Package Management, Version Control .. and much more

Udo Seidel

About Me … Just A Kickstart

- Linux and Open source: since 1996
- s+c: 2002-2005
- Many titles/positions since 2000 ...
- … now
  - Cyber Security
  - Customer Success Manager
- Massive digital Online traces ;-)

Today’s Agenda

- Self-made/unofficial package maintainer
- Journey and experience telling
  - Motivation
  - Getting there
  - Optional paths

How it all started

- Software
  - Unifi Network Controller
  - Signal messenger for Desktop
  - decK
- “Clean” package management
  - Must be a package
  - No mix of package technologies
- “Open source” to a very large extent

First Building Block

- All about package spec/control
- Version management
- Public and open
- Backup plan?? → Oh YES
  - Off-line
  - Online

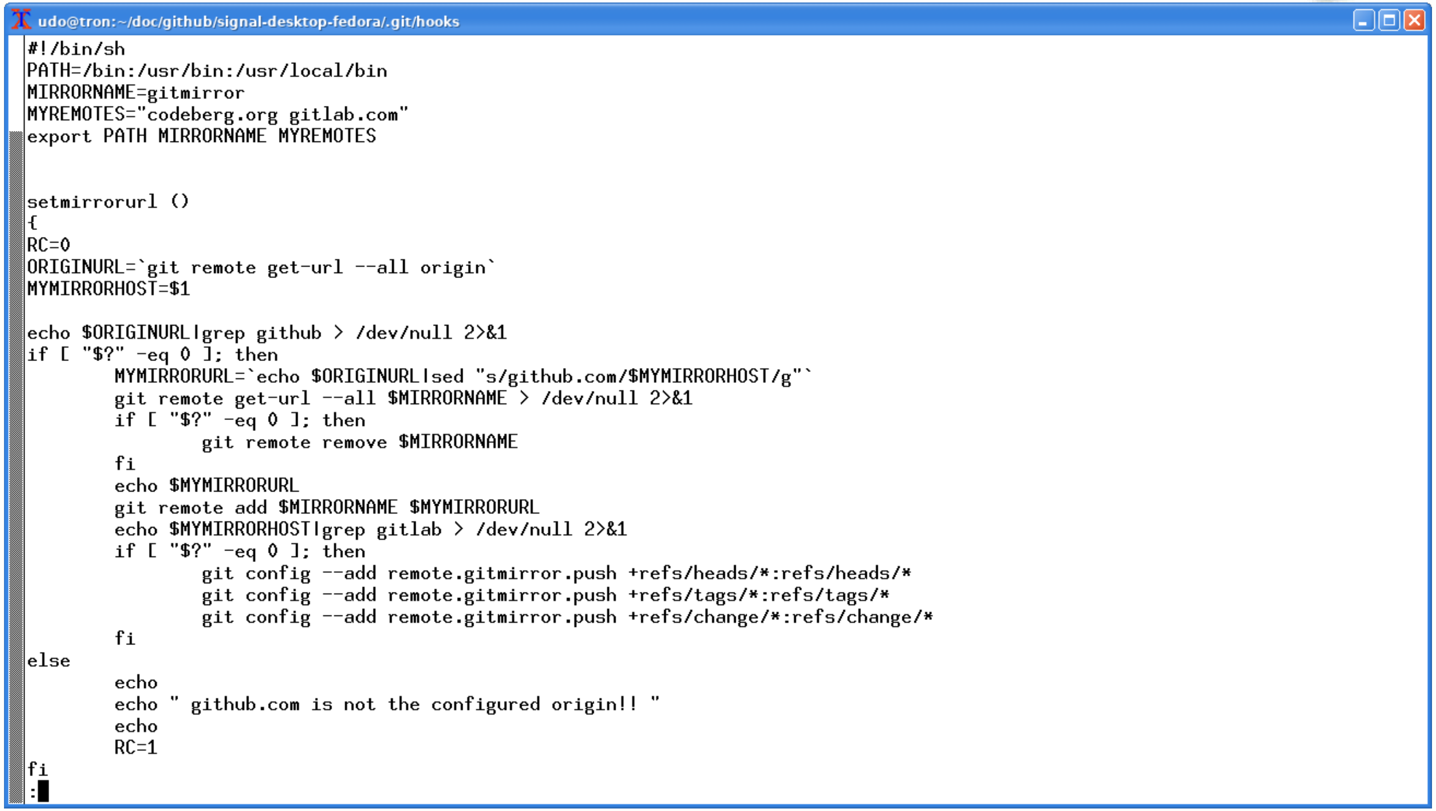
CONCLUDING THE FELLOWSHIP
FOLLOW-UP AFTER THE FELLOWSHIP IS OVER

Github, Codeberg .. and Gitlab

- “Copy” and conquer
- Decide on primary/master-ish
- Git hooks framework
- Devil is in the details

- Here: Github and *post-commit*

Example/Demo



```
#!/bin/sh
PATH=/bin:/usr/bin:/usr/local/bin
MIRRORNAME=gitmirror
MYREMOTES="codeberg.org gitlab.com"
export PATH MIRRORNAME MYREMOTES


setmirrorurl ()
{
RC=0
ORIGINURL=`git remote get-url --all origin`
MYMIRRORHOST=$1

echo $ORIGINURL|grep github > /dev/null 2>&1
if [ "$?" -eq 0 ]; then
        MYMIRRORURL=`echo $ORIGINURL|sed "s/github.com/$MYMIRRORHOST/g"`
        git remote get-url --all $MIRRORNAME > /dev/null 2>&1
        if [ "$?" -eq 0 ]; then
                git remote remove $MIRRORNAME
        fi
        echo $MYMIRRORURL
        git remote add $MIRRORNAME $MYMIRRORURL
        echo $MYMIRRORHOST|grep gitlab > /dev/null 2>&1
        if [ "$?" -eq 0 ]; then
                git config --add remote.gitmirror.push +refs/heads/*:refs/heads/*
                git config --add remote.gitmirror.push +refs/tags/*:refs/tags/*
                git config --add remote.gitmirror.push +refs/change/*:refs/change/*
        fi
else
        echo
        echo " github.com is not the configured origin!! "
        echo
        RC=1
fi
:
```

Checkpoint

- Done:
  - Version management as such
  - Backup plans
- Open
  - Build platform/framework

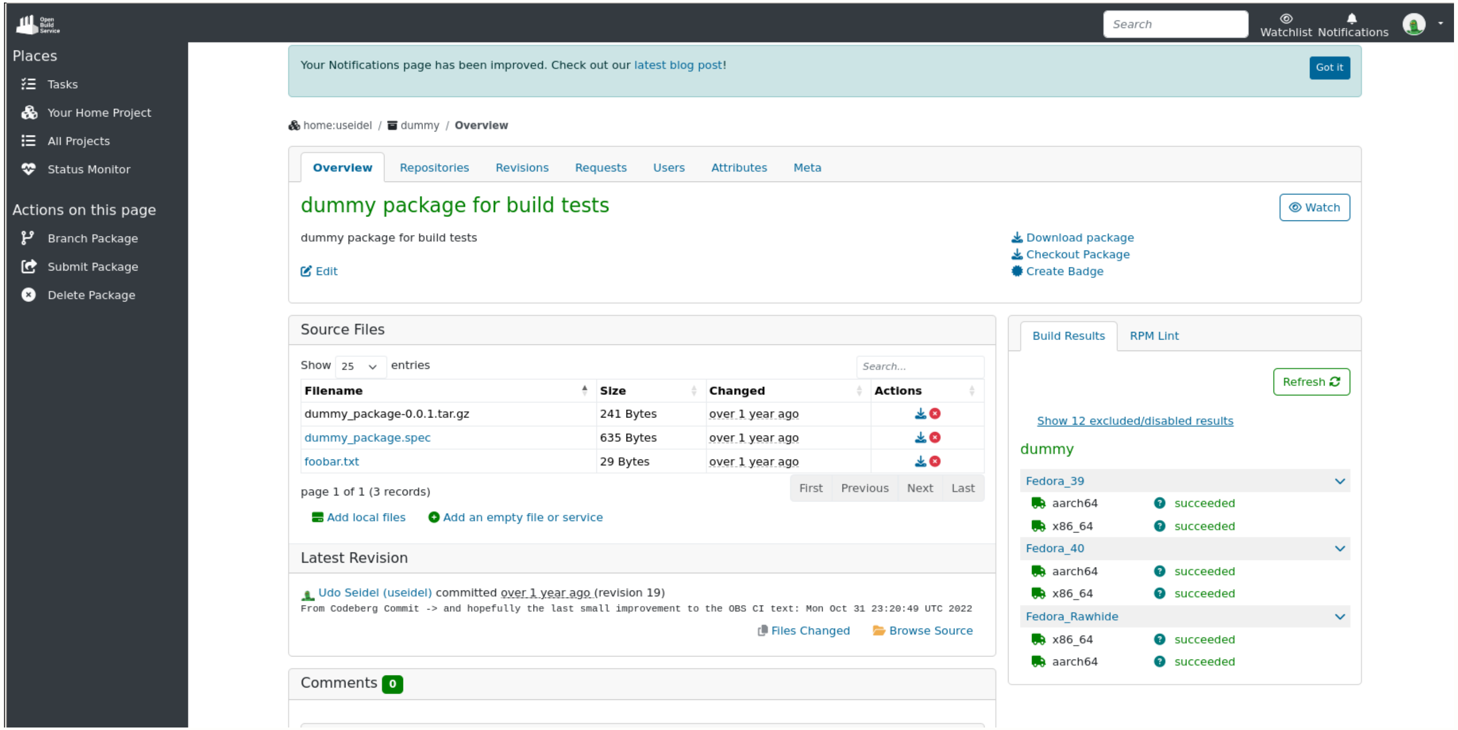
Second Building Block

- Build environment
- Public and open
- Backup plan?? → Yes … kind of …
  - Online
  - Off-line???

OBS and copr

- Open Build Service (OBS)
  - Many target/build distributions
  - Easy to integrate as package repository
  - No internet connections during build process
- copr (cool other package repo)
  - Red Hat centric
  - One repository config file per (build) package
  - Access to internet during build process available

Example/Demo - OBS



Search

Watchlist Notifications

Places

- Tasks
- Your Home Project
- All Projects
- Status Monitor

Actions on this page

- Branch Package
- Submit Package
- Delete Package

Your Notifications page has been improved. Check out our latest blog post!

Got it

home:useidel / dummy / Overview

Overview Repositories Revisions Requests Users Attributes Meta

dummy package for build tests

Watch

dummy package for build tests

Edit

Download package
Checkout Package
Create Badge

Source Files

Show 25 entries

Search...

Filename	Size	Changed	Actions
dummy_package-0.0.1.tar.gz	241 Bytes	over 1 year ago	
dummy_package.spec	635 Bytes	over 1 year ago	
foobar.txt	29 Bytes	over 1 year ago	

page 1 of 1 (3 records)

First Previous Next Last

Add local files Add an empty file or service

Latest Revision

Udo Seidel (useidel) committed over 1 year ago (revision 19)

```
From Codeberg Commit -> and hopefully the last small improvement to the OBS CI text: Mon Oct 31 23:20:49 UTC 2022
```

Files Changed Browse Source

Comments 0

Build Results RPM Lint

Refresh

Show 12 excluded/disabled results

dummy

Fedora_39

aarch64	succeeded
x86_64	succeeded

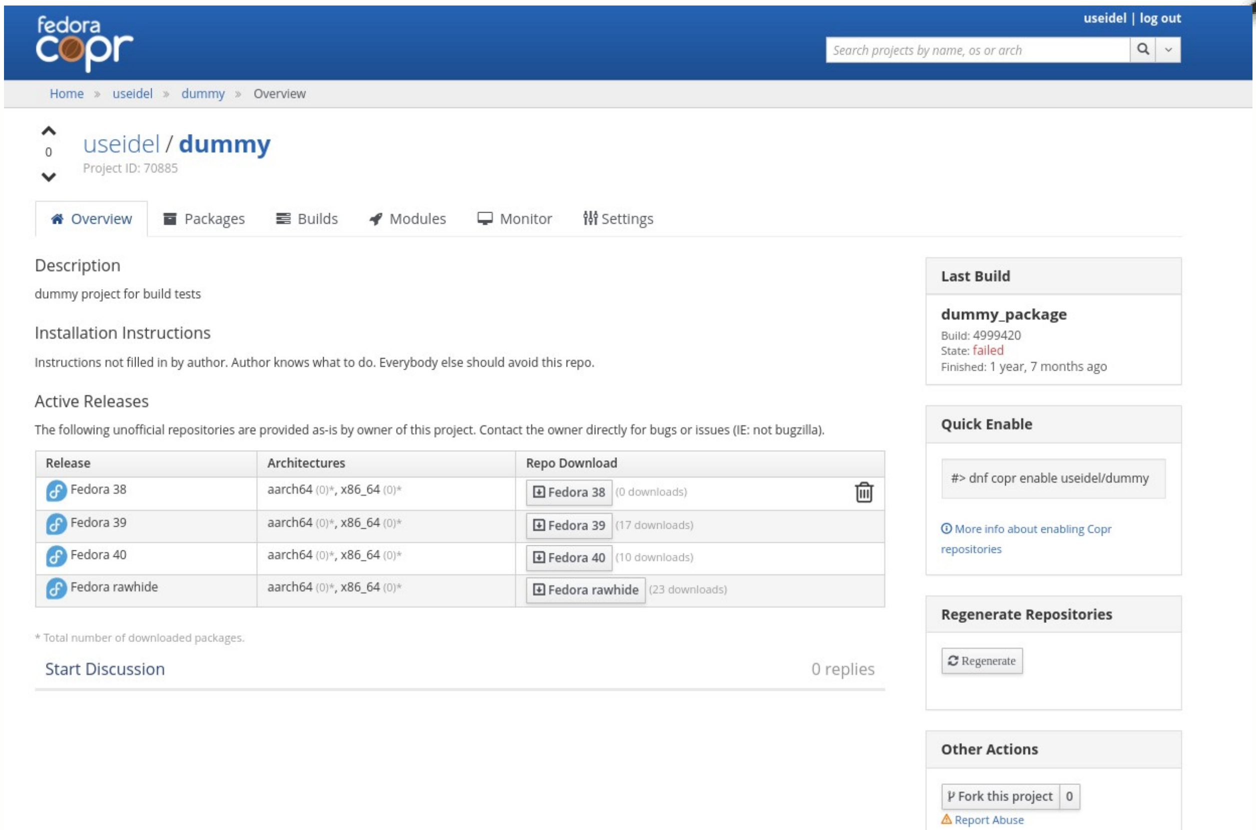
Fedora_40

aarch64	succeeded
x86_64	succeeded

Fedora_Rawhide

x86_64	succeeded
aarch64	succeeded

Example/Demo - copr



useidel / dummy

Project ID: 70885

Description

dummy project for build tests

Installation Instructions

Instructions not filled in by author. Author knows what to do. Everybody else should avoid this repo.

Active Releases

The following unofficial repositories are provided as-is by owner of this project. Contact the owner directly for bugs or issues (IE: not bugzilla).

Release	Architectures	Repo Download
Fedora 38	aarch64 (0)*, x86_64 (0)*	Fedora 38 (0 downloads)
Fedora 39	aarch64 (0)*, x86_64 (0)*	Fedora 39 (17 downloads)
Fedora 40	aarch64 (0)*, x86_64 (0)*	Fedora 40 (10 downloads)
Fedora rawhide	aarch64 (0)*, x86_64 (0)*	Fedora rawhide (23 downloads)

* Total number of downloaded packages.

Start Discussion 0 replies

Last Build

dummy_package

Build: 4999420

State: failed

Finished: 1 year, 7 months ago

Quick Enable

```
#> dnf copr enable useidel/dummy
```

More info about enabling Copr repositories

Regenerate Repositories

Other Actions

Checkpoint

- Done:
  - Version management
  - Build platform/framework
- Open
  - Connection/automation → “The CI”
    - Provided by already configured frameworks

      OR
    - Additional software/framework needed

Connecting The Bits

- Set-up of corresponding projects
  - Version control framework
  - Package build framework
- API for
  - Access
  - Authentication

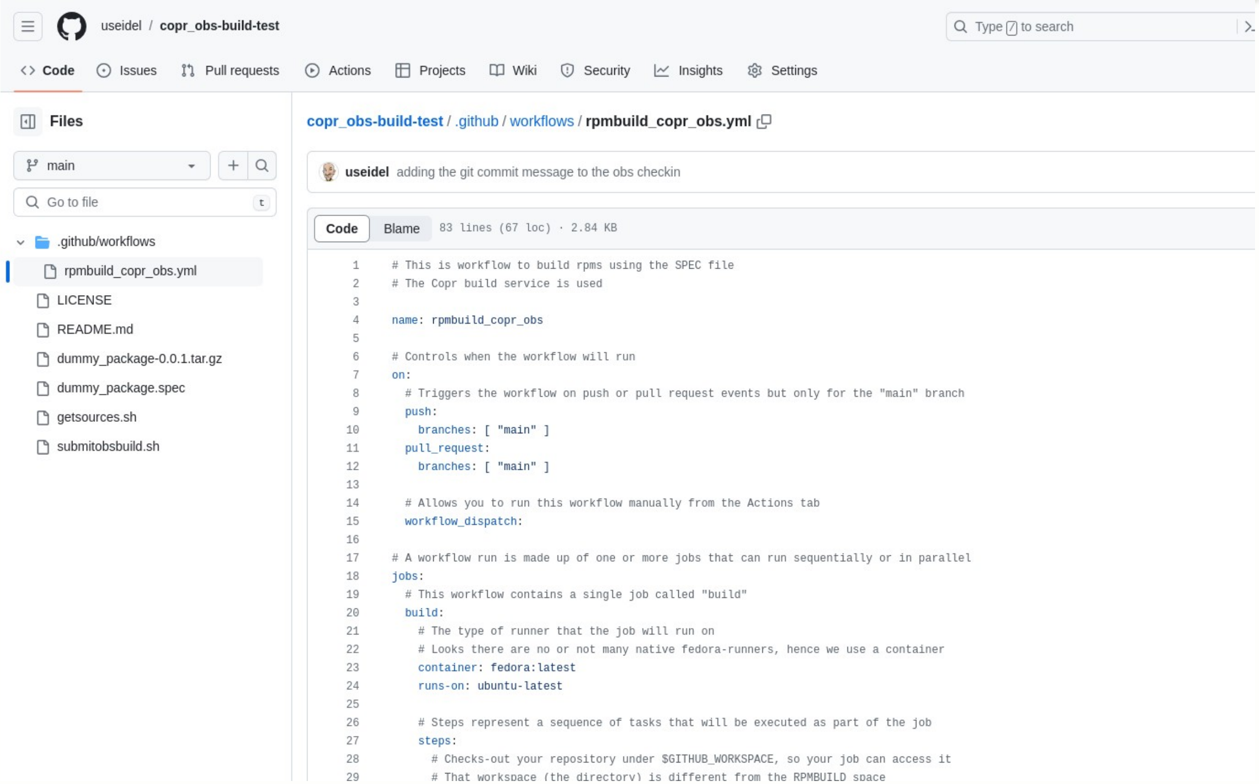
Option 1: Github Actions

- Very mature and it is integrated
- Carefully check available workers
- Again: details matter
- API -> Github repository secrets
- Easy to
  - Reuse
  - Extend via scripting

Example - Github Actions

- github.com/useidel/copr-build-test
- github.com/useidel/obs-build-test2
- github.com/useidel/copr_obs-build-test
- Ingredients
  - Workflow source file
  - SPEC file
  - Scripts for
    - Getting package source files
    - Submitting builds

Example - Github Actions



copr_obs-build-test / .github / workflows / rpmbuild_copr_obs.yml

useidel adding the git commit message to the obs checkin

83 lines (67 loc) · 2.84 KB

```yaml
# This is workflow to build rpms using the SPEC file
# The Copr build service is used

name: rpmbuild_copr_obs

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the "main" branch
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    # Looks there are no or not many native fedora-runners, hence we use a container
    container: fedora:latest
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
      # That workspace (the directory) is different from the RPMBUILD space
```

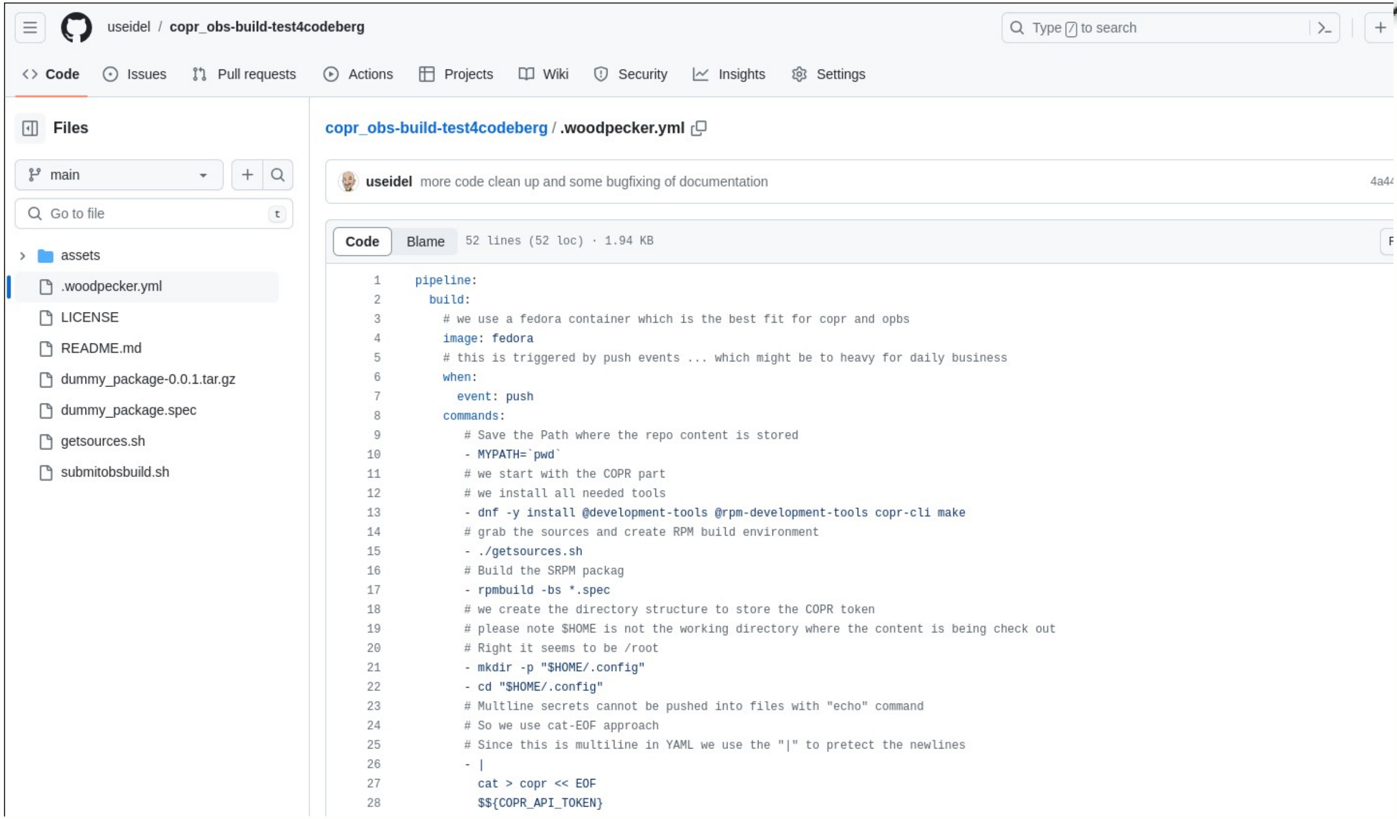
Option 2: Codeberg ...Woodpecker

- Less mature
  - Including documentation
  - Requires additional stuff → Woodpecker aka Codeberg-CI
- API access via secrets … but wait for the details
- Easy to
  - Reuse
  - Extend via scripting

Example – Codeberg ...Woodpecker

- github.com/useidel/copr_obs-build-test4codeberg
- Ingredients
  - Workflow source file
  - SPEC file
  - Shell scripts for
    - Getting package source files
    - Submitting builds

Example - Codeberg-CI/Woodpecker



copr_obs-build-test4codeberg / .woodpecker.yml

useidel more code clean up and some bugfixing of documentation 4a4

52 lines (52 loc) · 1.94 KB

```yaml
pipeline:
  build:
    # we use a fedora container which is the best fit for copr and opbs
    image: fedora
    # this is triggered by push events ... which might be to heavy for daily business
    when:
      event: push
    commands:
       # Save the Path where the repo content is stored
       - MYPATH=`pwd`
       # we start with the COPR part
       # we install all needed tools
       - dnf -y install @development-tools @rpm-development-tools copr-cli make
       # grab the sources and create RPM build environment
       - ./getsources.sh
       # Build the SRPM packag
       - rpmbuild -bs *.spec
       # we create the directory structure to store the COPR token
       # please note $HOME is not the working directory where the content is being check out
       # Right it seems to be /root
       - mkdir -p "$HOME/.config"
       - cd "$HOME/.config"
       # Multline secrets cannot be pushed into files with "echo" command
       # So we use cat-EOF approach
       # Since this is multiline in YAML we use the "|" to pretect the newlines
       - |
         cat > copr << EOF
         $${COPR_API_TOKEN}
```

What Is Missing/Outstanding/Next?

- Using *release* – not *push*
- Link to original software repositories
- Use Gitlab-CI as well

When time permits … :-D

Additional Information

- Github Actions
- OBS
- copr
- Woodpecker-CI
- useidel@github

- *Pictures by different artists from Unsplash.com*