

# Einfach mehr aus einer Subskription herausholen

Zwei einfache Projekte für Red Hat Enterprise Linux

Jörg Kastning

Universität Bielefeld

15. März 2020



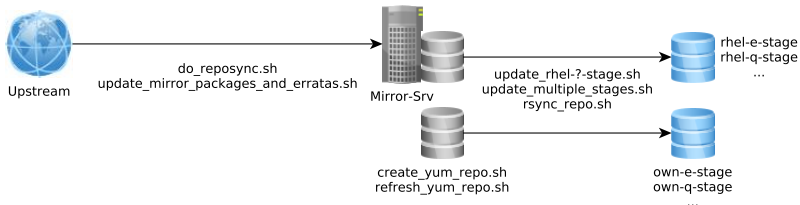
# whoami

|                        |                                                                          |
|------------------------|--------------------------------------------------------------------------|
| Name:                  | Jörg Kastning                                                            |
| Years spent in IT:     | 15                                                                       |
| Years spent as:        | SysAdmin, DevOp, IT Project Manager, etc.                                |
| Operating:             | Red Hat Enterprise Linux, VMware vSphere<br>Firewall, Loadbalancer, etc. |
| Currently employed at: | Bielefeld University                                                     |
| Member of:             | Red Hat Accelerators                                                     |

*"While I am not a representative of Red Hat and my views about Red Hat are my own, I am a member of Red Hat Accelerators community which gives me a connection to Red Hat and through which I engage with other Red Hat Accelerators."*

# What is the Poor Man's Mirror?

- ▶ Skriptsammlung rund um `reposync`, `rsync` und Co.
- ▶ Erstellt einen lokalen Spiegelserver der Red Hat Upstream-Repos
- ▶ Erstellt eure eigenen RPM-Repos
- ▶ Setzt Repos für Staging auf



# Voraussetzungen

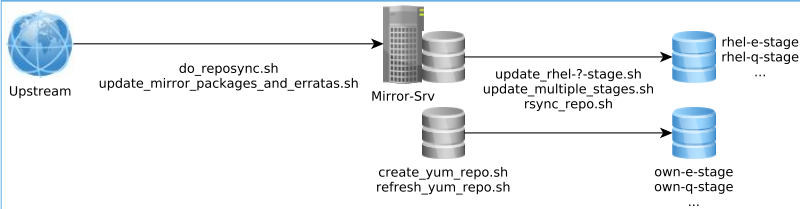
Mindestanforderungen zur Nutzung:

- ▶ Ein RHEL-Server mit einer gültigen Subskription
- ▶ Einen Web-Server eurer Wahl, um die Repos in eurem LAN bereitzustellen
- ▶ Die Pakete `reposync`, `rsync` und `createrepo` müssen installiert sein
- ▶ YUM sollte alle weiteren Abhängigkeiten automatisch auflösen

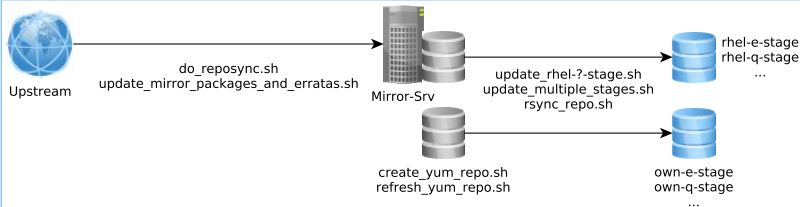
# Beschränkungen

- ▶ Es können nur Repos mit gültiger Subskription gespiegelt werden
- ▶ Nicht so mächtig wie Red Hat Satellite (aber auch nicht so teuer ;-))

## RHEL 7



## RHEL 8



# Wo kriege ich das Ding?

URL: `https://github.com/Tronde/poor-man-s-rhel-mirror/releases`

The screenshot shows the GitHub interface for the repository `Tronde / poor-man-s-rhel-mirror`. The repository has 5 watchers, 12 stars, and 5 forks. The navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Wiki, Security, Insights, and Settings. The 'Releases' tab is selected, showing a list of releases. The latest release is titled 'RHEL 8 version of the Poor Man's Mirror' and was released on Dec 25, 2019. The release description states: 'I've adapted the RHEL 7 version of the Poor Man's Mirror to fit RHEL 8. If you like to give it a try I would be happy if you drop some feedback and let me know how it works. Thanks!'. There are two assets listed: 'Source code (zip)' and 'Source code (tar.gz)'. The release is labeled '3.0.0' and has a 'Compare' button. A 'Draft a new release' button is also visible.

Tronde / **poor-man-s-rhel-mirror** Unwatch 5 ★ Star 12 🍴 Fork 5

<> Code Issues 0 Pull requests 0 Actions Wiki Security Insights Settings

**Releases** Tags

Draft a new release

Latest release

3.0.0  
ef99b4a

Compare

**RHEL 8 version of the Poor Man's Mirror** Edit

👤 Tronde released this on Dec 25, 2019

I've adapted the RHEL 7 version of the Poor Man's Mirror to fit RHEL 8.

If you like to give it a try I would be happy if you drop some feedback and let me know how it works.

Thanks!

Assets 2

Source code (zip)

Source code (tar.gz)

on Dec 25, 2019

3.0.0-beta

ef99b4a zip tar.gz

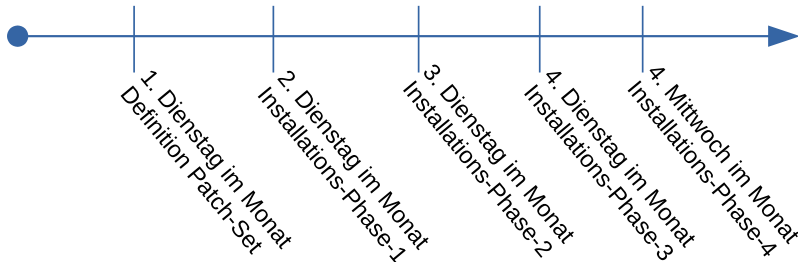
# RHEL-Patchmanagement with Ansible

Eine Ansible-Rolle plus einiger zusätzlicher Skripte, welche ein automatisiertes Patch-Management mit folgenden Funktionen ermöglicht:

- ▶ Zeitlich (Cron) gesteuerte Installation von Red Hat Security-Advisories (RHSA)
- ▶ Installation erfolgt als monatliches Patch-Set in vier zeitlich versetzten Phasen
- ▶ Zuordnung von Hosts zu Patch-Phasen erfolgt über Gruppen im Ansible Inventory
- ▶ Werden Pakete aktualisiert, erfolgt in jedem Fall ein Neustart des Hosts

# Use Case

- ▶ Monatliche Installation von RHSA gewährleistet ein Mindestmaß an Sicherheit
- ▶ Handelt der Systembetreiber nicht, erfolgt Zwangsinstallation
- ▶ Zusammenfassung der RHSA in Patch-Sets, zur Installation der gleichen RHSA in unterschiedlichen Phasen (Staging)





# Wo kriege ich das Ding?

<https://github.com/Tronde/ansible-role-rhel-patchmanagement>

[https://galaxy.ansible.com/tronde/ansible\\_role\\_rhel\\_patchmanagement](https://galaxy.ansible.com/tronde/ansible_role_rhel_patchmanagement)

 [Community Authors](#) > [Tronde](#) > [ansible\\_role\\_rhel\\_patchmanagement](#)



Tronde

## **ansible\_role\_rhel\_patchmanagement**

Patchmanagement for Red Hat Enterprise Linux.

✓ 4.4 / 5 Score  154 Downloads

 Login to Follow

 Issue Tracker

 GitHub Repo

Weitere Informationen in deutscher Sprache unter:

<https://www.my-it-brain.de/wordpress/ansible-patch-management-fuer-red-hat-systeme/>