December 9th, 2020 - TSC

Participants

Susan Bowen (HPE), Jonas Arndt (HPE), Louis (LF), Alex Vul (Intel), Piotr Zedlewski (Inte), Bob Monkman (Intel),

Agenda

- Continue discussions on our Release Management Process
- Potential move under an umbrella organization
- Align on a better time slot for the TSC Proposal Meeting
- Continue discussion on Topic Proposal(s) for LFN Virtual Dev & Testing Forum, Feb 1-4, 2021

Notes

Release Management discussion

- Returned to the question of whether to have maintenance releases.
- Decisions relating to the frequency of releases and whether to have major and maintenance or a different strategy:
    - How far back do we support?
    - How to name/number the releases.
    - Branching strategy.
    - Snapshot releases.

- Supporting previous versions
    - Proposal: Don't create maintenance releases and backport them.
    - For other projects that do maintenance releases, they are tied to one specific major release.

- How to name/number the releases
    - Could use numbers, but make them 1.0 for a major release, 1.01 for small change, then the next big release becomes 2.0. (new feature or improvement or evolution)
    - Could use code names for releases, such as *"city"* Then if there is a maintenance release it is described as "maintenance release for (name of) *city* release."
    - Could use the last two digits of the year and two digits for the month, so a release in January of 2021 would be called "21.01" A point release a month later would be called "21.01.1"
    - Piotr Zedlewski also described
    - the SPDK project strategy, which specifies LTS (Long Term Support) releases. Its numbering is "2.1.0 LTS release". Once a year with a maintenance release. Links:

      From Piotr Zedlewski (Intel) to Everyone: 09:22 AM

      https://spdk.io/release/2020/06/01/20.01.2_20.04.1_release/SPDK

      Storage Performance Development Kit

      spdk.io

      https://spdk.io/

- Branching Strategy
    - The Git branching model: https://nvie.com/posts/a-successful-git-branching-model/, Includes Main branch, Dev Branch, Feature Branches. (see chart)
    - Different branches, different streams by function (telemetry, eventing as examples). When one of the new branches becomes stable, it goes into the release train.
    - The SPDK project has only the main and release branches. Concern there is developing the new and experimental and doing them directly in the release branch.
    - Experimental work (new features/functions) should never break the existing release (main branch).
    - Desire for ODIM to get to a loosely coupled Get to open stack model of self-certification (Alex) need to get to that. Rigorous (API tests passed – this will fall onto the dev team).
    - Common core with supporting services ODIM relies on. Loosely coupled and pluggable is good – better off the project will be. Rafal and Alex want to talk about how to make it modular and loosely coupled for future expansion – they would have people dedicated to doing that work.
    - You should be able to clone from main branch with confidence

- Snapshot releases
    - When a new feature/function branch becomes stable, it goes into the release train.
    - There is ongoing CI CD and no maintenance releases. Releases that are a snapshot.

9:40 am TOPIC: TSC PROPOSAL MEETING scheduling

- Tuesdays from Pacific 6 am to 7 am

- 21st and 28th
- Meet first week of 2021 (release Jan 31st!)

9:50 TOPIC: Move under umbrella

- CNCF - Sandbox project? (Would ODIM ever get out of that?)
- Martin, when discussion with CNCF it may never become a qualified project. They didn't have resources to review sandbox projects and make ODIM a regular (qualified) project.
- Alex, we need to do the homework. If CNCF develops an alternative bare-metal toolkit.
- Ran out of time

9:58 LFN and LFE? (Louis said) wherever home for ODIM

- Louis says good to get exposure in LFN Virtual Dev & Testing Forum, Feb 1-4, 2021.
- Is the request to have an ODIM session?

Meeting Recording

Chat File