

FaaS

Welcome to FaaS

- [Welcome to FaaS](#)
- [Introduction](#)
- [Project Information](#)
- [Documentation](#)
- [Test and Integration Plan](#)
- [Release Planning](#)
 - [List of all subpages \(used or unused\)](#)
- [Release Notes](#)

Introduction

In general, FaaS project aims to create a common abstraction layer on top of a physical network, so northbound API or services can be easier to be mapped onto the physical network as concrete device configuration. The common abstraction layer models the physical network as a topology that consists of abstracted node - fabrics. Each fabric is abstraction of a portion of the physical network usually within the same control plane and uses similar data path technique, such as VXLAN or VLAN. Every fabric offers a set of unified services as well as primitive constructs to create and manage a logical network life cycle according to users' requirement.

Using FaaS to deploy network services

- decouples user network services from vendor and technology specific implementation, avoid vendor locked in.
- enable service deploy and control automation, massively reduce OPEX as well as CAPEX.
- improve service deployment agility.

If you think of a computer as a network, FaaS provides a set of system calls for network applications. Those system calls has two layers. One layer is called driver layer which abstracts vendor/technology specific technique, which is realized on top of a specific fabric object. The second layer is built on top of the whole network which consists of a topology of fabrics. The second layer provides neutron like user level logical network abstractions. With those two layers' abstraction, the controller can be extended to support more devices and technology while keeps users' services transparent to the change.

Moreover, applications built on top of FaaS is using high level primitives to program the network. To make an analogy, using FaaS to build applications is like a using C lib other than assembly to program a machine.

We also realize that the FaaS will evolve as underneath technology evolves just like system calls are extended as OS evolves. FaaS has to evolve in a backward compatible fashion.

Project Information

- [Project Proposal](#)
- Trello Board: <https://trello.com/b/RcqkTo70/opendaylight-faas>
- Chat: [odlfaas.slack.com](#) ([email me](#) for the invite)

Documentation

Getting Started for Users

Getting Started for Developers

Requirements

Test and Integration Plan

Project Facts

Project Creation Date: August 6, 2015

Lifecycle State: Incubation

Type: App

Primary Contact: Xingjun Chu
<xingjun.chu@huawei.com>

Project Lead: Xingjun Chu
<xingjun.chu@huawei.com>

Committers:

- Xingjun Chu <xingjun.chu@huawei.com>
- Henry Yu <henry.yu1@huawei.com>
- Danian Shi <shidianian@huawei.com>
- Ziyang Yao <yaoziyang@huawei.com>
- Khaldoon Al Zoubi <khaldoon.alzoubi@huawei.com>
- Guoli Yin <guoli.yin@huawei.com>
- Song Wei <songwei80@huawei.com>
- Dong Feng <leslin.dongfeng@huawei.com>
- Alex Zhang <alexzhang@china-mobile.com>
- Yapeng Wu <yapeng.wu@huawei.com>

IRC: [freenode.net@opendaylight-faas](irc://freenode.net/opendaylight-faas)

Mailing List: faas-dev@lists.opendaylight.org
Archives: unspecified

Meetings: See [Community Meetings](#)

Repository: [Repository](#)

Jenkins: [Jenkins](#)

Gerrit Patches: [Gerrit](#)

Open Bugs: [Bugzilla](#)

<https://wiki.opendaylight.org/view/FaaS:SystemTestPlan>

Release Planning

Release	Release Plan	Release Notes	Release Review	Installation Guide	User Guide	Developer Guide	Operations Guide	How-To's /Tutorials	Feature Tree	Feature Tests
Beryllium	Release Plan	Release Notes	Release Review	-	-	-	-	-	-	-
Boron	Release Plan	Release Notes	Release Review	-	-	-	-	-	-	-
Carbon	Release Plan	Release Notes	Release Review	-	-	-	-	-	-	-
Nitrogen	Release Plan	Release Notes	Release Review	-	-	-	-	-	-	-
Oxygen	Release Plan	Release Notes	Release Review	-	-	-	-	-	-	-
SNAPSHOT	-	-	-	-	User Guide	Developer Guide	Operations Guide	How-To's /Tutorials	-	System Test Plan

List of all subpages (used or unused)

FaaS/Release Plans/Beryllium Release Plan	FaaS:Beryllium	FaaS:Beryllium Release Notes
FaaS:Beryllium Release Plan	FaaS:Beryllium Release Review	FaaS:Boron Release Notes
FaaS:Boron Release Plan	FaaS:Boron Release Review	FaaS:Carbon Release Notes
FaaS:Carbon Release Plan	FaaS:Carbon Release Review	FaaS:Developer Guide
FaaS:GBPIntegration	FaaS:HighLevelArchitecture	FaaS:How-To's/Tutorials
FaaS:Main	FaaS:Nitrogen Release Notes	FaaS:Nitrogen Release Plan
FaaS:Nitrogen Release Review	FaaS:Oxygen Release Plan	FaaS:Programmer Guide
FaaS:SFCIntegration	FaaS:SystemTestPlan	FaaS:User Guide

Release Notes