Open Source MANO

OSM MR Hackfest – Hack 5a
End-to-end VNF Package Onboarding in OSM

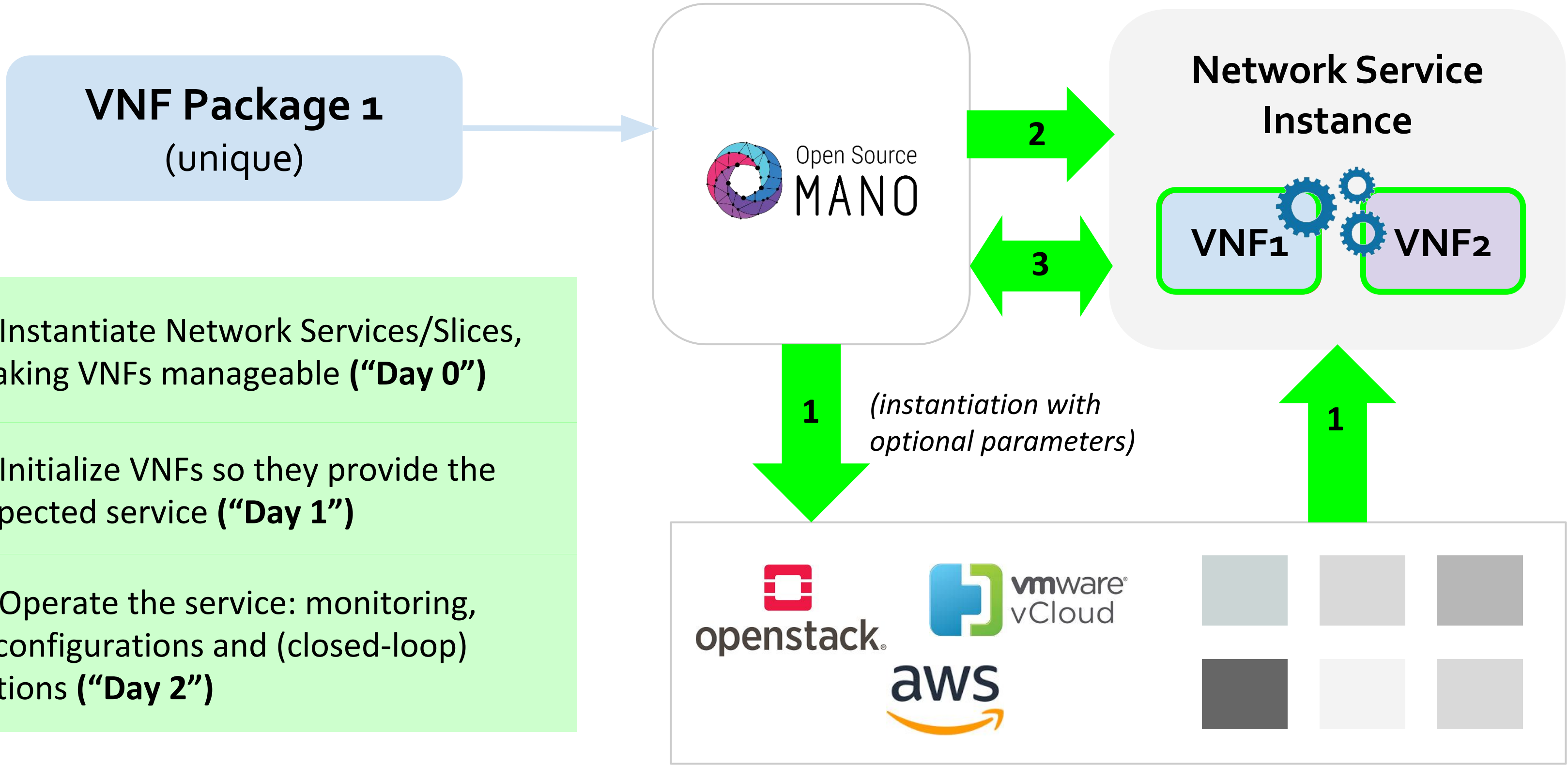
Gianpietro Lavado (Whitestack)
Felipe Vicens (Atos)

Open Source MANO

Introduction to the VNF Onboarding Task Force & Resources

OSM VNF Onboarding TF Vision

To achieve **completely onboarded VNFs**, with **unique packages** that **fulfill the NFV vision** by **providing value to CSPs.**



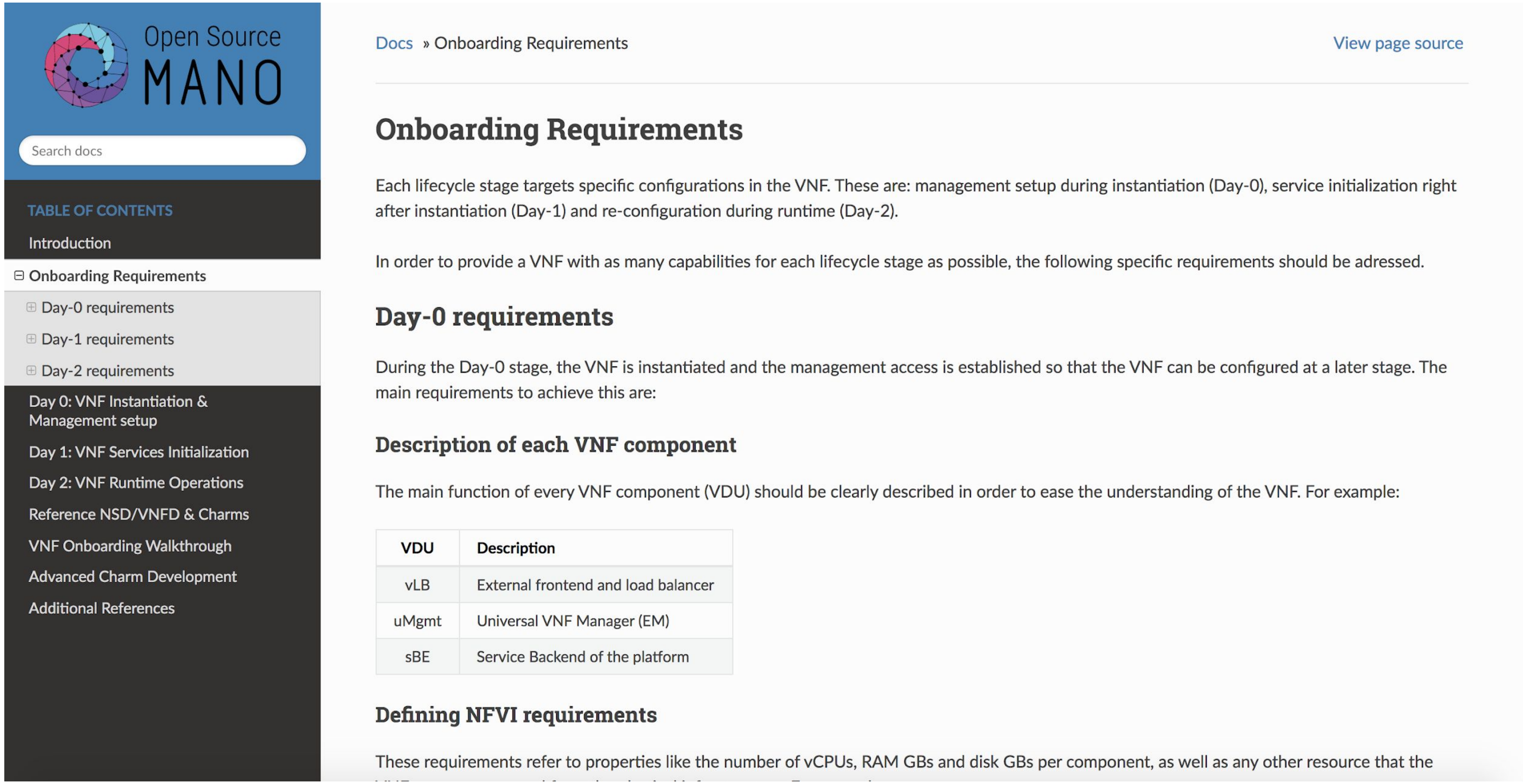
1. Instantiate Network Services/Slices, making VNFs manageable **("Day 0")**

2. Initialize VNFs so they provide the expected service **("Day 1")**

3. Operate the service: monitoring, reconfigurations and (closed-loop) actions **("Day 2")**

Resources: Documentation

- The **OSM VNF Onboarding Guidelines** (live, online documentation)



Onboarding Requirements

Each lifecycle stage targets specific configurations in the VNF. These are: management setup during instantiation (Day-0), service initialization right after instantiation (Day-1) and re-configuration during runtime (Day-2).

In order to provide a VNF with as many capabilities for each lifecycle stage as possible, the following specific requirements should be adressed.

Day-0 requirements

During the Day-0 stage, the VNF is instantiated and the management access is established so that the VNF can be configured at a later stage. The main requirements to achieve this are:

Description of each VNF component

The main function of every VNF component (VDU) should be clearly described in order to ease the understanding of the VNF. For example:

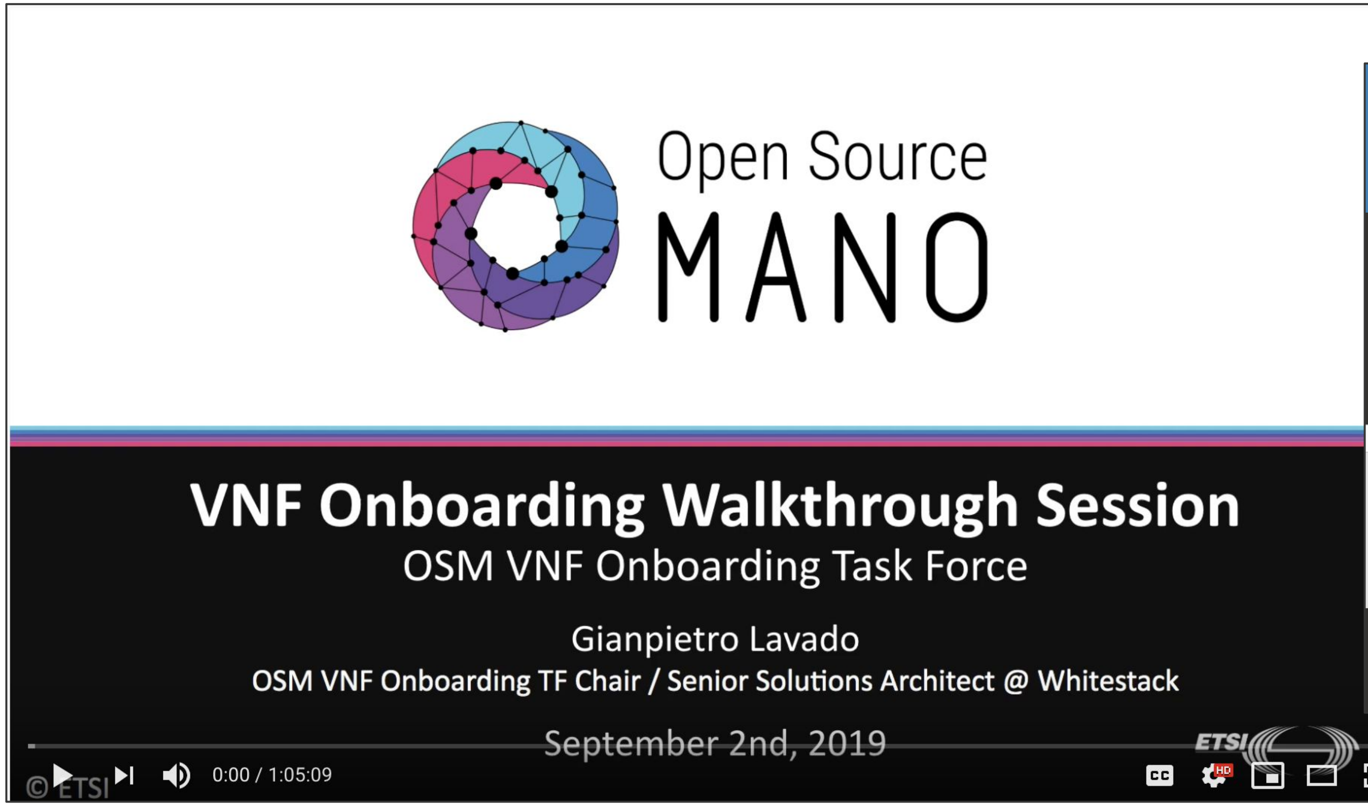
VDU	Description
vLB	External frontend and load balancer
uMgmt	Universal VNF Manager (EM)
sBE	Service Backend of the platform

Defining NFVI requirements

These requirements refer to properties like the number of vCPUs, RAM GBs and disk GBs per component, as well as any other resource that the

Resources: Onboarding Examples

- An onboarding example is available as a walkthrough video and document:



VNF Onboarding Walkthrough Session

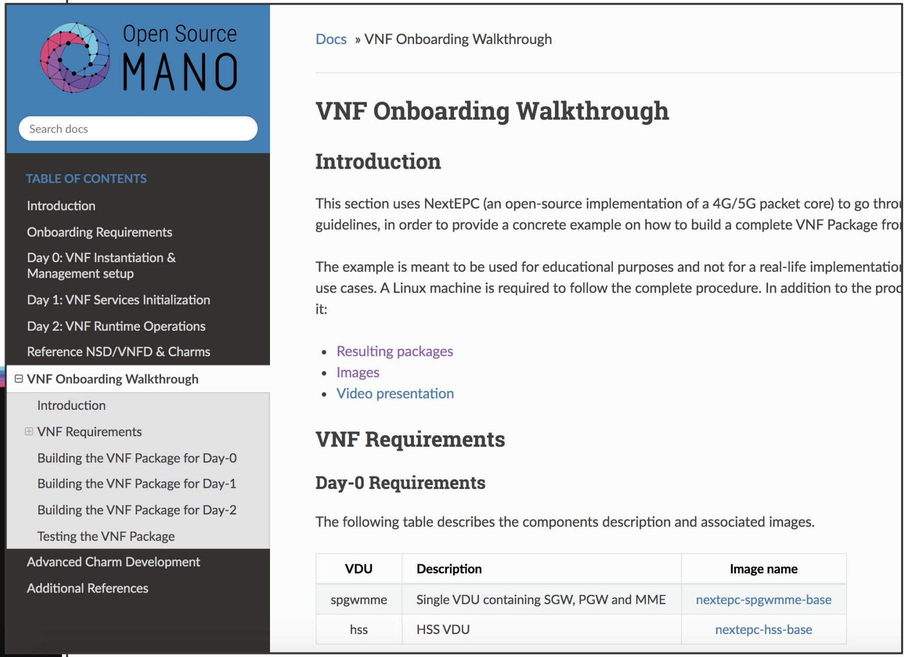
OSM VNF Onboarding Task Force

Gianpietro Lavado

OSM VNF Onboarding TF Chair / Senior Solutions Architect @ Whitestack

September 2nd, 2019

https://www.youtube.com/watch?v=hZzwwy9wNRE



Docs » VNF Onboarding Walkthrough

VNF Onboarding Walkthrough

Introduction

This section uses NextEPC (an open-source implementation of a 4G/5G packet core) to go thro
guidelines, in order to provide a concrete example on how to build a complete VNF Package fro

The example is meant to be used for educational purposes and not for a real-life implementatio
use cases. A Linux machine is required to follow the complete procedure. In addition to the proc
it:

- Resulting packages
- Images
- Video presentation

VNF Requirements

Day-0 Requirements

The following table describes the components description and associated images.

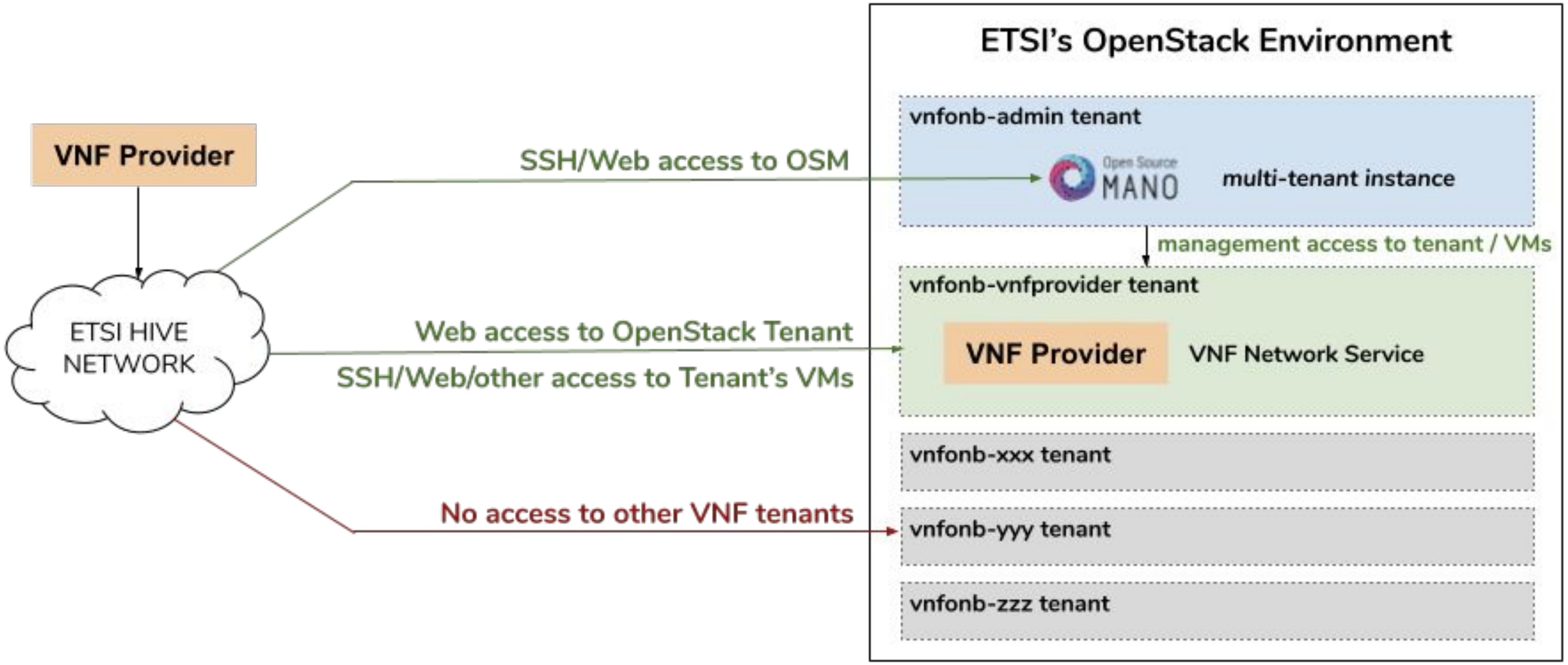
VDU	Description	Image name
spgwmme	Single VDU containing SGW, PGW and MME	nextepc-spgwmme-base
hss	HSS VDU	nextepc-hss-base

Open Source MANO
Search docs
TABLE OF CONTENTS
Introduction
Onboarding Requirements
Day 0: VNF Instantiation & Management setup
Day 1: VNF Services Initialization
Day 2: VNF Runtime Operations
Reference NSD/VNFD & Charms
⊟ VNF Onboarding Walkthrough
Introduction
⊞ VNF Requirements
Building the VNF Package for Day-0
Building the VNF Package for Day-1
Building the VNF Package for Day-2
Testing the VNF Package
Advanced Charm Development
Additional References

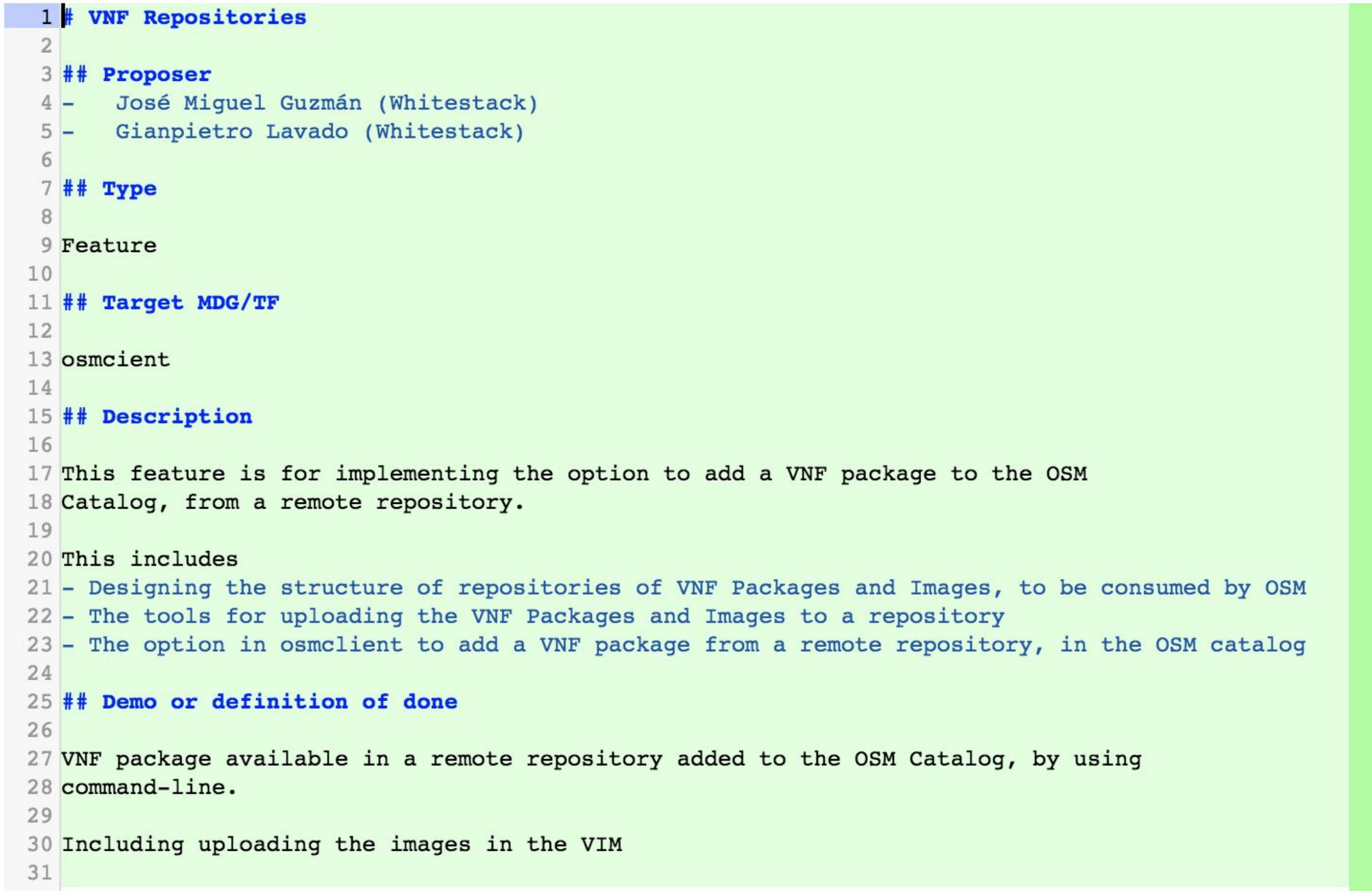
https://osm.etsi.org/docs/vnf-onboarding-guidelines/06-walkthrough.html#

Resources: Onboarding Labs

- “Onboarding Labs” are available to VNF providers that want to onboard their packages on OSM with community support.



Upcoming: OSM VNF Repositories



```
# VNF Repositories

## Proposer
-   José Miguel Guzmán (Whitestack)
-   Gianpietro Lavado (Whitestack)

## Type

Feature

## Target MDG/TF

osmcient

## Description

This feature is for implementing the option to add a VNF package to the OSM
Catalog, from a remote repository.

This includes
- Designing the structure of repositories of VNF Packages and Images, to be consumed by OSM
- The tools for uploading the VNF Packages and Images to a repository
- The option in osmclient to add a VNF package from a remote repository, in the OSM catalog

## Demo or definition of done

VNF package available in a remote repository added to the OSM Catalog, by using
command-line.

Including uploading the images in the VIM

```

OSM VNF Repositories (Feature 8178)

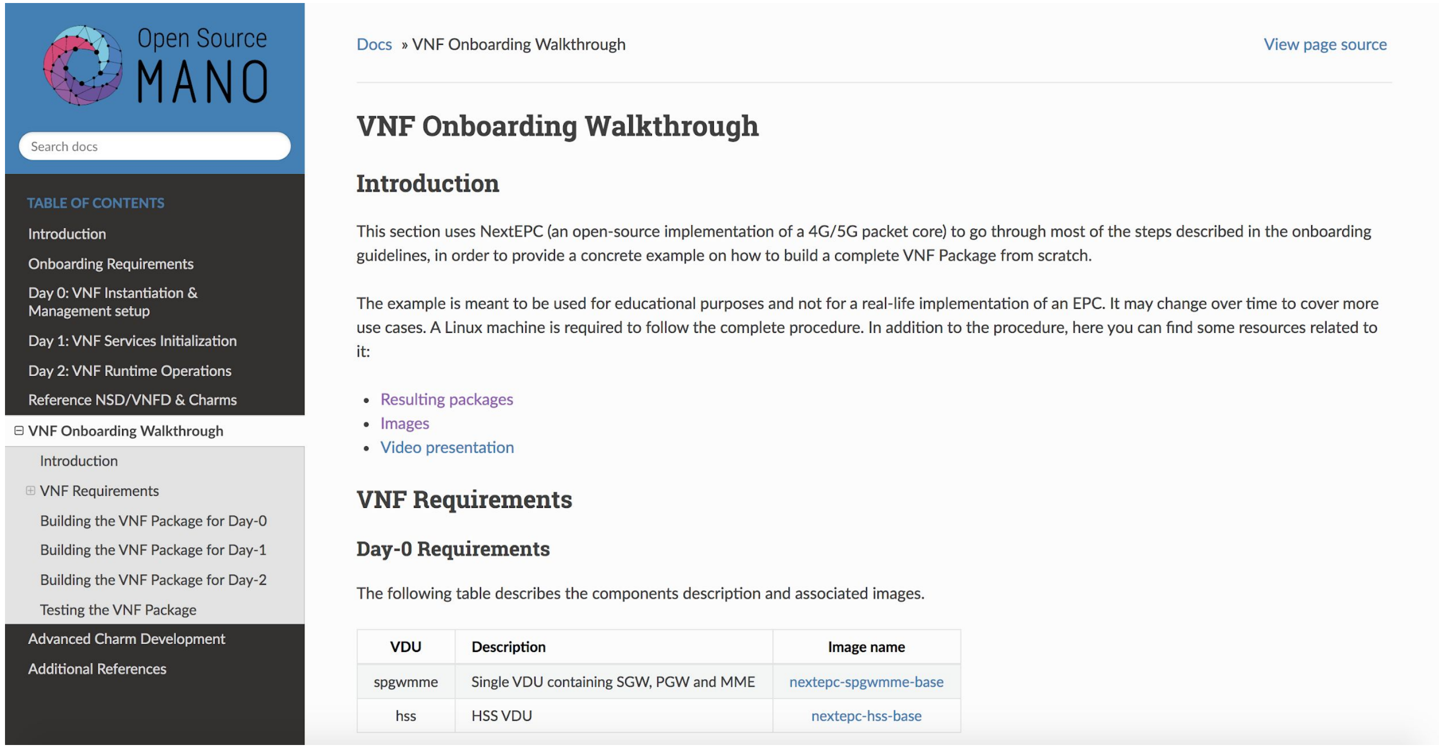
- Will let OSM operators add a VNF package to the OSM local catalogue, from an existing, remote repository.
- OSM client would be used for achieving any interaction with remote repositories, which would have a predefined structure.
- Uploading images to the VIM would also occur automatically.
- Feature: https://osm.etsi.org/gerrit/#/c/osm/Features/+/8178/6/osm-repositories.md

Open Source MANO

Hands-on: Building an Evolved Packet Core VNF package from scratch

Hands-on: Building an EPC VNF package

We will follow this guide.



Open Source MANO

Search docs

Docs » VNF Onboarding Walkthrough

View page source

VNF Onboarding Walkthrough

Introduction

This section uses NextEPC (an open-source implementation of a 4G/5G packet core) to go through most of the steps described in the onboarding guidelines, in order to provide a concrete example on how to build a complete VNF Package from scratch.

The example is meant to be used for educational purposes and not for a real-life implementation of an EPC. It may change over time to cover more use cases. A Linux machine is required to follow the complete procedure. In addition to the procedure, here you can find some resources related to it:

- Resulting packages
- Images
- Video presentation

VNF Requirements

Day-0 Requirements

The following table describes the components description and associated images.

VDU	Description	Image name
spgwmme	Single VDU containing SGW, PGW and MME	nextepc-spgwmme-base
hss	HSS VDU	nextepc-hss-base

Important considerations:

- Use **Visual Studio** or similar tool to quickly modify text files.
- The EPA section won’t be used this time, to match VIM available resources for Hackfest.
- For charms, the traditional “reactive framework” will be used (migration to Python Operators Framework is pending)

Open Source MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub

ETSI