Docker Images in Debian-Paketen

Jiří Kraml
Chemnitzer Linuxtage 2019

Ziel

- Mittels apt einen Docker Container starten

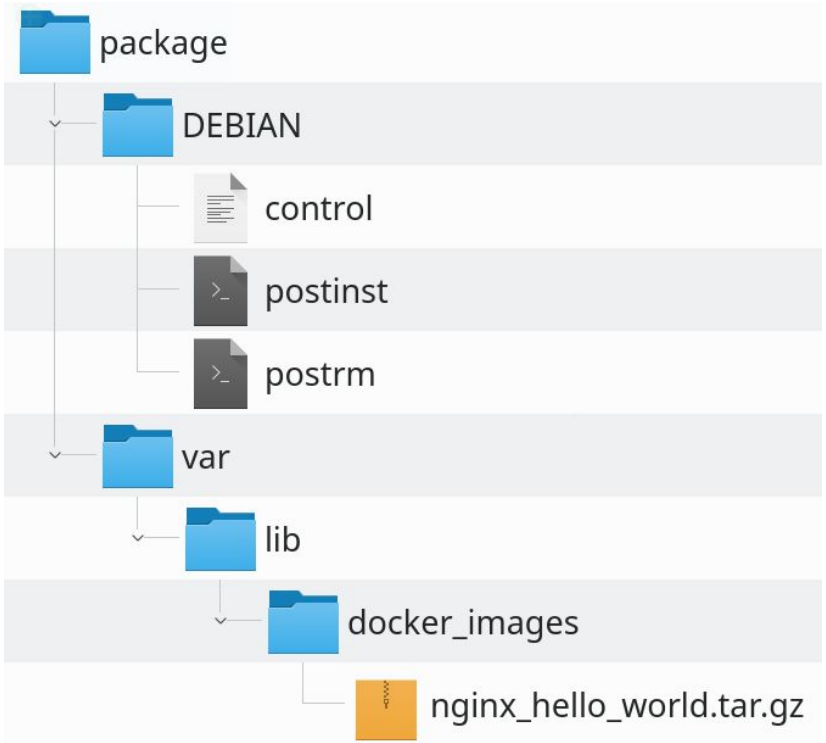
- Sehr simples Image
  - Nginx mit hello-world-page
- Ubuntu 18.04

Ablauf

- Wie ein Package aussieht
- Was zu installieren ist
- Demo
- Updates?
- Packages splitten?

Was ist in einem Package?

- Packageinformationen
- Hilfsskripte
- Dateien



Package Metadata

```
Package: docker-nginx-helloworld
Version: 1.0.0
Architecture: all
Maintainer: Jiri Kraml <jiri.kraml@zigpos.com>
Homepage: https://zigpos.de
Description: Hello World nginx in Docker
Depends: docker.io | docker-ce-cli
```

Abhängigkeiten

- Docker, aber welches?

- Standardquellen: docker.io
- Docker-Repository: docker-ce-cli

```
Depends: docker.io | docker-ce-cli
```

Was zu installieren ist

- Docker Image
- Container

Image installieren

Image dem Dockerdaemon bekannt machen
-> docker save/load

Inhalt:

- Image-Datei
- Postinst-Skript, welches docker load aufruft

Container einrichten & automatisch starten

- Autostart mit:
  - docker run mit Restart
  - docker start/stop
  - docker-compose up/down

Container einrichten (docker run --restart …)

--restart: always oder unless-stopped?

Inhalt:

- postinst-Skript, welches docker run aufruft

--restart policy
Restart policy to apply when a container exits. Supported values are:

Policy	Result
no	Do not automatically restart the container when it exits.
on-failure[:max-retries]	Restart only if the container exits with a non-zero exit status. Optionally, limit the number of restart retries the Docker daemon attempts.
always	Always restart the container regardless of the exit status. When you specify always, the Docker daemon will try to restart the container indefinitely. The container will also always start on daemon startup, regardless of the current state of the container.
unless-stopped	Always restart the container regardless of the exit status, but do not start it on daemon startup if the container has been put to a stopped state before.

Container einrichten (docker start/stop)

Container muss erst erstellt werden

Init-System muss docker start und stop aufrufen
-> systemd Unit

Inhalt:
- systemd .service-Datei
- postinst-Skript

```
[Unit]
Description=Hello World Nginx

[Service]
ExecStart=/usr/bin/docker start hello-world-nginx
ExecStop=/usr/bin/docker stop hello-world-nginx
Type=oneshot
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Container einrichten (docker-compose up/down)

docker-compose.yml benötigt

Init-System muss docker-compose aufrufen

-> systemd Unit

Inhalt:

- systemd .service-Datei
- docker-compose.yml
- postinst-Skript

```
version: '3'
services:
  hello-world-nginx:
    image: "zigpos/hello-world-nginx:2019a"
    ports:
     - "80:80"
```

Demo

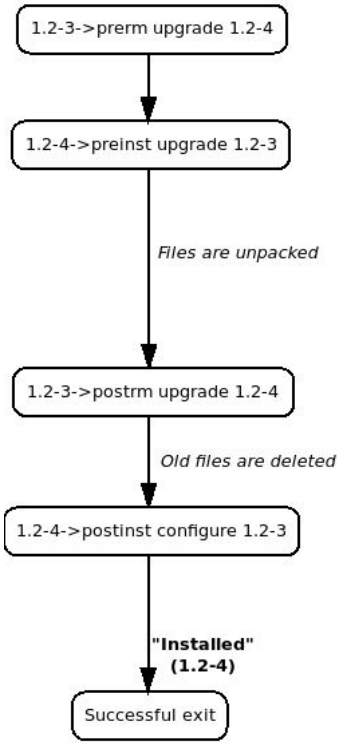
Updates

Ablauf:

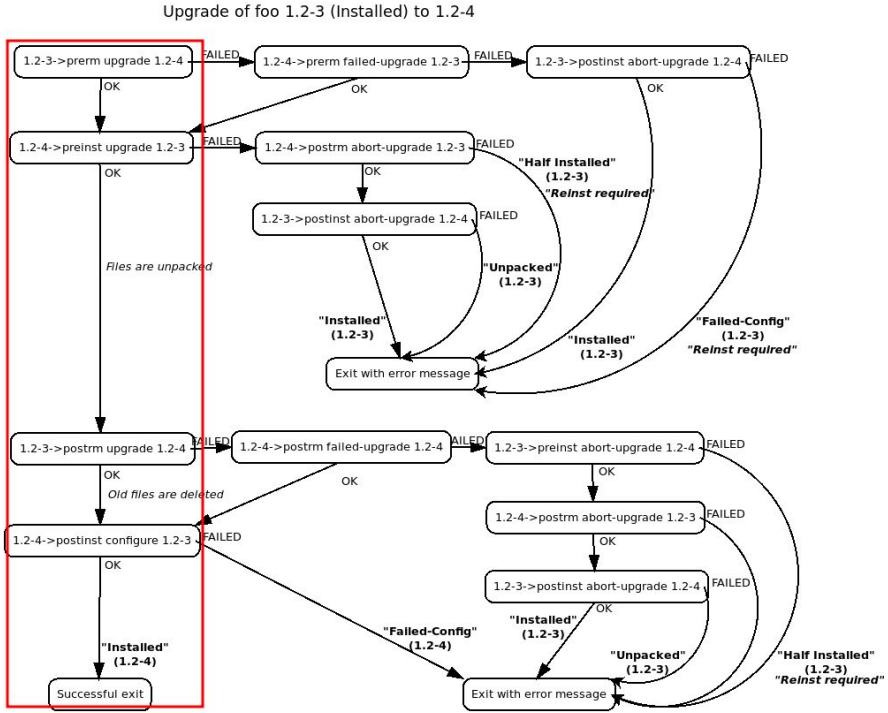
1. Container stoppen & entfernen
2. Image ersetzen
3. Container erstellen & starten

Happy Path:

1. altes prerm
2. neues preinst
3. (neue Dateien entpacken)
4. altes postrm
5. (alte Dateien löschen)
6. neues postinst



Updates


Upgrade of foo 1.2-3 (Installed) to 1.2-4
1.2-3->prerm upgrade 1.2-4
FAILED
1.2-4->prerm failed-upgrade 1.2-3
FAILED
1.2-3->postinst abort-upgrade 1.2-4
FAILED
OK
OK
OK
1.2-4->preinst upgrade 1.2-3
FAILED
1.2-4->postrm abort-upgrade 1.2-3
FAILED
"Half Installed"
(1.2-3)
"Reinst required"
OK
1.2-3->postinst abort-upgrade 1.2-4
FAILED
OK
Files are unpacked
"Unpacked"
(1.2-3)
"Installed"
(1.2-3)
"Installed"
(1.2-3)
"Failed-Config"
(1.2-3)
"Reinst required"
Exit with error message
1.2-3->postrm upgrade 1.2-4
FAILED
1.2-4->postrm failed-upgrade 1.2-4
FAILED
1.2-3->preinst abort-upgrade 1.2-4
FAILED
OK
OK
OK
Old files are deleted
1.2-4->postrm abort-upgrade 1.2-3
FAILED
1.2-4->postinst configure 1.2-3
FAILED
OK
OK
1.2-3->postinst abort-upgrade 1.2-4
FAILED
OK
"Failed-Config"
(1.2-4)
"Installed"
(1.2-3)
"Unpacked"
(1.2-3)
"Half Installed"
(1.2-3)
"Reinst required"
"Installed"
(1.2-4)
Successful exit
Exit with error message

Ein Package oder mehrere?

- Tradeoff: Modularität vs. Einfachheit

- Zusätzliche Package States möglich
  - Half-Installed
  - Half-Configured
  - Unpacked
  - …

- Monolithisches Package kann sehr groß werden

Zusammenfassung

- Übersicht über Debianpackages
- Einspielen der Images
- Erstellen der Container
- Einfaches Beispiel mit docker run --restart

Fragen?

Präsentation und Begleitmaterial unter:
gitlab.com/zigpos/public-events/clt2019