# Fedora dist-git Packit onboarding

| | |
|---|---|
| **Date:** | 2024-05-05 |
| **modified:** | 2024-05-05 |
| **tags:** | Packit, Fedora, Packages, Packaging, dist-git |
| **description:** | Onboard a Fedora dist-git repository to Packit |
| **category:** | Linux |
| **slug:** | fedora-dist-git-packit-onboarding |
| **Author:** | Dominik Wombacher |
| **lang:** | en |
| **transid:** | fedora-dist-git-packit-onboarding |
| **Status:** | published |

Packit, oh my god, that's a tool and service that gave me a pretty hard time to understand how it works. Not necessarily because it's a complicated tool. But it expects a good portion of background knowledge how things work in Fedora and the specific wording. If you are new and make your first steps, it becomes very challenging and frustrating to get started.

So first, what's Packit? Very simplified: Packit get triggered when a new version of a software is released. It can then trigger package builds. Or create PRs in Fedora packages to update them. The main goal is to reduce the work maintainer have to keep Fedora packages up to date.

My first false assumption was that Packit always requires to have a configuration in the repository of the upstream project. Another mistake was that Packit will help me to test and build packages during development. You can probably imagine that those two misunderstandings lead to a good portion of confusion and wasted time.

So how to benefit from Packit if you don't control the upstream project and they are not interested to onboard their repository to the Packit Service? Fedora runs Release Monitoring a service that checks for new releases in configured projects on a regular basis. If there is one, it will publish a message into Fedora Messaging. A RabbitMQ based messaging service that allows other services to react on events. Packit will pick up such an event and check if there is a packit configuration file in the dist-git package repo. If that's the case, Packit will execute the configured actions.

So for example, Packit creates a PR in the package repo. The PR bumps the package version and adds a changelog entry. Package maintainer can add additional changes to the PR if necessary and upload the new source file into the lookaside cache.

Then your job boils down to merge the PR into the rawhide branch. If correctly configured, Packit will pick up this event, remember everything in the Fedora world becomes an event in the Fedora Messaging bus, and trigger the build. If that build is successful the update is triggered and the package becomes available in rawhide. Next, if you have other branches you want to make the package available, you merge it and then again Packit picks up the event, starts a build and the update.

This is a massive time saver already, even if you still have a couple of steps to do. That's the workflow I prefer, but you can of course adjust it and let Packit do even more in an automated way. The decision is up to you :)

What I explained here is just one part of what Packit can do. I encourage you to explore the project and documentation to dive deeper.

For me it was a massive "AHA" moment when I had my first Packit config working and saw the magic happen :)

Following the `.packit.yaml` file I prepared for the first Fedora package I'm going to release:

```yaml
# See the documentation for more information:
# https://packit.dev/docs/configuration/

upstream_project_url: https://github.com/aws/aws-ec2-instance-connect-config
upstream_package_name: aws-ec2-instance-connect-config
downstream_package_name: ec2-instance-connect

jobs:
- job: pull_from_upstream
    trigger: release
    # Keeping dist-git branches non-divergent
    # Requirs manual local merge from rawhide to stable release branches
    # https://packit.dev/docs/fedora-releases-guide#keeping-dist-git-branches-non-divergent
    dist_git_branches:
    - fedora-rawhide

- job: koji_build
    trigger: commit
    allowed_pr_authors:
    - packit
    - all_admins
    - all_committers
    - '@cloud-sig' # string with @ needs quotes to be valid yaml
    allowed_committers:
    - all_admins
    - all_committers
    - '@cloud-sig' # string with @ needs quotes to be valid yaml
    dist_git_branches:
    - fedora-all
    - epel-all

- job: bodhi_update
    trigger: commit
    allowed_builders:
    - packit
    - all_admins
    - all_committers
    - '@cloud-sig' # string with @ needs quotes to be valid yaml
    dist_git_branches:
    - fedora-branched # rawhide updates are created automatically
    - epel-all
```