Configure Repo for Documentation

This page provides instructions for configuring your component's documentation area so the O-RAN-SC Jenkins will automatically build and deploy the material to https://docs.o-ran-sc.org/ upon change of any file within your docs/ directory.

If you have questions about this process or need help, please contact the O-RAN-SC Documentation Project Technical Lead, weichen ni (niweichen@chinamobile.com)

Note: please add weichen ni as a reviewer on all Gerrit doc changesets if you have any questions.

Configuration needed for your _repo/

Please use the automated setup process that is documented here:

      https://docs.releng.linuxfoundation.org/projects/global-jjb/en/latest/jjb/lf-rtdv3-jobs.html

If that process fails or you need to trouble-shoot problems, this page gives step by step details.

Manual Step 1: Add files to your-repo root

Step 1.1: Create or extend file .gitignore in the root of your repository with the following content:

```
# documentation
.tox
docs/_build/
```

Step 1.2: Create RTD config file

Create a file in the root of your repository called .readthedocs.yaml (yes there's a leading dot) with the following content:

```
---
version: 2

formats:
  - htmlzip

build:
  os: "ubuntu-22.04"
    tools:
      python: "3.7"

python
  install:
    - requirements: docs/requirements-docs.txt

sphinx:
  configuration: docs/conf.py
```

Step 1.3: Create file tox.ini

Create or extend a file in the root of your repostory called tox.ini with the following content.

Please note that Python projects must not use all entries shown here. The entry "skipsdist = true" is only appropriate if the project has no setup.py file that defines how to build distributions.

```
# documentation only
[tox]
minversion = 2.0
envlist =
    docs,
    docs-linkcheck,
skipsdist = true

[testenv:docs]
basepython = python3
deps =
    sphinx
    sphinx-rtd-theme
    sphinxcontrib-httpdomain
    recommonmark
    lfdocs-conf

commands =
    sphinx-build -W -b html -n -d {envtmpdir}/doctrees ./docs/ {toxinidir}/docs/_build/html
    echo "Generated docs available in {toxinidir}/docs/_build/html"
whitelist_externals = echo

[testenv:docs-linkcheck]
basepython = python3
deps = sphinx
       sphinx-rtd-theme
       sphinxcontrib-httpdomain
       recommonmark
       lfdocs-conf
commands = sphinx-build -W -b linkcheck -d {envtmpdir}/doctrees ./docs/ {toxinidir}/docs/_build/linkcheck
```

Manual Step 2: Add files to your-repo/docs

In these steps you will create files in the "docs" subfolder of your repository. Please note that "docs" is a magic string, the directory name must be exactly that. Please use templates from the DOC area to create basic documentation files like "index.rst" and so on.

Step 2.1: Make a new directory "docs/_static"

```
mkdir docs/_static
```

Step 2.2: Create an image file logo.png in the new docs/_static/ directory. Here's a good source for the image:

https://gerrit.o-ran-sc.org/r/gitweb?p=doc.git;a=blob;f=docs/_static/logo.png;h=c3b6ce56468d87a3d9463ee75297b3895fc9a414;hb=refs/heads/master

Step 2.3: Create file docs/conf.py with exactly the following content:

```
from docs_conf.conf import *
linkcheck_ignore = [
    'http://localhost.*',
    'http://127.0.0.1.*',
    'https://gerrit.o-ran-sc.org.*'
]
```

Step 2.4 Create file docs/conf.yaml with the following content, but be sure to use your repo name, not "your-repo":

```
---
project_cfg: oran
project: your-repo
```

Step 2.5: Create a new image file docs/favicon.ico here's a good source for the icon:

https://gerrit.o-ran-sc.org/r/gitweb?p=doc.git;a=blob;f=docs/favicon.ico;h=00b0fd0ef0b4e78fbb8cdb413ce84561dfeb404f;hb=refs/heads/master

Step 2.6: Create file docs/requirements-docs.txt with exactly the following content:

```
sphinx
sphinx-rtd-theme
sphinxcontrib-httpdomain
recommonmark
lfdocs-conf
```

Step 3: Test locally

Working in the root of the repository, issue the following command to generate documentation. This assumes you have Python and Tox installed on the machine where you are working.

```
tox -e docs,docs-linkcheck
```

Note a common error message (see below) comes from RST formatting warning (warnings treated as errors). Fix the underlying RST (in the example that's installation-guides.rst) and try again.

```
### Possible/typical error message in the output // fix by fixing the underlying RST document Warning, treated
as error: /home/fedora/tmp/doci/aaa/docs/index.rst:23:toctree contains reference to document 'installation-
guides' that doesn't have a title: no link will be generated ERROR: InvocationError for command '/home/fedora
/tmp/doci/aaa/.tox/docs-linkcheck/bin/sphinx-build -W -b linkcheck -d /home/...(exited with code 2)
```

If everything correct, you will see a succeed or successful output after running 'tox -e docs,docs-linkcheck'.

It is the time to upload your change to gerrit. When you merged your change, you project docs will show on the page : https://readthedocs.org/projects/o-ran-sc-doc/. If you can scroll down, and you project name will display on the right.

Please let DOC teamweichen niknow when you complete Step 3, as the configuration work in your_repo is finished, DOC team will handle Step 4 and make your docs work.

DOC team will do the Step 4 in DOC repo! If you are not from OSC DOC project, please do not do this step.

Step 4: Add a link to your-repo documentation

4.1 Clone the doc area via ssh. (best to just go here and select the ssh option "https://gerrit.o-ran-sc.org/r/admin/repos/doc"

```
git clone "ssh://"lfid"@gerrit.o-ran-sc.org:29418/doc"
```

4.2 Add a mapping from key to URL in the conf.py file, for example:

https://www.sphinx-doc.org/en/1.5/markup/inline.html#cross-referencing-arbitrary-locations

$ grep -r portal-ric-dashboard .
./index.rst:* :doc:`RIC Dashboard <portal-ric-dashboard:index>`
./conf.py:intersphinx_mapping['portal-ric-dashboard'] = ('https://docs.o-ran-sc.org/projects/o-ran-sc-portal-ric-dashboard/en/%s' % branch, None)

```
intersphinx_mapping['your-repo'] = ('https://docs.o-ran-sc.org/projects/o-ran-sc-your-repo/en/%s' % branch,
None)
```

4.3 Add a link to your mapping key in an appropriate file such as index.rst, for example:

```
* :doc:`Your Project <your-repo:index>`
```

Step 5: Test publication

Edit a file in your docs/ area, commit the file to git and push your commit to Gerrit for review. You should see a documentation build job start.