



# State of RPM packaging

---

Odilon Sousa • 08.11.2022



What?  
Why?  
How?

## What we build?

<https://yum.theforeman.org/pulpcore/>

## Why?

- Katello consumes pulpcore from RPMs
- AAP/Galaxy\_ng also consumes pulpcore from RPMs

## How?

Katello Koji + Jenkins + Obal

<https://ci.theforeman.org/>

<http://koji.katello.org/>

<https://github.com/theforeman/obal>

---

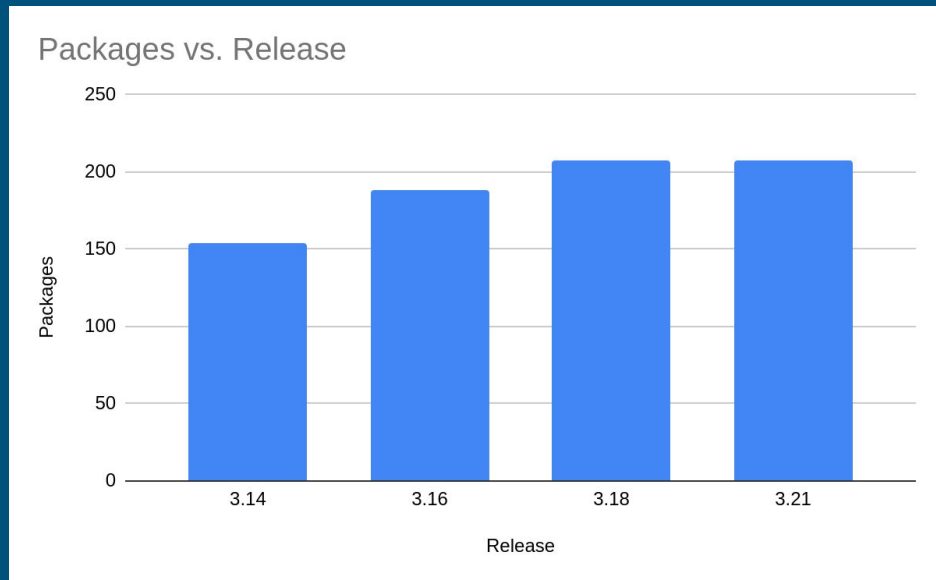
# Number of packages

**Packages for EL8**

**3.14 to 3.21**

**From 154 to 207**

**3 Python releases**



# How we deal with new dependencies

---

## How we catch new dependencies?

- During branching
- During packages updates
- While trying to run the release pipeline

## Guideline for adding new dependencies

We try to stay as close as possible to Fedora Python packaging guidelines

We don't use EPEL for soft dependencies, we build all necessary dependencies.

# What about PEP-517

---

## PEP-517

- Introduced a better way to build python modules.
- No more setup.py
- Our build system is not ready this new technological era, we still use setup.py

<https://peps.python.org/pep-0517/>

## No setup.py?

- We need to inject a minimal setup.py to build on EL7/EL8, EL9 is smarter.
- We inject at our Spec file.

<https://github.com/theforeman/pulpcore-packaging/blob/rpm/3.21/packages/python-importlib-metadata/python-importlib-metadata.spec#L42>

The challenges when you need  
to maintain 4 releases at the  
same time, with different  
python versions

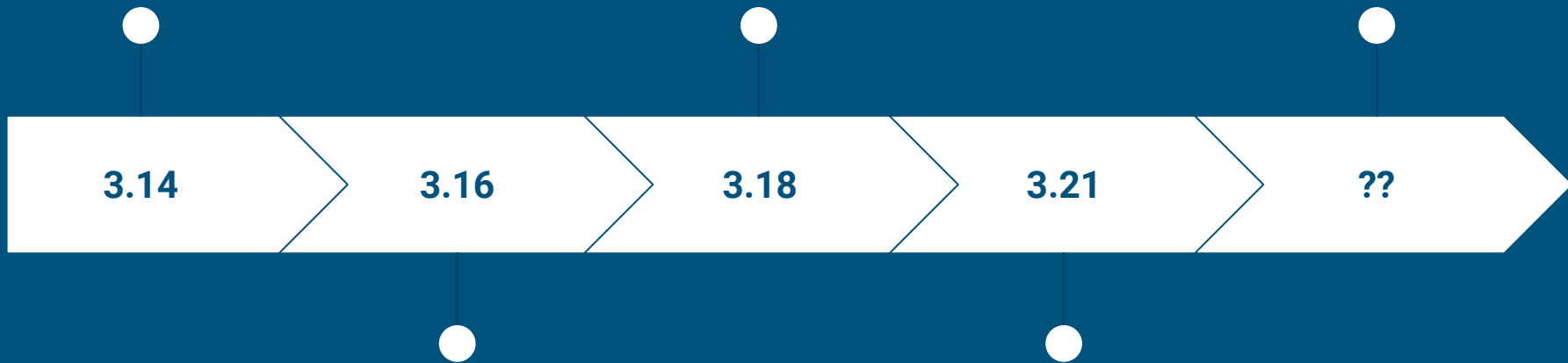
Python 3.6

EL7 and EL8 RPMs

Python 3.9

EL7 SCL(3.8) / EL 8  
Module / EL9

Who knows?



Python 3.8

EL7 SCL / EL8 Module

Python 3.9

EL8 Module / EL9

# Mass Rebuild

---

Everytime we bump the python version, we need to mass rebuild.

To run on EL9 we had to bump to python 3.9, from 3.8.

Pulp was one of the first projects to fully support EL9.



# How we test installations?

---

## What we test?

Every package update runs a test, a scratch build and it tries to mock installs to see if any dependency is failing.

## How we test

Before any package get released, we run pulp-installer to make sure that pulp installs. This allow us to test a big matrix of releases from Pulp and Python.

# What we can improve?

1. Better way to identify new dependencies
  2. Branching is slow and is oriented by failing
  3. A streamlined approach from Pypi to RPM
-

# How can I package my pulp component?

Reach us on #pulp-dev or create  
one issue at

<https://github.com/theforeman/pulpcore-packaging>

And we will guide you through the  
process

---

# Questions?



That's all

