

GitOps limitations

CNF LCM Automation

Our goal is to automate CNF LCM

- Installation
- Termination
- Upgrade

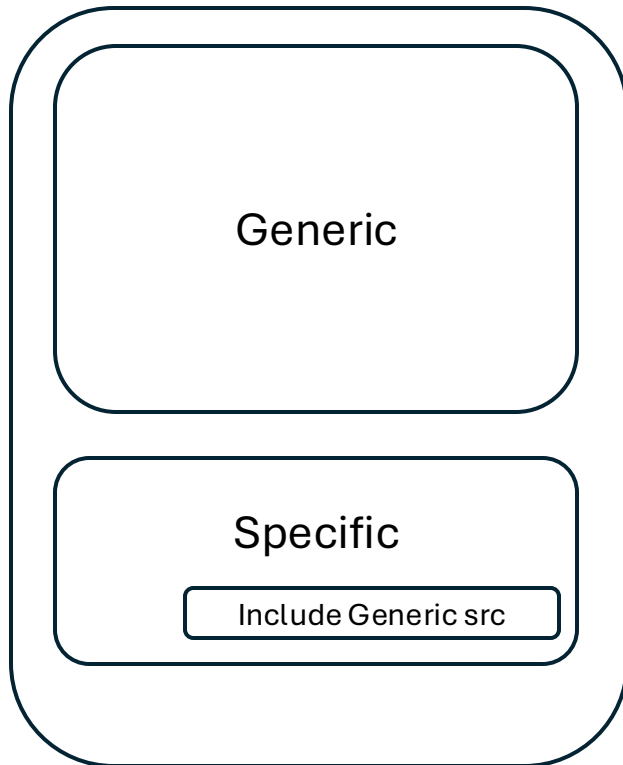
CNF Automation

- Vendors are very creative and more or less Cloud native
 - 1 top helm charts
 - Multiple Helm Charts with static sequence order
 - Vendor operators
 - Vendor "orchestrator"
- Not sure there is 1 structure/method to tackle LCM due to the diversity of the CNF implementations
- We want 1 single source of truth for that we imagined the generic (Helm releases in 1 place)/specific (values for the different target clusters) repositories approach, but it triggers also issues
- Termination procedures are usually missing and rarely complete (remaining resources) - commenting resources in head kusto does not mean that they will be cleanup
- The main difficulty deals with upgrades as for upgrade, the procedure may differ from the installation
- The less cloud native, the more dependencies

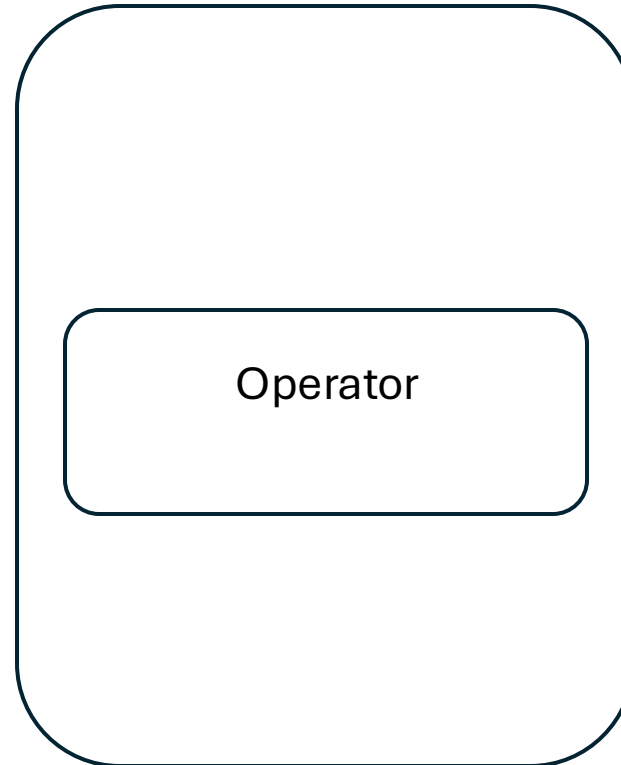
1 major stake for upgrade

- Be able to invalidate all the automation objects at once – restart automation as it would be an installation ~reset the reconciliation loops
- In fact if we do not invalidate them we need to manage the sequences by suspending some resources before the other
 - It is risky
 - It breaks the full automation path

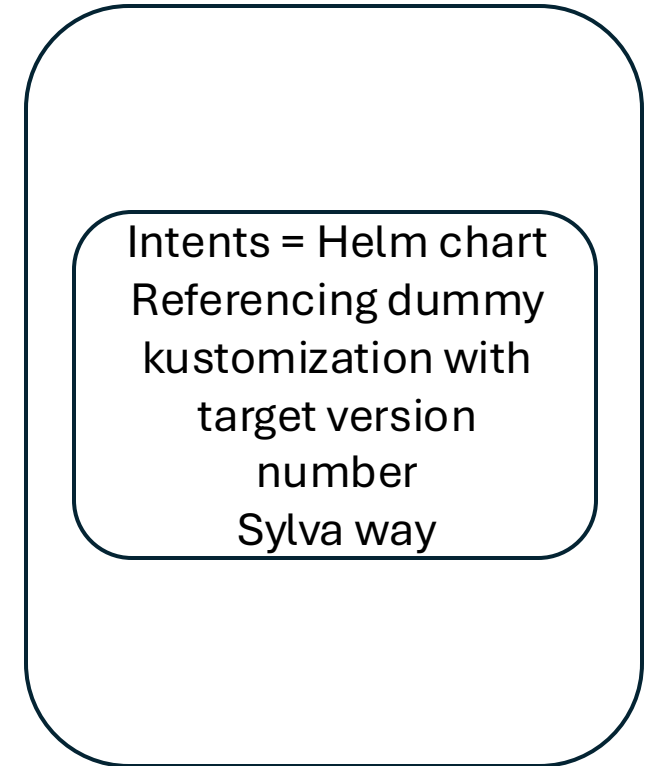
Focus on 3 upgrade scenarios



"easy" to understand for operations – everything is written
Inclusion to link and sync
Generic and Specific sources to avoid starting upgrading with part of old intents



Dev to be done...1 per CNF...
Hard to be generic
Need to understand each vendor logic



Works well for complex Sylva upgrade
Hard to understand for operations (no more intent just the helm chart that will produce the manifests + values/secrets)

Dev of a Helm chart per function

Testing an alternative approach

- Custom development in Go of a tool watching GitRepositories updates and setting involved Kustomizations and HelmReleases status to False
- It forces Flux to reconcile Kustomizations respectful of dependencies order
- Still under test on complex CNF, to validate the approach
- If successful, should be generic and not depend on the Flux intents architecture (mono-repo, specific / generic, repository composition using includes, etc.)