Other options to generate documentation that we tested

Doxygen plugin -> HTML published plugin (html)/ LaTeX (pdf)

Description: This was the first discovered method

- html: using Doxygen plugin + HTML publisher It involves some customization at doxygen level + custom html header/footer
- pdf: it generates a .pdf using latex
- Input files: .md , .rst
- Output: .html & .pdf
- Pros:
    - standard tools: doxygen, html publisher, LaTeX suite
    - doxygen plugin available in Jenkins, you just need to install it; html publisher plugin available in Jenkins, you just need to install it
    - destination files are generated fast
    - standard reStructuredText or Markdown
- Cons:
    - takes some time to customize the output in matters of template, requires custom html header/footer
    - latex suite is quite substantial in amount of packages and consumed space (around 1.2 GB)
- Tested: roughly, functional tests only

Maven & clouddocs-maven-plugin (actually used to generate openstack-manuals)

Description: It represents the standard tool to generate Openstack documentation manuals, uses maven, maven plugins, clouddocs-maven-plugins; location of finally generated files is the object of a small Bash script that will reside as Post-actions

- Input files: .xml
- Output: .html & .pdf
- Pros:
    - quite easy for initial setup
    - uses openstack documentation generation flows as for openstack-manuals (clouddocs-maven-plugin), maven installs all you need generate the documentation
- Cons:
    - could be tricky to generate a custom layout, knowledge about Maven plugins required, .pom editing
    - dependent of multiple maven plugins
    - input files are .xml and xml editing knowledge is required
- Tested: roughly, functional tests only

Sphinx & LaTeX suite

Description: The easiest to install, the cleanest in matter of folder & files structure, uses standard tools available in repositories; location of finally generated files is the object of a small Bash script that will reside as Post-actions

- Input files: .rst as default
- Output: .html & .pdf
- Pros:
    - standard tools: Python Sphinx, LaTeX suite
    - destination files are generated fast
    - standard reStructuredText as default; other inputs can be configured
    - Sphinx's installation is very clean in matters of folder structure; the cleanest from all tested variants
    - latex suite is also easy to install via yum/apt and available in general repos
    - everyone is migration from other tools to Spinx lately; it provides more control and better looking documentation
    - can be used also for source-code documentation, specially if you use Python
- Cons:
    - takes some time to customize the output in matters of template, requires custom html header/footer
    - latex suite is quite substantial in amount of packages and consumed space (around 1.2 GB)
- Tested: roughly, functional tests only

Documentation tracking

Revision: 9d00c5d1ebb28bf0c69178c277eef56984a47fb9

Build date: Wed Jun 17 21:49:13 UTC 2015