Harmonization documentation across LFN projects

Eric Debeau, Orange
Greg Glover, AT&T

March, 26 2018
Requirements vary

• Different users from different (vendors, service providers, academic…)
  - Developer
  - User
  - Contributor
  - Newbie

• Different stages on the project
  - Stable documentation
  - Documentation for the project life

• Different expectations depending on project/component
  - Installation guide
  - User guide
  - Developer guide
  - End to end call flows
Tool chain

RST → Sphinx → HTML

repos → readthedocs.io
Some Best Practice examples

- OpenDayLight
  - JavaDoc documentation

- OPNFV
  - Python components documentation

- ONAP Call flows

- APIs
  - Swagger spec
  - Versioning
xtesting.core.feature module

Define the parent classes of all Xtesting Features.

Feature is considered as TestCase offered by Third-party. It offers helpers to run any python method or any bash command.

class xtesting.core.feature.BashFeature(**kwargs)

Bases: xtesting.core.feature.Feature

Class designed to run any bash command.

def execute(**kwargs)

Execute the cmd passed as arg

Args:

kwargs: Arbitrary keyword arguments.

Returns:

0 if cmd returns 0, -1 otherwise.
ONAP Call flow: plantUML

.. uml::

@startuml
title vFW/vFWCL and vDNS VNF Onboarding (R1)
vFW/vFWCL and vDNS use the same flows but they are separate VNFs/Services

ONAP_User -> SDC : vFW_vDNS resource onboarding (HEAT)
note right : vFWCL (vpg & vfw, vsn)\nvFW (vpg, vfw, vsn)\nvDNS (vpg, vlb,vdns) + DNSScaling (vdns)

ONAP_User -> SDC : vFW_vDNS service onboarding
ONAP_User -> SDC : vFW_vDNS distribution

SDC -> SO : artifact distribution\nNOTIFY,DOWNLOAD,DEPLOY_OK
SDC -> AAI : artifact distribution\nNOTIFY,DOWNLOAD,DEPLOY_OK
SDC -> SDNC : artifact distribution\nNOTIFY,DOWNLOAD,DEPLOY_OK
@enduml

VFW/vFWCL and vDNS VNF Onboarding (R1)

VFW/vFWCL and vDNS use the same flows but they are separate VNFs/Services

http://docs.onap.org/en/latest/use-cases/vfw.html
ONAP REST API

- Swagger file mandatory for ONAP
- Still various possibilities to describe
- Proposal example for new component
Other LFN Best Practice Experiences?
ONAP Documentation Goals for Beijing (Release 2)

• Updates from Amsterdam (Release 1)
• Interfaces:
  - API documentation – Standard spécification, descriptions & versioning
  - API differentiation - Rest, Healthcheck, Components
  - OpenStack Interfaces
• ONAP for Newbies
  - Call flow description – End to end use cases
  - End User - End to end descriptions
  - ONAP “Light” for SEs, Architects
  - Wiki Sandbox for Developers
• VNF Reference List
• ONAP Health & Maturity metrics
• Tool Chain and Utilities
  - Tools to facilitate RST conversion
  - Strong Doc8 validation
  - LF Tool Chain documentation
ONAP volunteers?

https://onap.readthedocs.io/en/amsterdam/
THANK YOU;-)
Welcome to the OpenDaylight Handbook!

This handbook provides details on various aspects of OpenDaylight from the user guides to the developer guides and tries to act as a single point of contact for all documentation related articles in OpenDaylight. If you would like to contribute to the Handbook please refer to the Documentation Guide.

Content for OpenDaylight Users

The following content is intended for people who would like to deploy, use, or just learn more about OpenDaylight.

- Release Notes
- Getting Started Guide
- OpenDaylight User Guide
- OpenDaylight with Openstack Guide

Content for OpenDaylight Developers

The following content is intended for developers building applications or code on top of OpenDaylight, but who do not plan to modify OpenDaylight code itself.

- Developer Guide
- Java API Documentation

Content for OpenDaylight Contributors

The following content is intended for developers who either currently participate in the development of OpenDaylight or would like to start.
whether you are an employee of a member company or just passionate about network transformation.

**Testing Frameworks**

- Testing Framework Overview
- Testing User Guides
- Testing Developer Guides

**Developer**

- Documentation Guide
- OPNFV Projects

**Infrastructure**

- Infrastructure Overview
- Continuous Integration - CI
- Cross Community Continuous Integration - XCI

Found a typo or any other feedback? Send an email to users@opnfv.org or talk to us on IRC.

© Copyright 2017, OPNFV. Licensed under CC BY 4.0.

Created using Sphinx 1.3.1.

Main Page

Welcome to the fd.io Developer Wiki

This wiki supports Linux Foundation ID single-sign-on and registration with the link at the top this page. Other components of the fd.io Project that do not support single-sign-on will directly request your Linux Foundation ID username and password for login.

fd.io projects

- VPP
- Honeycomb
- CSIT
- NSH SFC

https://wiki.fd.io/view/Main_Page
Gerrit Guide

Gerrit is an opensource web-based collaborative code review tool that integrates with Git. Gerrit provides a framework for reviewing code commits before it merges into the code base. The changes are not made a part of the project until a code review completes. Gerrit is also a good collaboration tool for storing the conversations that occur around the code commits.

Note
Here is more information on Gerrit

Prerequisites

Before you get started, you should have:

- an LFID account (sign up here)
- git installed (see: http://www.git-scm.com/downloads)
- git configured with your name, e-mail address and editor

```bash
    git config --global user.name "Firstname Lastname"
    git config --global user.email "email@address.com"
    git config --global core.editor "text-editor-name"
```

Note
User name and e-mail address (including capitalization) must match what you entered when creating your account.