



LFN Developer & Testing Forum

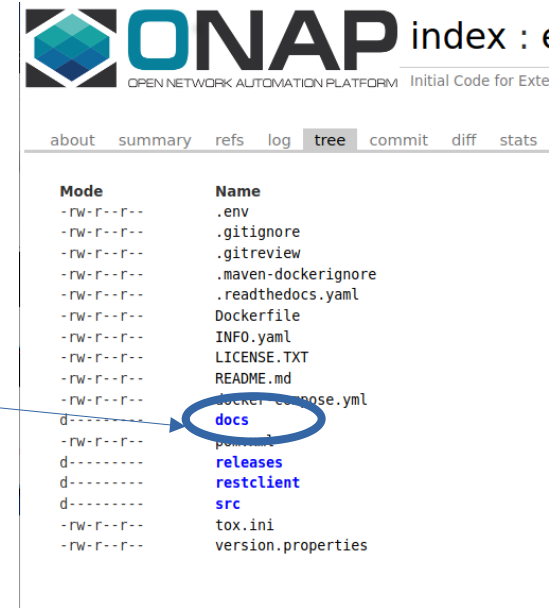
ONAP Documentation Starter Kit

Intro + Demo

Eric Debeau, Andreas Geissler, Sofia Wallin | 2021 February LFN Developer & Testing Forum

Documentation organization

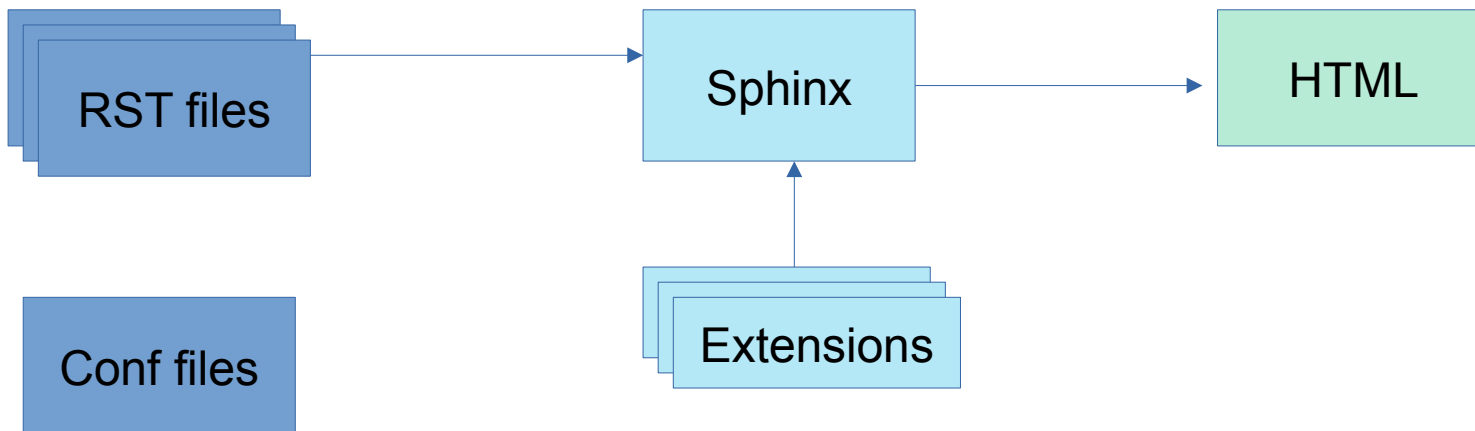
- Each project manages its own documentation
 - Documentation files are stored in a dedicated directory under each repo (docs)
- A mechanism enables to link all the projects together



Documentation As Code

- Documentation is based on RST files
 - RST (reSTructuredText) is based on a textual format
 - Enabling powerful capabilities
 - Links, navigation, index, search
 - Extensions to include code, diagrams...
- RST files are managed in git repos as any other file

Tooling chain



Each project with doc
[git.onap.org/\\$project/doc](https://git.onap.org/$project/doc)

LFN Jenkins jobs
<https://git.onap.org/ci-management/tree/jjb/doc>

ReadTheDocs
<https://docs.onap.org/en/latest/>

Configuration files

Conf.py

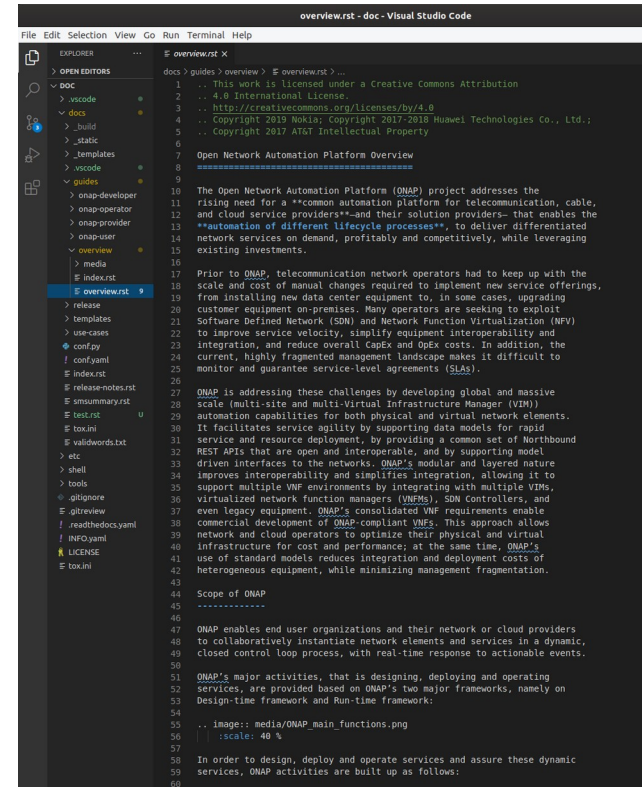
```
from docs_conf.conf import *
branch = 'latest'
master_doc = 'index'
linkcheck_ignore = [
    'http://localhost',
]
intersphinx_mapping = {}
html_last_updated_fmt = '%d-%b-%y %H:%M'
def setup(app):
    app.add_stylesheet("css/ribbon.css")
```

Conf.yaml

```
---
project_cfg: onap
project: onap
# Change this to ReleaseBranchName to modify the header
default-version: latest
#
```


Setup a local environment

- To help writing RST files
 - Syntax highlight, spell errors...
- To help detecting errors
 - RST syntax, bad links
- To preview output contents as produced by ONAP



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
  DOC
  .vscode
  docs
  _build
  _static
  templates
  .vscode
  guides
  onap-developer
  onap-operator
  onap-provider
  onap-user
  overview
  media
  overview.rst
  release
  templates
  use-cases
  confpy
  confyaml
  index.rst
  release-notes.rst
  E smsummary.rst
  E test.rst
  E tox.ini
  E valldocwords.txt
  etc
  shell
  tools
  .gitignore
  .gitreview
  / .readthedocs.yaml
  / INFO.yaml
  LICENSE
  E tox.ini
  overview.rst - doc - Visual Studio Code
  1 .. This work is licensed under a Creative Commons Attribution
  2 .. 4.0 International License.
  3 .. http://creativecommons.org/licenses/by/4.0
  4 .. Copyright 2018 Huawei; Copyright 2017-2018 Huawei Technologies Co., Ltd.;
  5 .. Copyright 2017 AT&T Intellectual Property
  6
  7 Open Network Automation Platform Overview
  8 =====
  9
  10 The Open Network Automation Platform (ONAP) project addresses the
  11 rising need for a common automation platform for telecommunication, cable,
  12 and cloud service providers—and their solution providers—that enables the
  13 automation of different lifecycle processes, to deliver differentiated
  14 network services on demand, profitably and competitively, while leveraging
  15 existing investments.
  16
  17 Prior to ONAP, telecommunication network operators had to keep up with the
  18 scale and cost of manual changes required to implement new service offerings,
  19 from installing new data center equipment to, in some cases, upgrading
  20 customer equipment on-premises. Many operators are seeking to exploit
  21 Software Defined Network (SDN) and Network Function Virtualization (NFV)
  22 to improve service velocity, simplify equipment interoperability and
  23 integration, and reduce overall CapEx and OpEx costs. In addition, the
  24 current, highly fragmented management landscape makes it difficult to
  25 monitor and guarantee service-level agreements (SLAs).
  26
  27 ONAP is addressing these challenges by developing global and massive
  28 scale (multi-site and multi-Virtual Infrastructure Manager (VIM))
  29 automation capabilities for both physical and virtual network elements.
  30 It facilitates service agility by supporting data models for rapid
  31 service and resource deployment, by providing a common set of Northbound
  32 REST APIs that are open and interoperable, and by supporting model
  33 driven interfaces to the networks. ONAP's modular and layered nature
  34 improves interoperability and simplifies integration, allowing it to
  35 support multiple VNF environments by interacting with multiple VNFs,
  36 virtualized network function managers (VNFMs), SDN controllers, and
  37 even legacy equipment. ONAP's consolidated VNF requirements enable
  38 commercial development of ONAP-compliant VNFs. This approach allows
  39 network and cloud operators to optimize their physical and virtual
  40 infrastructure for cost and performance; at the same time, ONAP's
  41 use of standard models reduces integration and deployment costs of
  42 heterogeneous equipment, while minimizing management fragmentation.
  43
  44 Scope of ONAP
  45 -----
  46
  47 ONAP enables end user organizations and their network or cloud providers
  48 to collaboratively instantiate network elements and services in a dynamic,
  49 closed control loop process, with real-time response to actionable events.
  50
  51 ONAP's major activities, that is designing, deploying and operating
  52 services, are provided based on ONAP's two major frameworks, namely on
  53 Design-time framework and Run-time framework:
  54
  55 .. image:: media/ONAP_main_functions.png
  56 .. scale: 40 %
  57
  58 In order to design, deploy and operate services and assure these dynamic
  59 services, ONAP activities are built up as follows:
  60
```

Guide on Wiki: <https://wiki.onap.org/display/DW/Local+environment+to+write+RST+files>

Automatic preview

Preview overview.rst - doc - Visual Studio Code

EXPLORER

- docs
- guides
- overview
- media
- release
- templates
- use-cases
- conf.py
- conf.yaml
- index.rst
- release-notes.rst
- summary.rst
- test.rst
- tox.ini
- validwords.txt
- etc
- shell
- tools
- gitignore
- gitreview
- readthedocs.yaml
- INFO.yaml
- LICENSE
- tox.ini

Preview overview.rst

ONAP Overview - Open Network Automation Platform Overview

Open Network Automation Platform Overview

The Open Network Automation Platform (ONAP) project addresses the rising need for a common automation platform for telecommunication, cable, and cloud service providers—and their solution providers—that enables the automation of different lifecycle processes, to deliver differentiated network services on demand, profitably and competitively, while leveraging existing investments.

Prior to ONAP, telecommunication network operators had to keep up with the scale and cost of manual changes required to implement new service offerings, from installing new data center equipment to, in some cases, upgrading customer equipment on-premises. Many operators are seeking to exploit Software Defined Network (SDN) and Network Function Virtualization (NFV) to improve service velocity, simplify equipment interoperability and integration, and reduce overall CapEx and OpEx costs. In addition, the current, highly fragmented management landscape makes it difficult to monitor and guarantee service-level agreements (SLAs).

ONAP is addressing these challenges by developing global and massive scale (multi-site and multi-Virtual Infrastructure Manager (VIM)) automation capabilities for both physical and virtual network elements. It facilitates service agility by supporting data models for rapid service and resource deployment, by providing a common set of Northbound REST APIs that are open and interoperable, and by supporting model driven interfaces to the networks. ONAP's modular and layered nature improves interoperability and simplifies integration, allowing it to support multiple VNF environments by integrating with multiple VIMs, virtualized network function managers (VNFM), SDN Controllers, and even legacy equipment. ONAP's consolidated VNF requirements enable commercial development of ONAP-compliant VNFs. This approach allows network and cloud operators to optimize their physical and virtual infrastructure for cost and performance; at the same time, ONAP's use of standard models reduces integration and deployment costs of heterogeneous equipment, while minimizing management fragmentation.

Scope of ONAP

ONAP enables end user organizations and their network or cloud providers to collaboratively instantiate network elements and services in a dynamic, closed control loop process, with real-time response to actionable events.

ONAP's major activities, that is designing, deploying and operating services, are provided based on ONAP's two major frameworks, namely on Design-time framework and Run-time framework:

```
graph TD
    subgraph ONAP
        subgraph Design-time_framework [Design-time framework]
            SD[Service Design]
        end
        subgraph Run-time_framework [Run-time framework]
            SDep[Service Deployment]
            SOps[Service Operations]
        end
    end
```

In order to design, deploy and operate services and assure these dynamic services, ONAP activities are built up as follows:

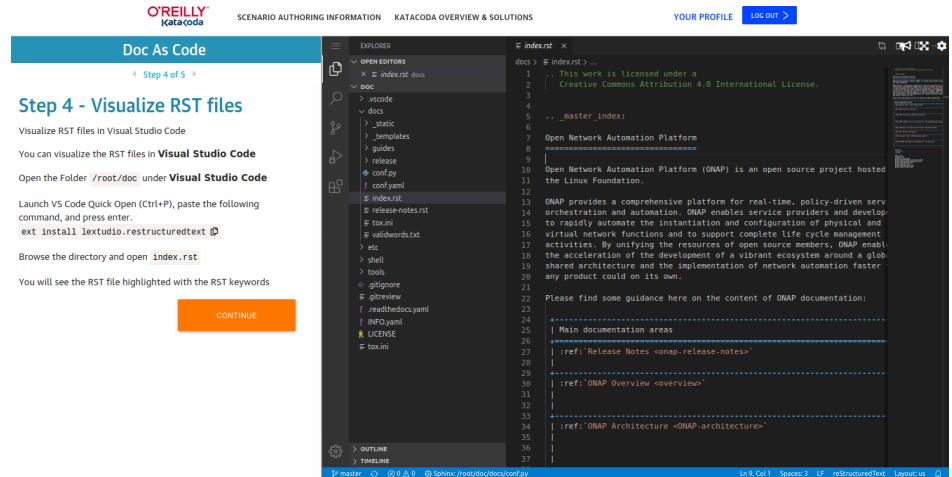
- Service design – Service design is built on a robust design framework that allows specification of the service in all aspects – modeling the resources and relationships that make up the service, specifying the policy rules that guide the service behavior, specifying the applications, analytic and closed control loop events needed for the elastic management of the service.
- Service deployment – Service deployment is built on an orchestration and control framework that is policy-driven (Service Orchestrator and

A short demo

- Linux Ubuntu 19.10
- Libraries installed
- Visual Studio Code with extensions

A beta tool to start with

- No installation required
- All actions in the browser
- For beginners to start



https://www.katacoda.com/ericdebeau/scenarios/doc_as_code
<https://www.katacoda.com/courses/vscode/playground>



OLF NETWORKING

LFN Developer & Testing Forum