Apache CloudStack release cadence and procedures

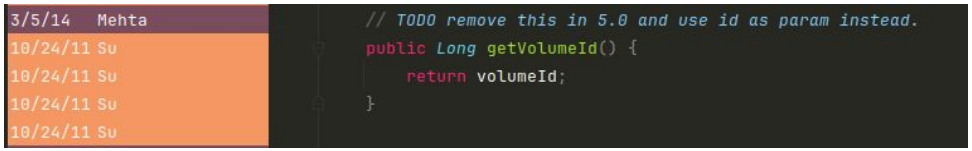
Daniel Augusto Veronezi Salvador
<gutoveronezi@apache.org>

Agenda

- Introduction
- Current cadence and processes
- Why we need to change
- What we need to improve and change
- Alternative discussed so far
- Questions

Introduction

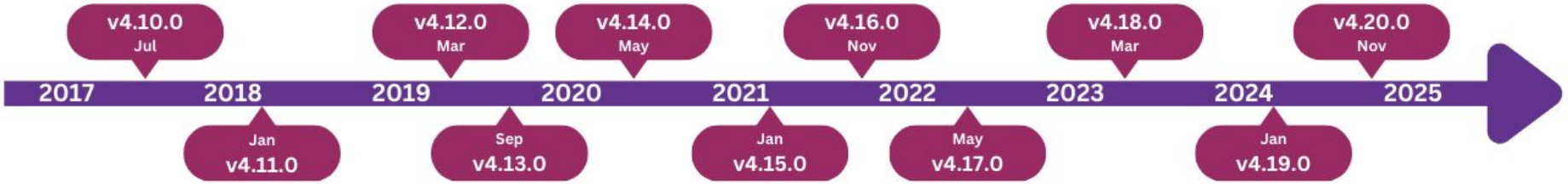
- We claim to follow the **Semantic Versioning 2.0.0**
- Version 4.y.z.n since 2014
  - Current version is 4.19.1.1
- The next major is an old wish
  - Thread **Why CloudStack 5** in dev ML, started in January 2019
  - Thread **[PROPOSAL] version naming : drop the 4** in dev ML, started in January 2024
  - Discussion **Define a release schedule for the project (#8970)** in GitHub, started in April 2024



src/main/java/org/apache/cloudstack/api/command/user/volume/MigrateVolumeCmd.java

Current cadence and processes

- We **try** to release 2 major versions per year
  - We have a release every 9 months (on average)



Current cadence and processes

- Every version becomes an **LTS**
  - We do not release **Regular** versions as we need*
- Each LTS is supported for 18 months
- We do not have a well-defined process for deprecating things
  - Generally we add configurations to change the processes (and keep the previous behavior as default)

*https://cwiki.apache.org/confluence/display/CLOUDSTACK/LTS

Why we need to change

- The community is always overloaded with several LTS releases
  - There were cases that we had almost 4 LTSs at the same time
  - Users/operators can get confused without knowing what to expect from a “not major release”
- Some incompatible changes are being introduced without proper planning and consensus
- In the past years, LTSs have been a bit unstable
- We only increase the complexity of the platform with configurations and workarounds
- We are not able to accurately predict the next steps

Why we need to change

- Our current code-base has non-standard technologies and frameworks, which creates entry barriers for new contributors and project further development
  - We have a home-made JPA framework
  - We do not follow RESTful in the essence (e.g.: GET methods change data)

- To introduce well established market frameworks, we need to introduce incompatibilities and break some deprecated methods and processes that we are supporting and using

What we need to improve and change

- Follow the **Semantic Versioning 2.0.0** de facto
- Define a solid and consistent release cycle
  - Resume the thread **[Discussion] Release Cycle** started in the dev ML, in August 2021
  - Resume the discussion **Define a release schedule for the project (#8970)** started in GitHub, in April 2024
- Define processes for introducing disruptive changes
  - We can look at how other projects do that, and adapt standard and solid methods and processes to our needs
- Work close together on common goals more often

Alternative discussed so far

- Release an **LTS** in a specific month every 2 years
  - In the meanwhile, we can release as many **Regular/minor** versions we need
- Use a “rolling upgrade” approach for disruptive changes
  - Introduce the new/redesigned features disabled by default in an LTS, allowing users to validate them
  - After validated, enable the new/redesigned features by default and remove the support for the old ones
- Automate everything we can (for that, we need well-defined protocols)

Questions?

gutoveronezi@apache.org