EdgeX Semver Explore

Thursday April 8, 2021
EdgeX Semver Explore: Background

History
https://github.com/edgexfoundry/git-semver
- Established March 2019
- Written in Golang
- CLI tool that manages semver versions across multiple edgex repositories

Main motivation driving explore
- Tech debt
- Aligning to other more widely used “industry” tools
- Potential feature gap with LTS releases

Objective
- Need decision to move forward with switching tools that maintain software versioning and release management. Maybe need to present to a wider audience to make decision.
EdgeX Semver
Explore: Status Quo

Branching strategy
• All code is merged into master
• Master is treated as the pre-release branch.
• Only branch when necessary (typically for dot releases)
• Keeps branching to a minimum...simplifies process

Workflow
• When features are merged to master, semver pre-release version is bumped
• Tag on every commit to master (potentially useful for go mods)
• Named release process is managed separately in cd-management/release branch
• Release Czar can create release in central repo
• cd-management/release would get difficult managing multi-branch releases (i.e. LTS releases)
EdgeX Semver
Explore: Changes

Branching strategy
• New tooling will bump semver major/minor/patch when code is merged into master and master may not be able to be considered master as a “pre-release” branch
• Move from current strategy where master is treated as a development pre-release branch
• Follows more closely to “Git Flow” strategy
• Feature changes to master branch would constitute as a major, minor “Named Release” (Hanoi, Ireland, Jakarta, Etc).
• Critical bugfixes would be branched off master and merged into master causing a patch release. A named release branch could also be created if more testing is required
EdgeX Semver Explore: Changes

Workflow
• Conventional commits dictate semver versioning rules (can be customized)
• "dev" (or similar) branch becomes main development branch during release development
• "dev" branch treated as pre-release branch.
• Only create tags when conventional commit rules dictate (can be customized per repo)
• Git tags become system of record for version tracking
EdgeX Semver
Explore: Changes

Pipeline Changes
• Changes would be required to the Jenkins pipelines to incorporate new tooling to trigger the releases
• Release Czar no longer needs to maintain release yaml files in cd-management
• Release Czar could coordinate with WG chairs to manage release PR’s
EdgeX Semver Explore: Branching

Develop and Master branches

Develop and Feature branches
EdgeX Semver Explore: Sample Workflow

Simplified use case

[m] e66ea38 feat: initial commit - Ernesto Ojeda

[m] 946d6b4 release: init to v1.0.0 - edgex-jenkins
EdgeX Semver Explore: Sample Workflow

Development Branch

[main] f26ea4f feat: initial commit - Ernesto Ojeda

v1.0.0

[main] 46d3b09 release: init to v1.0.0 - edgex-jenkins

[dev] 1349654 init dev branch - Ernesto Ojeda
EdgeX Semver Explore: Sample Workflow

New Feature Added, semver bumped

[main] 26b2bc1 feat: initial commit - Ernesto Ojeda

[main] 32647b2 release: init to v1.0.0 - edgex-jenkins

[feature/new-service] 72e356c feat(core): new core service - Ernesto Ojeda

[dev] 9dde29b Merge pull request #1 from ... - Ernesto Ojeda
EdgeX Semver Explore: Sample Workflow

Another Feature Added, docs... no semver bump

[main] 059b4d0 feat: initial commit - Ernesto Ojeda

v1.0.0

[main] 1640d92 release: init to v1.0.0 - edgex-jenkins

[feature/new-service] 4f67d2b feat(core): new core service - Ernesto Ojeda

v1.1.0-dev.1

[dev] ee08b16 Merge pull request #1 from ... - Ernesto Ojeda

[feature/documentation] 8ce9585 docs(core): update service docs - Emilio Reyes

[dev] 5109346 Merge pull request #2 from ... - Emilio Reyes
EdgeX Semver Explore: Sample Workflow

Another Feature Added, fix... semver bump

[main] b9fcb27 feat: initial commit - Ernesto Ojeda

v1.0.0
[main] 9624001 release: init to v1.0.0 - edgex-jenkins

[feature/new-service] b193e13 feat(core): new core service - Ernesto Ojeda

v1.1.0-dev.1
[dev] f211e1a Merge pull request #1 from ... - Ernesto Ojeda

[feature/documentation] 5529b03 docs(core): update service docs - Emilio Reyes

[dev] 155f5e77 Merge pull request #2 from ... - Emilio Reyes

[bugfix/buffer-overflow] e999b74 fix(messaging): fix buffer overflow in messaging - Lenny Goodell

v1.1.0-dev.2
[bugfix/buffer-overflow] 07891e fix(core): fix buffer overflow in core service - Lenny Goodell

[dev] 76a36b2 Merge pull request #3 from ... - Lenny Goodell
EdgeX Semver Explore: Sample Workflow

Finally, ready for full release
Where do we go from here?

- Are there a more commonly used tools?
- Are we following the right paradigm with semver? i.e. separate branch holding semver version, pre-release bumping on every merge to master vs only bump when conventional commit matches specific noun
- What do other open-source projects do for releases?
- Can we decentralize releases down to the repo level?
- Ultimately can we remove people from the equation?
EdgeX Semver
Explore: Path Forward

Semver/Release Automation Ecosystem
• Automatic semver versioning/bumping based on conventional commits
• Auto-generated CHANGELOG via conventional commits
• Auto-generated Release Notes with conventional commits
• Plugin based architecture allowing for more extensibility
• Decentralized, each repo is released on its own
• Pre-release version would still be supported

Is this the right approach?
• Would require a fairly heavy lift to switch
• Probably will require process changes with branching
• Aligning to well maintained industry tools will bring down our tech debt