ETSI GS NFV-EVE 004 V1.1.1 (2016-03)

GROUP SPECIFICATION

Network Functions Virtualisation (NFV);
Virtualisation Technologies;
Report on the application of Different Virtualisation Technologies in the NFV Framework

Disclaimer

This document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

DGS/NFV-EVE004

Keywords

network, NFV, virtualisation

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights 5

Foreword 5

Modal verbs terminology 5

1 Scope 6

2 References 6
2.1 Normative references 6
2.2 Informative references 6

3 Definitions and abbreviations 7
3.1 Definitions 7
3.2 Abbreviations 7

4 Virtualisation technologies 7
4.1 Introduction 7
4.2 Hypervisor-based solutions 7
4.2.1 Overview 7
4.2.2 Application to NFV 8
4.3 OS Containers 8
4.3.1 Overview 8
4.3.2 Application to NFV 9
4.4 Higher-level containers 9
4.4.1 Overview 9
4.4.2 Application to NFV 10
4.5 Nesting of virtualisation technologies 10
4.5.1 Overview 10
4.5.2 Application to NFV 10
4.5.2.1 General 10
4.5.2.2 Virtualisation Container Interface Mapping 11
4.5.2.3 Virtual Network Interface Mapping 11
4.6 Mixing of virtualisation technologies 12
4.6.1 Overview 12
4.6.2 Application to NFV 13
4.7 Mixing & Nesting of virtualisation technologies 13

5 Impact on the NFV architectural framework 14
5.1 Overview 14
5.2 OS Containers 14
5.2.1 Impact on the NFVI 14
5.2.2 Impact on Management and Orchestration 14
5.2.2.1 General 14
5.2.2.2 Impact on ETSI NFV Phase 1 specifications 14
5.2.2.3 Impact on ETSI NFV IFA specifications 15
5.2.3 Impact on VNFs 15
5.2.4 Impact on Security 15
5.2.5 Impact on NFV acceleration 16
5.3 Higher-level containers 16
5.3.1 Impact on the NFVI 16
5.3.2 Impact on Management and Orchestration 16
5.3.3 Impact on VNFs 17
5.3.4 Impact on Security 17
5.3.5 Impact on NFV acceleration 17
5.4 Nested Virtualisation 17
5.4.1 Impact on the NFVI 17
5.4.2 Impact on Management and Orchestration 17
5.4.3 Impact on VNFs 17
5.4.4 Impact on Security 17
5.4.5 Impact on NFV acceleration 17

5.5 Mixed Virtualisation 17
5.5.1 Impact on the NFVI 17
5.5.2 Impact on Management and Orchestration 18
5.5.3 Impact on VNFs 18
5.5.4 Impact on Security 18
5.5.5 Impact on NFV acceleration 18
5.6 Nested and Mixed Virtualisation 18
5.6.1 Impact on the NFVI 18
5.6.2 Impact on Management and Orchestration 18
5.6.3 Impact on VNFs 18
5.6.4 Impact on Security 18
5.6.5 Impact on NFV acceleration 18

6 Pros and Cons analysis 18
6.1 Features 18
6.2 Performance considerations 19
6.3 Security 19
6.4 Resources management 19
6.4.1 Resources utilization 19
6.4.2 Resources isolation 19
6.4.3 Instantiation Time 20
6.5 Lifecycle Management 20
6.6 Open Source Support 20

Annex A (informative): Authors & contributors 21

Annex B (informative): Change History 22

History 23

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document reviews virtualisation technologies and studies their impact on the NFV architectural framework and specifications. It also provides an analysis of the pros and cons of these technologies.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".

[i.2] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".

[i.3] ETSI GS NFV-INF 001: "Network Functions Virtualisation (NFV); Infrastructure; Overview".

[i.4] ETSI GS NFV-INF 004: "Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain".

[i.5] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration".

[i.6] ETSI GS NFV-INF 007: "Network Functions Virtualisation (NFV); Infrastructure; Methodology to describe Interfaces and Abstractions".

[i.7] ETSI GS NFV-INF 005: "Network Functions Virtualisation (NFV); Infrastructure; Network Domain".

[i.8] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV); Management & Orchestration; Network Service Descriptor template".

[i.9] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV); Management and Orchestration; VNF Packaging Specification".

[i.10] ETSI GS NFV-IFA 002: "Network Functions Virtualisation (NFV); Acceleration Technologies; VNF Interfaces Specification".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS NFV 003 [i.1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [i.1] and the following apply:

DevOps	Development and Operations
IPC	Interprocess Communication
LSM	Linux$^{\text{TM}}$ Security Module
OS	Operating System
OSS	Operations Support System
QoS	Quality of Service
TCP	Transmission Control Protocol
TPM	Trusted Platform Module
UTC	Coordinated Universal Time

4 Virtualisation technologies

4.1 Introduction

The ETSI NFV architectural framework as described in ETSI GS NFV 002 [i.2] identifies a virtualisation layer as a component of the NFV Infrastructure (NFVI). Typically, this type of functionality is provided for computing and storage resources in the form of hypervisors and VMs.

However, the NFV architectural framework does not restrict itself to using any specific virtualisation layer solution. Rather, it is expected that diverse virtualisation layers with standard features and open execution reference points towards Virtualised Network Functions (VNFs) and hardware can be used interchangeably.

4.2 Hypervisor-based solutions

4.2.1 Overview

A hypervisor is a software program that partitions the resources of a single hardware host and creates Virtual Machines (VM) isolated from each other. Each virtual machine appears to have the host's processor, memory and other resources, all to itself.

Each VM is assigned a virtualised CPU (vCPU), a virtualised NIC (vNIC) and a virtualised storage device (vStorage) created by the hypervisor. As pointed out in ETSI GS NFV-INF 004 [i.4], in practice, a vCPU may be a time sharing of a real CPU and/or in the case of multi-core CPUs, it may be an allocation of one or more cores to a VM. It is also possible that the hypervisor emulates a CPU instruction set that is different from the native CPU instruction set. However, emulation will significantly impact performance.

The hypervisor software runs either directly on top of the hardware (bare metal hypervisor, also known as Type I hypervisor) or on top of a hosting operating system (hosted hypervisor, also known as Type II hypervisor).

4.2.2 Application to NFV

The use of hypervisors is one of the present typical solutions for supporting the deployment of VNFs.

Although the NFV framework as defined in [i.2] is agnostic to the virtualisation technology, Management and Orchestration functions and interfaces as specified in ETSI GS NFV-MAN 001 [i.5] assume that virtualisation containers are virtual machines created by hypervisors. In particular, the operations and procedures at the Nf-Vi reference point were designed to act upon virtual machines rather than virtualisation containers in general. According to ETSI GS NFV-INF 004 [i.4], the VIM uploads the hypervisor on the compute nodes via the Nf-Vi/C reference point and requests the creation, modification and deletion of VMs via the Nf-Vi/H reference point. Requirements on hypervisors to make them suitable for use in an NFV environment are described in ETSI GS NFV-INF 004 [i.4].

The hypervisor-based approach does not place any constraint on the type of operating system that the VNF components (VNFCs) of a VNF are using. Both the operating system and the actual network application are part of the software image delivered by the VNF provider and loaded on the VM.

4.3 OS Containers

4.3.1 Overview

Container-based virtualisation, also called operating system (OS)-level virtualisation, is an approach to virtualisation which allows multiple isolated user space instances on top of a kernel space within the OS. The isolated guests are called containers.

Figure 1 provides a high-level comparison of the software architectures for hypervisor solutions where the VNFC software image loaded in the virtualisation container includes both a guest OS kernel and the actual application, and OS container solutions where the VNFC software image loaded in the virtualisation container only includes the actual network application.

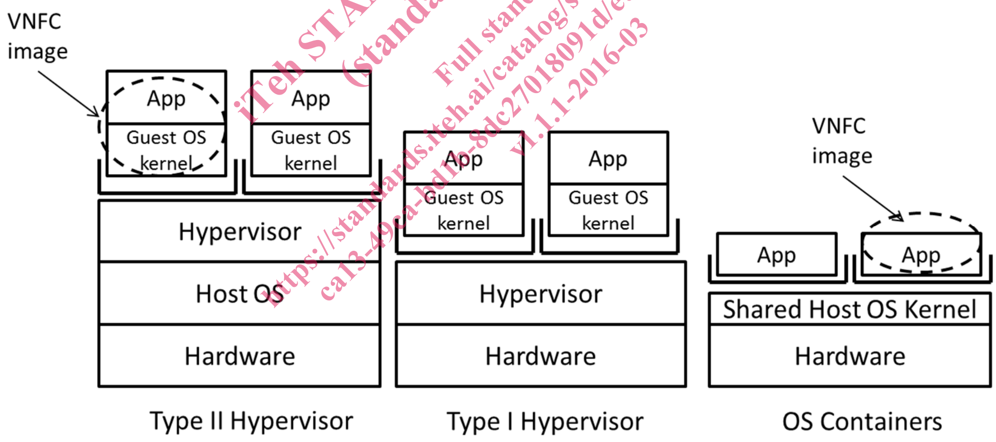

Figure 1: Hypervisor vs. OS Container solutions

The OS virtualisation technology allows partially shared execution context for different containers. Such a shared execution context is frequently referred to as a container pod. A pod might include shared file systems, shared network interfaces and other shared OS resources that are accessible from every container within that pod.

In addition to hypervisor-based execution environments that offer hardware abstraction and thread emulation services, the OS container execution environment provides kernel services as well. Kernel services include:

- Process control.

EXAMPLE 1: OS process creation; scheduling; wait and signal events; termination.

- Memory management.

EXAMPLE 2: Allocation and release of regular and large pages; handling memory-mapped objects and shared memory objects.

- File system management.
- File management.

EXAMPLE 3: Creation, removal, open, close, read and write file objects.

- Device management.

EXAMPLE 4: Request, release, configuration and access.

- Communication services.

EXAMPLE 5: Protocol stack services, channel establishment and release, PDU transmission and reception.

- System information maintenance.

EXAMPLE 6: Time and date, system and OS resource data, performance and fault indicators.

The OS container-to-VNFC logical interface is typically realized via:

- kernel system calls;
- signals to container processes;
- virtual file system mapped logical objects; and
- direct procedure calls into the container context.

OS virtualisation provides storage abstraction on file system level rather than on block device level. Each container has its separate file system view, where the guest file system is typically separated from the host file system. Containers within the same pod might share file systems where modifications made in one container are visible in the others.

Container file systems are realized either with standalone or with layered file systems. Standalone file systems are mapped into real file systems where all modifications made by the guest are stored in the backing real file system. Layered file systems take one or more base layers, and a writable overlay. A single layer is formed either from a real file system or from another layered file system structure. Layers are transparently overlaid and exposed as a single coherent file system. Typically, the lowermost layer contains an OS distribution with packages, libraries and run-times, while the overlay contains instance-specific customizations and modifications made by the container. While base layers are semi-permanently stored in image repositories, an overlay is disposable and its life time is coupled with the container life time.

4.3.2 Application to NFV

Because OS container solutions are based on a rather lightweight design where all VNFC instances share the same OS kernel, they are often considered as an alternative to hypervisor solutions when there is a need for deploying many instances of a network function (e.g. per-user instances for virtual residential gateway deployments).

4.4 Higher-level containers

4.4.1 Overview

Higher level containers are a level of virtualisation technologies more dealing with software code and its development, deployment, and runtime environment. So the level of abstraction is on the runtime environment, where source code written in a certain programming or scripting language is deployed onto the NFVI. A few characteristics of such systems are that:

1) source code is held and versioned in a code repository;

2) source code dependencies are explicitly defined and packaged into the deployed software;

3) code can be deployed into development, staging, or production environments without change;

4) configuration of the software is stored in the environment, typically through environment variables;

5) backing services such as data stores, message queues, and memory caches are accessed through a network and no distinction is made between local or third party services; and

6) processes are stateless and therefore enable easy scale-out.

Typically, those containers are used in continuous deployment models enabling fast DevOps models for telecommunication services.

4.4.2 Application to NFV

Higher-level containers are applied for VNFs that are delivered in source code and run on a particular execution engine on an NFVI. Compared to OS Containers and hypervisors, they are often considered as an alternative solution when there is a need for deploying a VNF in form of source code in so called DevOps environments.

4.5 Nesting of virtualisation technologies

4.5.1 Overview

Within the NFVI, the virtualisation layer may be composed of multiple nested sub-layers, each using a different virtualisation technology. In this case, only the top sub-layer and its technology are visible to the Virtualised Infrastructure Manager (VIM) and the partitions it creates provide the role of the virtualisation container as defined in ETSI GS NFV 003 [i.1]. Resource partitioning in the other sub-layers is typically provisioned by means outside the scope of NFV Management and Orchestration functions (e.g. by a dedicated non-NFV infrastructure OSS). An example shown in figure 2 is the case of a three levels virtualisation layer, where the top level uses a higher layer virtualisation technology, the layer below running OS container technology and the lowest layer uses the hypervisor technology. In this case, several higher-level containers, each hosting a VNFC instance, can run within each of the OS containers on one or several virtual machines created by the hypervisor.

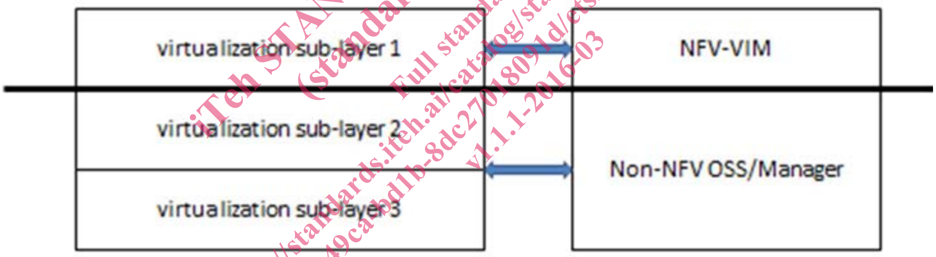
virtualization sub-layer 1
virtualization sub-layer 2
virtualization sub-layer 3
NFV-VIM
Non-NFV OSS/Manager

Figure 2: Example sub-layering of nested virtualisation technologies

This is known as recursive virtualisation in ETSI GS NFV-INF 007 [i.6], which highlights that an operating virtual functional block can itself be a host functional block.

Besides the above considerations, virtualisation technologies can also be used inside a VNFC. However, this usage results from a decision of the VNF provider and is therefore outside the scope of the NFV framework and of the present document.

4.5.2 Application to NFV

4.5.2.1 General

The primary benefit of nesting is operational, since it enables more operational flexibility in integrating NFV as part of a larger deployment.

EXAMPLE: Since NFV's primary application is in the space of telecommunications and networking, a service provider might choose to run other non-telco services and applications on the same infrastructure, but managing their life cycle manually or using a different system than NFV Management and Orchestration functions.

Another example is when due to the internal organization of a service provider, the operations are divided between different organizational structures; with nesting, the organizational divide can also be implemented in the technical space.