
 - pkgname – PackagenameS
 - pkgver – Packageversion
 - pkgrel – Packagerelease
 - arch – Architektur (i686, x86_64, any)
- empfohlene (weitere) Variablen – *meine Empfehlung*
 - license (ohne gibt's nen Warning im Build)
 - checksums (md5 oder sha256) – sollte halt drin sein
 - pkgdesc – Beschreibung



PKGBUILD 2/2

- Maintainer – # Maintainer name <e@mail.adresse>
- build()
- package()
- .SRCINFO wird aus der PKGBUILD mit makepkg erzeugt
 - \$ makepkg -printsrcinfo > .SRCINFO
 - erster Commit = PKGBUILD & .SRCINFO
- guter Vortrag ÄUR Pakete bauen für Anfängerin der Mediathek des CCC:
 - <https://media.ccc.de/v/gpn18-10-aur-pakete-bauen-fr-anfnger>

- gute Infos im ArchWiki
- makepkg im {base-devel}
- obligatorische Variablen
 - pkgname – PackagenameS
 - pkgver – Packageversion
 - pkgrel – Packagerelease
 - arch – Architektur (i686, x86_64, any)
- empfehlende (weitere) Variablen – *meine Empfehlung*
 - license (ohne gibts nen Warning im Build)
 - checksums (md5 oder sha256) – sollte halt drin sein
 - pkgdesc – Beschreibung
- Maintainer – # Maintainer name <e@mail.adresse>
- build()
- package()

- .SRCINFO wird aus der PKGBUILD mit makepkg erzeugt
 - .SRCINFO enthält die Infos fuer AUR
 - erster Commit muss PKGBUILD und .SRCINFO erhalten
- guter Vortrag in der Mediathek des CCC

6 Docker






Docker

- 2013 – erstes Release
- Apache License 2.0
- Architektur: x86_64, ARM, s390x, ppc64le
- Binaries: alle 3 „großen“ OS
- Virtualisierung ohne VM
- chroot + cgroups + NetworkNS in „beautiful & Noobfreundlich“
- DockerHub – *Docker Container Repository*

- 2013 – erstes Release
- Source Code ist unter Apache License 2.0
- laeuft auf x86-64, ARM, s390x, ppc64le
- Binaries gibts fuer alle 3 grossen OS
- Virtualisierung ohne VM (unter Windows & MacOS teilweise schon)
- so aehnlich chroot + cgroups + NetworkNS, aber in schon & Noobfreundlich
- DockerHub – *Docker Container Repository*
- guter Vortrag vom Netz39

7 Dockerfile



Dockerfile

B1 Systems GmbH

AUR-Packages bauen und testen ohne ArchLinux

20/24



Dockerfile

```
# version:  z.y
# git:      https://github.com/username/repo
# description: was mache ich hier bloss...
# docs: docs.url

FROM base/image

ENV VARIABLE="content"

RUN befehle -switch \
&& befehle

USER username
WORKDIR /home/username

RUN befehle -swich

RUN befehle
```

- FROM base/image
 - Was ist unser Basisimage?
 - in der Regel zu finden im DockerHub
 - bringt Basis-System mit
- ENV VARIABLE="content"

ENV enviroment

 - Umgebungsvariablen fuer den Container
 - bspw. Hostname
- RUN befehle -switch
 - RUN -> befehl ausführen
 - oft fuer die Installation von Paketen via PackageMgmt genutzt
 - Anlegen von Benutzern & Zuweisung von rechten
- USER username – unser Nutzer
- WORKDIR /home/username – in welchem Verzeichnis wollen wir arbeiten?

7 Dockerfile

- Bsp mit Docker-Apache erzaehlen.

8 Aufgaben



Aufgaben

B1 Systems GmbH

AUR-Packages bauen und testen ohne ArchLinux

22/24



Aufgaben

- 1 Outdated Packages
- 2 eigenes PKGBUILD schreiben
- 3 Dockerfile aus Ansible

1. Outdated Package

- Outdated/orphaned Package im AUR suchen finden
- PKGBUILD auf aktuelle Version updaten
- Installation im frischen Docker testen


2. Eigenes PKGBUILD schreiben

- Software XYZ *nicht* im AUR finden
- PKGBUILD schreiben
- Installation im frischen Docker testen

3. Dockerfile aus Ansible

- Dockerfile mit Ansible erstellen
- Ansible-Docu zu Dockerfiles lesen
- Playbook schreiben

9 Vielen Dank für Ihre Aufmerksamkeit



Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de
oder +49 (0)8457 - 931096

B1 Systems GmbH - Linux/Open Source Consulting, Training, Development & Support