ONAP
OPEN NETWORK AUTOMATION PLATFORM

Options to convert architecture wikis to proper documentation

Andrea Visnyei, Gergely Csatari

21.10.2021

Approaches - 1) Convert once, work in rst

Pros:

- Converting from confluence to rst needs quite a lot of manual work. In this approach, this has to be done only once.
- Architecture docs are easily accessible for all releases in readthedocs

Cons:

- Developers need to learn documenting in rst
- Change in developer habits is not easy

Suggested change in the release milestones:

- M4: Confirm that architecture docs are in master and rtd

Approaches - 2) Convert in every release, work in Confluence

Pros:

- No change from the developers' point of view

Cons:

- It is not clear when the conversion should happen
- There is a “big bang” conversion in every release

Suggested change in the release milestones:

- M4: All architecture docs are converted and published to master and rtd

Methods – 1) c2m, pandoc, sphinx

- Uses a freeware java tool to convert the Confluence content to md
- .md is converted to rst with pandoc
- .rst is compiled with Sphinx

- Confluence plugins, like diagrams made with Giffy are not included
- Needs manual tweaking with tables and images
- Tables are not human-readable

Methods – 2) Confluence md plugin, pandoc, sphinx

- Uses a proprietary Confluence plugin to download the content in md
- .md is converted to rst with pandoc
- .rst is compiled with Sphinx

- .md download is not available in the API
- Confluence plugins, like diagrams made with Giffy are not included
- Needs manual tweaking with tables and images
- Tables are not human-readable

To do

- Problem with tables -> links and bulletpoints within tables break MD/RST format tables
- Search+replace or even a script that can be run on output could solve this (cells are delineated with \\ and rows with \ in output, + means bulletpoints, which could be problematic, since it's used in RST to show the start of a new row/column) ->script better, since some additional formatting is needed after removing the extra characters
- Question: what should be transferred? Only content under Documenting ONAP Architecture (e.g. currently maintained release docs, ONAP Architecture Principles)?
- JPG images at least have placeholders (e.g. |onap_nbi_dublin.jpg|), so adding them back in shouldn't be a problem, but flows created with PlantUML or draw.io are not marked in the output -> add a JPG version to Confluence and maybe an alt text or other placeholder to the original images?