

Journey of Automation - Github, Galaxy, Fedora

Pavel Cahyna, Sergei Petrosian

Automation is more important than ever before in software project management

- 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

1. GitHub Action that we run manually when we need a new release
2. GitHub Action that runs daily and releases new content if any of the repositories have updates

How We Use Conventional Commits

Format: `<type><!>: PR Title`

- **!** - **MAJOR** bump
- `feat` - **MINOR** bump
- `fix`, `ci`, `test`, etc. - **PATCH** bump

feat: Support custom data and logs storage paths

Processing Conventional PR Titles

1. Figure out the semantic version to use for the new tag
2. Build changelog based on PR types
3. `feat:` -> *New Features*
4. `fix:` -> **Bug Fixes*
5. `else` -> *Other Changes*
6. 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 cron-like 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