Journey of Automation - Github, Galaxy, Fedora

Pavel Cahyna, Sergei Petrosian

Automation is more important than ever before in software project management

Being able to automate the low level, labor intensive parts of project management is critical. There are many tools in the Fedora and Github ecosystems that facilitate project management, such as GitHub Actions, Packit, and more. Learn how the Linux System Roles team leveraged these tools to perform:

- Automated Ansible role release and publish to Ansible Galaxy
- Automated Ansible collection build, publish and release to Galaxy
- Automated Fedora RPM build and publish with Packit

Automated GitHub Releases

Our Use of Conventional Commits

Semantic Version

PR titles follow the conventional commits format, i.e include a type of changes in it in the form `<type><!>: PR Title`.
The type can be one of:

- `feat` - results in the `MINOR` version bump
- `fix`, `ci`, `test`, etc. - results in the `PATCH` version bump

Exclamation mark **!** after the type indicates the major update and results in the `MAJOR` version bump.

Processing Conventional PR Titles

We have a GitHub action that we run when we want to create a new release. This action does the following:

1. Figures out the semantic version to use for the new tag
2. Builds changelog based on PR types with `New Features`, `Bug Fixes`, and `Other Changes` sections
3. Pushes a new repository version and tag into GitHub

Releasing Individual Roles

Each repository has a GitHub action that watches for new releases, and when one occurs, publishes the role into Ansible Galaxy.

Publishing Collections

We have a GitHub action that runs daily and finds out if any of the role repositories has a new tag published.
If yes, the action converts the roles to a collection format and publishes the resulting collection to Galaxy.

Automated RPM Release with Packit