ETSI GR NFV-IFA 038 V4.1.1 (2021-11)

Network Functions Virtualisation (NFV) Release 4; Architectural Framework; Report on network connectivity for container-based VNF

Disclaimer

The present 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

DGR/NFV-IFA038

Keywords

container, network, NFV, VNF

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

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 prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

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

Modal verbs terminology

In the present document "**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 studies the necessary capabilities of the NFV-MANO framework to support the management of network connectivity, and its associated virtualised network resources, for container-based VNFs. It includes, among others, the support of multiple network interfaces per container, CIS cluster nodes network and external VNF connectivity.

The present document describes related use cases and provides recommendations on enhancements to the ETSI NFV specifications.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".

[i.2] ETSI GR NFV-IFA 029: "Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"".

[i.3] ETSI GS NFV-IFA 036: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Specification of requirements for the management and orchestration of container cluster nodes".

[i.4] ETSI GS NFV-IFA 040: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[i.5] IETF RFC 4271: "A Border Gateway Protocol 4 (BGP-4)".

[i.6] IETF RFC 7938: "Use of BGP for Routing in Large-Scale Data Centers".

[i.7] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

primary container cluster external network: network that is exposed external to the CIS cluster to which OS containers deployed within the CIS cluster are connected indirectly via a primary network interface, through native capabilities of the underlying container infrastructure

NOTE: Indirect connectivity is realized by connecting the OS containers of a cluster to the external network through, e.g. IP or application proxy.

primary container cluster internal network: network that is not exposed external to the CIS cluster and to which all OS containers deployed within the CIS cluster are connected through their primary network interface

secondary container cluster external network: network that is exposed external to the CIS cluster to which OS containers deployed within the CIS cluster are connected directly via additional network interfaces other than the primary network interface

NOTE: Direct connectivity is realized by connecting the OS containers of a cluster without IP proxy to the external network.

EXAMPLE: In Kubernetes®, additional network interfaces are realized through CNI™ plugins.

secondary container cluster internal network: network that is not exposed external to the CIS cluster and to which OS containers deployed within the CIS cluster are connected via an additional network interface other than their primary network interface

EXAMPLE: In Kubernetes®, additional network interfaces are realized through CNI™ plugins.

secondary network configuration profile: declarative descriptor to describe the attributes for secondary container cluster internal/external network

NOTE: In a Kubernetes® environment, a Network Attachment Definition (NAD) can be an example of SNCP.

3.2 Symbols

Void.

3.3 Abbreviations

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

ARP	Address Resolution Protocol
CISI	CIS Instance
CNI™	Container Network Interface
CNM	Container Network Management
CRD	Custom Resource Definition
eBGP	external BGP
iBGP	internal BGP
IPAM	IP Address Management
NAD	Network Attachment Definition
ND	Neighbour Discovery
RR	Route Reflector
SNCP	Secondary Network Configuration Profile
veth	virtual ethernet

4 General

4.1 Background

ETSI GR NFV-IFA 029 [i.2] identifies the Container Network Management (CNM), a functionality of the CISM to manage container network resources provided by the CIS via an interface. ETSI GS NFV-IFA 040 [i.4] specifies the requirements of OS container network management service interface produced by the CISM accordingly. This information provides the base for further investigation in the direction of the management of network connectivity and its associated network resources for container-based VNFs.

4.2 Introduction

The present document investigates aspects of network connectivity for container-based VNF. It starts with the study on the following main technical scenarios:

- Network connectivity for OS containers among VNFs within the same or different NFVI-PoPs, e.g. network provisioning and reconfiguration.
- Network connectivity for OS containers in VNFs within the same CIS cluster node as specified in ETSI GS NFV-IFA 036 [i.3], or across different CIS cluster nodes.
- Network connectivity between OS containers through multiple network interfaces, e.g. to support separating VNF traffics concerning multiple network planes.

4.3 Network connectivity for OS containers among VNFs

In an NFV system, an NS can consist of multiple VNFs. A VNF can be container-based, or VM-based, or hybrid (i.e. with some VNFCs implemented in containers and others in VMs). Furthermore, both the OS containers in different container-based VNFs and the VMs in VM-based VNFs are expected to communicate with each other independently of the type of VNF, as shown in figure 4.3-1, irrespective of whether they are referenced by the same or different NS. NFV-MANO can establish the network connectivity among the OS containers or among the OS containers and the VMs.

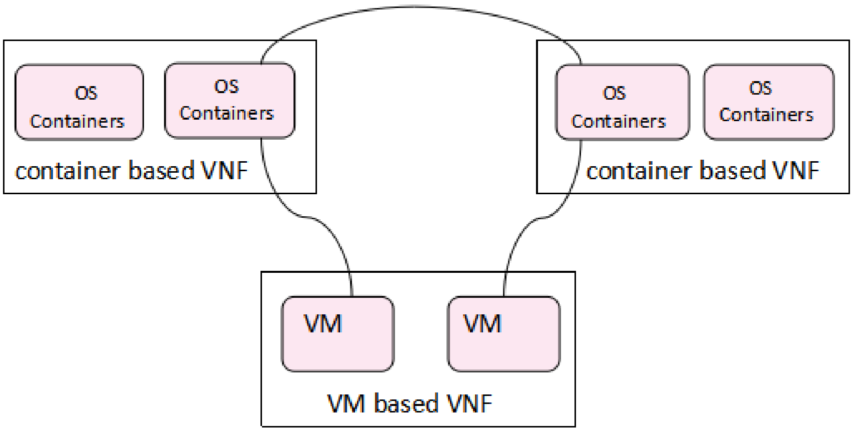


Figure 4.3-1: Example of network connectivity for OS containers among VNFs

4.4 Network connectivity for OS containers among CIS cluster nodes

A CIS cluster node can have one or more network interfaces for cluster interaction. When the container-based VNFs are instantiated, OS containers are deployed in CIS cluster nodes. A container-based VNF can be deployed on one or multiple CIS cluster nodes.

In case the OS containers of one VNF are deployed on multiple CIS cluster nodes, it is expected that inter-node networks enable the network connectivity for multiple CIS cluster nodes. Figure 4.4-1 illustrates an example where the network connectivity of the OS containers in CIS cluster node 1 and the OS containers in CIS cluster node 2 is created through virtual networks (e.g. vSwitch, IP route, IP VLAN, MAC VLAN, Host network) residing in the CIS cluster nodes and their NICs.

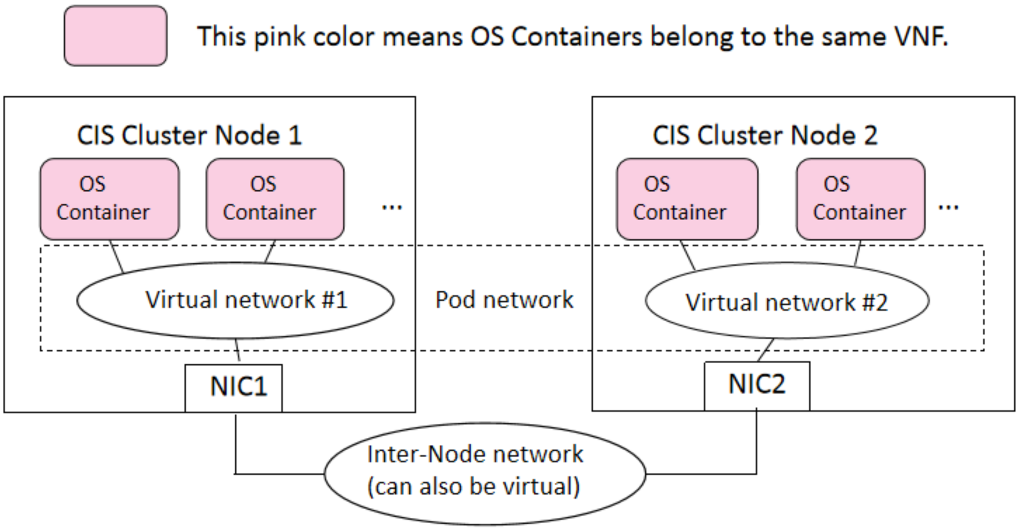


Figure 4.4-1: Example of network connectivity for OS containers among CIS cluster nodes

4.5 Network connectivity for OS containers attaching multiple networks

Multiple network planes, such as control, data and management, are typically realized by different networks which host different types of associated control, data or management traffic.

CIS cluster nodes can have one or multiple network interfaces which are used for the connection between OS containers and to support the attachment to multiple networks.

In particular, platforms like Kubernetes® assume that each Pod has a unique, routable IP address inside the CIS cluster. Typically, in Kubernetes® each Pod only has one network interface. However, when the OS container technology is applied to telecommunication networks, like 5G network, additional requirements are expected to be supported. For instance, enabling multiple network interfaces per Pod would support differentiated network service requirements, such as high QoS, low latency, high bandwidth, and other network characteristics.

NOTE: In the present clause and the rest of the present document, Pod is an example of Compute MCIO as specified in ETSI GS NFV-IFA 040 [i.4] and minimum granularity of managed object accommodating one or more OS containers in Kubernetes® environment.

As depicted in figure 4.5-1, OS containers which belong to one container-based VNF can attach to multiple networks of the VNF according to network descriptions in the VNFD. Multiple networks can be associated to multiple types of network planes within a VNF, such as management, signalling and data, and also to external networks. The VNF external networks are used to connect the VNF with other container-based VNFs, VM-based VNFs and PNFs via the external network interface. Similarly, the VNF internal networks are used to connect OS containers within the same VNF.

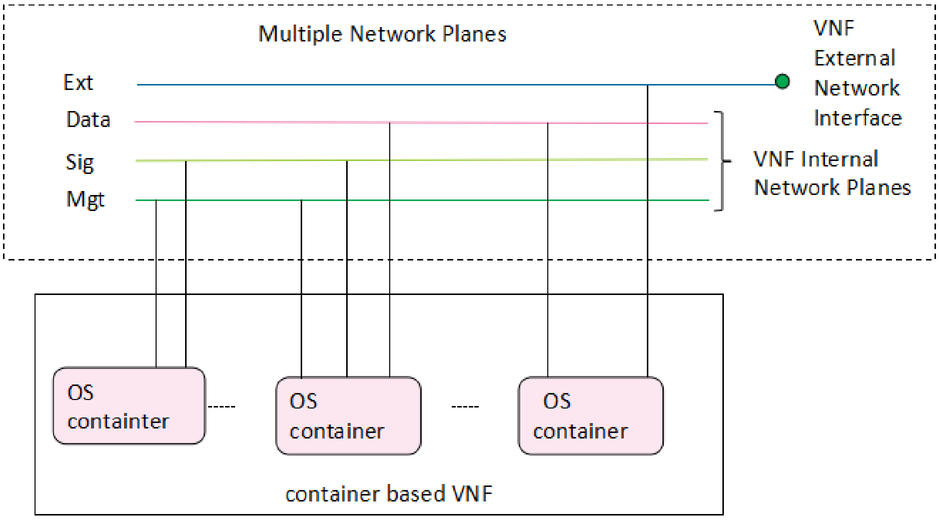


Figure 4.5-1: Example of network connectivity for OS containers attaching multiple network planes

4.6 Network connectivity for VNFs using load balancer

Network connectivity between container-based VNF instances, or container-based VNF instance and VM-based VNF instances can be setup using load balancing. A load balancer is generally used for distributing traffic among different VNFs; the load balancer accepts incoming traffic and distributes it among different VNF instances as per the load balancing algorithm.

Following are some of the use cases for load balancing based connectivity:

- Single public IP address to reach multiple services.
- Load balancing traffic among VNF instances placed in different clusters.
- Blue-Green deployments.
- High availability.
- Reverse proxy, handling specific security configurations.
- Abstraction to backend details which makes backend scalable/flexible.

In case container-based VNF instances, or container-based VNF instance and VM-based VNF instances are deployed in different nodes, a load balancer is used as traffic manager sitting in front of backend VNFs to route incoming traffic among them sent from client VNF.

For the actual steps of connectivity, first the client VNF resolves the load balancer FQDN using DNS and forwards the packets to the load balancer IP address. The load balancer in turn routes incoming packets to CIS cluster nodes from where packets are forwarded to specific VNF.

Both client as well as backend VNFs can be container-based or VM-based VNFs.

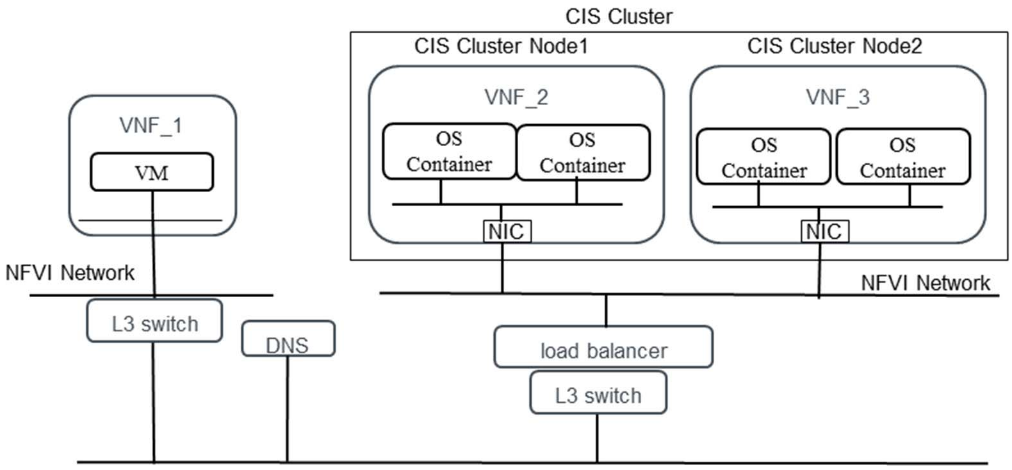


Figure 4.6-1: Example of connectivity between VNFs using load balancer

5 Use cases

5.1 Network connectivity for OS containers among VNFs

5.1.1 General

Clause 5.1 of the present document describes how to create network connectivity for container-based VNFs during NS LCM and VNF LCM operations.

5.1.2 Actors and roles

Table 5.1.2-1 defines the actors including the NFV-MANO entities, such as the NFVO, VNFM, CISM, and VIM and describes their roles for the case of managing the NS that consists of container-based VNFs and VM-based VNFs in different NFVI-PoPs.

NOTE: A solution for the network connectivity between container-based VNF and VM-based VNF is out of the scope of the present document.

Table 5.1.2-1: Network connectivity for OS containers among VNFs actors and roles

#	Role	Description
1	OSS/BSS	Request the operations of NS lifecycle management.
2	NFVO	Responsible for NS lifecycle management.
3	VNFM	Responsible for VNF lifecycle management.
4	CISM	Responsible for the lifecycle management of containerized workloads.
5	VIM	Allocate VM related virtual resources, including virtual network resources.

5.1.3 Network connectivity creation during NS LCM operations

5.1.3.1 Introduction

This use case describes how to create network connectivity between container-based VNF instances, or between container-based VNF instances and VM-based VNF instances in the same NS instance while performing NS LCM operations. Specifically, in this use case, an NS might be comprised of container-based VNFs which can be in the same NFVI-PoP, or in different NFVI-PoPs. The network connectivity requirements of the NS are described by the VLDs and CPDs in the NSD and the VNFDs.

5.1.3.2 Pre-conditions

Table 5.1.3.2-1 describes the use case pre-conditions.

Table 5.1.3.2-1: Network connectivity creation during NS LCM operations pre-conditions

#	Pre-condition	Additional Description
#1	NSD and VNF packages, used for the instantiation of the NS composed of container-based VNFs and VM-based VNFs, are onboarded.	

5.1.3.3 Post conditions

Table 5.1.3.3-1 describes the use case post-conditions.

Table 5.1.3.3-1: Network connectivity creation during NS LCM operations post-conditions

#	Post-condition	Additional Description
#1	NS instance is created successfully, and the creation of network connectivity for container-based VNF and VM-based VNF is completed, for VNFs within or across NFVI-PoPs.	

5.1.3.4 Flow description

Table 5.1.3.4-1 describes the flow of information in this use case.

Table 5.1.3.4-1: Network connectivity creation during NS LCM operations flow description

#	Actor/Role	Description
Begins when	OSS/BSS	The OSS/BSS requests the NFVO to perform NS instantiation or VNF instantiation as part of NS update.
Step 1	NFVO->CISM NFVO->VIM	Based on the NS VL connectivity requirements expressed in the NSD (e.g. specific connectivity across VM-based VNFs and container-based VNFs), the NFVO determines the configuration of the network resources (includes the creation of network resources for VM-based VNF connectivity) by sending requests to the VIM in case of VM-based VNFs or the CISM in case of container-based VNFs. Based on the internal VNF connectivity requirements expressed in the VNFDs, the NFVO determines the configuration of the network resources by sending requests to the VIM or the CISM. This assumes the case of provisioning of externally managed VLs.
Step 2	CISM->NFVO VIM->NFVO	The VIM or the CISM inform the NFVO that the requested network resources have been configured.
Step 3	NFVO->VNFM	The NFVO requests the VNFM to initiate VNF instantiation for constituent VNFs of the NS. The NFVO includes references to the network resources to be used by the container-based VNF as internal VNF VLs (these are externally managed VLs). The VNFM fetches the VNFD and creates the "Individual VNF instance" resource.
Step 4	VNFM->CISM VNFM->VIM	In case of container-based VNF instantiation, the VNFM requests the CISM to provision resources for the containerized workloads of the container-based VNF. The CISM allocates the resources for the containerized workloads. In case of VM based VNF instantiation, the VNFM requests the VIM (in case of indirect mode, the VNFM will interact with the VIM via the NFVO) to provision virtualised resources for the VM based VNF. The VIM allocates the virtualised resources for the VNF.
Step 5	CISM->VNFM VIM->VNFM	CISM/VIM inform the VNFM that VNF related virtualised resources have been allocated successfully.
Step 6	VNFM->NFVO	Upon completion of the VNF instantiation by the VNFM, the VNFM sends to the NFVO a VnfLcmOperationOccurrenceNotification to indicate that the VNF LCM operation occurrence has been "COMPLETED".

#	Actor/Role	Description
Ends when	NFVO	Upon completion of the creation of all involved VNF instances and network connectivity needed for the NS, the NFVO sends to the OSS/BSS an NsLcmOperationOccurrenceNotification to indicate that the NS LCM operation has been "COMPLETED".

5.1.4 Modify network connectivity during NS update operations

5.1.4.1 Introduction

This use case describes how to modify the network connectivity for container-based VNF instances and VM-based VNF instances in an NS instance during Update NS operation with the option of ChangeExtVnfConnectivity.

5.1.4.2 Pre-conditions

Table 5.1.4.2-1 describes the use case pre-conditions.

Table 5.1.4.2-1: Modify network connectivity during NS update operations pre-conditions

#	Pre-condition	Additional Description
#1	The VNF and NS Descriptors describe the relevant networking information for container-based VNFs and VM-based VNFs. The requirements of the new external connectivity for the VNFs is described in the NSD.	
#2	NS and VNF packages used for the NS composed of container-based VNFs and VM-based VNFs are onboarded.	
#3	The VNF instances for which a change of external connectivity is to be requested are in INSTANTIATED state.	

5.1.4.3 Post conditions

Table 5.1.4.3-1 describes the use case post-conditions.

Table 5.1.4.3-1: Modify network connectivity during NS update operations post-conditions

#	Post-condition	Additional Description
#1	The change of external VNF connectivity for container-based VNF instance(s) and VM-based VNF instance(s) in the NS instance is successfully completed.	

5.1.4.4 Flow description

Table 5.1.4.4-1 describes the flow of information in this use case.

Table 5.1.4.4-1: Modify network connectivity during NS update operations flow

#	Actor/Role	Description
Begins when	OSS/BSS	The OSS/BSS requests the NFVO to initiate the NS update operation for changing the external VNF connectivity of VNF instances belonging to the NS instance. Based on the information contained in the NSD, the current information and state of the NS instance and the NS update request with external VNF connectivity data, the NFVO determines the VNF instances that require changes to the external connectivity.
Step 1	NFVO -> VNFM	The NFVO requests the VNFM to modify the external connectivity of the VNF instance.

#	Actor/Role	Description
Step 2	VNFM->CISM; VNFM->VIM	For container-based VNF instances, the VNFM requests the CISM to connect/disconnect the network connectivity with the external networks. For VM-based VNFs, the VNFM disconnects/connects the network connectivity of the VNF instance to the target external virtual links by issuing the corresponding virtualised resource management requests to the VIM.
Step 3	CISM->VNFM; VIM -> VNFM	For container-based VNF instances, the CISM disconnects the appropriate groups of one or more OS container with connectivity to the external network from their existing external network to the new external network. The CISM informs the VNFM that network connectivity has been changed per the request. For VM-based VNFs, the VIM completes the requested modifications and informs the VNFM.
Step 4	NFVO->CISM; NFVO->VIM	Based on the information in the NSD and outcomes from the VNF external connectivity change, the NFVO requests the CISM to terminate unused virtualised resources for external network connectivity associated with the container-based VNF. The NFVO requests the VIM to terminate unused external networks and/or link ports associated with the VM-based VNF.
Step 5	CISM->NFVO; VIM->NFVO	The CISM/VIM informs the NFVO that the changes of virtualised resources supporting the external connectivity have been completed and the unused resources have been successfully deleted.
Step 6	VNFM->NFVO	Upon completion of change external VNF connectivity by the VNFM, the VNFM informs the NFVO that the VNF LCM operation occurrence has been completed.
Ends when	NFVO	Upon completion of change external VNF connectivity for the VNF instances of the NS instance, the NFVO informs the OSS/BSS that the NS update operation has been completed and includes the list of affected instances during this operation.

5.2 Network connectivity of OS containers within or across CIS cluster nodes

5.2.1 General

Clause 5.2 describes a use case about creating network connectivity for groups of one or more OS container within the same CIS cluster node or across CIS cluster nodes during LCM operations on containerized VNFs.

5.2.2 Actors and roles

Table 5.2.2-1 defines the actors and describes their roles including the NFV-MANO functional blocks, such as the NFVO, VNFM, NFV-MANO functions such as the CISM, and the CIS Instance (CISI) for the case of the creation of the OS container network connectivity.

Table 5.2.2-1: Network connectivity of OS containers within or across CIS cluster nodes actors and roles

#	Role	Description
1	NFVO	Initiate the instantiation operation of the container-based VNF.
2	VNFM	Responsible for performing the lifecycle management of container-based VNFs.
3	CISM	Responsible for the lifecycle management of containerized workload.
4	CCM	Responsible for the management of CIS clusters, see ETSI GS NFV-IFA 036 [i.3].
5	CISI	Create network connectivity for the groups of one or more OS container on the container run-time environment.

5.2.3 Network connection creation during containerized workload LCM operations

5.2.3.1 Introduction

This use case describes how to create network connectivity for groups of one or more OS container inside a CIS cluster node or across CIS cluster nodes for the MCIOs when performing the container-based VNF LCM operation. Specifically, when infrastructure resources for the groups of one or more OS container are allocated, the groups of one or more OS containers which belong to the same container-based VNF instance will be created together with the network connectivity for the groups of one or more OS container. The groups of one or more OS containers can be deployed on a same CIS cluster node or on different CIS cluster nodes.

5.2.3.2 Pre-conditions

Table 5.2.3.2-1 describes the use case pre-conditions.

Table 5.2.3.2-1: Network connection creation during containerized workload LCM operations pre-conditions

#	Pre-condition	Additional Description
#1	The CCM provides the cluster node resources for the CISM to perform the containerized workload LCM operation.	
#2	The NFVO requests the VNF LCM operations for the container-based VNF.	

5.2.3.3 Post conditions

Table 5.2.3.3-1 describes the use case post-conditions.

Table 5.2.3.3-1: Network connection creation during containerized workload LCM operations post-conditions

#	Post-condition	Additional Description
#1	Completion of the VNF LCM operations for the container-based VNF	

5.2.3.4 Flow description

Table 5.2.3.4-1 describes the flow of information in this use case.

Table 5.2.3.4-1: Network connection creation during containerized workload LCM operations flow description

#	Actor/Role	Description
Begins when	NFVO	The NFVO initiates a VNF instantiation operation.
Step 1	NFVO->CISM	Based on information in the NSD and the VNFD the NFVO determines the need to configure multiple secondary container cluster networks and requests the CISM to configure them.
Step 2	CISM->NFVO	The CISM informs the NFVO that the requested network resources have been configured.
Step 3	NFVO -> VNFM	The NFVO requests the VNFM to perform the container-based VNF instantiation. The NFVO provides references to VNF external and VNF internal networks (the latter as externally managed VLs). The VNFM fetches the VNFD and initiates the VNF instantiation process.
Step 4	VNFM->CISM	The VNFM requests CISM to create MCIOs based on the information of MCIOPs, see ETSI GS NFV-IFA 040 [i.4].

#	Actor/Role	Description
Step 5	CISM->CISI	The CISM sends commands to CISI to create network connectivity for the compute resources and storage resources for MCIOs. If there is only a primary container cluster internal network, for VNF internal network connectivity, there is no further information in the MCIOP. Otherwise, the CISM sends the OS container's network connectivity information (see note) for MCIOs to the CISI, then the CISI connects the group of one or more OS container within the same CIS cluster node or across CIS cluster nodes.
Step 6	CISI->CISM	The CISI responds to CISM that the resources for MCIOs have been successfully connected.
Step 7	CISM->VNFM	Upon completion of the MCIO creation, the CISM returns the VNFM a notification.
Step 8	VNFM->NFVO	Upon completion of the MCIO creation and network connectivity among the groups of one or more OS container, the VNFM completes the VNF instantiation, and sends to the NFVO a notification that the VNF LCM operation occurrence has been "COMPLETED".
Ends when	NFVO	The NFVO is notified that the instantiation of container-based VNF is completed.
NOTE: This information is used to describe how to map with external network if there is only a primary container cluster external network. And this information is also used to describe the requirements of multiple networks associated to the VNF. For example, in a Kubernetes® environment a Pod network can be realized via secondary container cluster external networks, defined by including attributes of multiple networks.		

5.3 Network connectivity between groups of one or more OS container and multiple networks

5.3.1 General

This clause describes how to create multiple networks for the groups of one or more OS container inside the container-based VNF instance, and how the groups of one or more OS container belonging to the VNF are connected to multiple networks.

5.3.2 Actors and roles

Table 5.3.2-1 defines the actors and describes their roles including the NFV-MANO functional blocks, such as NFVO, VNFM, NFV-MANO functions such as the CISM and CISI for the case of network connectivity between groups of one or more OS container and multiple networks during container-based VNF instantiation procedure.

Table 5.3.2-1: Network connectivity between groups of one or more OS container and multiple networks actors and roles

#	Role	Description
1	NFVO	Initiate the instantiation operation of the container-based VNF.
2	VNFM	Responsible for performing the lifecycle management of container-based VNFs.
3	CISM	Responsible for the lifecycle management of containerized workload.
4	CISI	Create network connectivity for the resources for groups of one or more OS container on the container run-time environment.

5.3.3 Network connection creation for groups of one or more OS container and multiple networks

5.3.3.1 Introduction

This use case describes how to create MCIOs, multiple networks and network connectivity for the groups of one or more OS container during the instantiation procedure of the container-based VNF.

During the VNF instantiation procedure, network resources including multiple networks, ports to implement network connectivity between the groups of one or more OS container and multiple networks inside the same VNF instance are created. Groups of one or more OS container of the same VNF instance can be connected to each other through the created networks.

5.3.3.2 Pre-conditions

Table 5.3.3.2-1 describes the use case pre-conditions.

Table 5.3.3.2-1: Network connection creation for groups of one or more OS container and multiple networks pre-conditions

#	Pre-condition	Additional Description
#1	The attributes of multiple networks for MCIOs are defined in VNFD/MCIOP.	.
#2	VNF package is onboarded to the NFVO and contains VNFD and MCIOPs.	

5.3.3.3 Post conditions

Table 5.3.3.3-1 describes the use case post-conditions.

Table 5.3.3.3-1: Network connection creation for groups of one or more OS container and multiple networks post-conditions

#	Post-condition	Additional Description
#1	Completion of the container-based VNF instantiation	

5.3.3.4 Flow description

Table 5.3.3.4-1 describes the flow of information in this use case.

Table 5.3.3.4-1: Network connection creation for groups of one or more OS container and multiple networks flow description

#	Actor/Role	Description
Begins when	NFVO	The NFVO initiates a VNF instantiation operation.
Step 1	NFVO->CISM	Based on information in the VNFD the NFVO determines the need to configure secondary container cluster networks and requests the CISM to configure them.
Step 2	CISM->NFVO	The CISM informs the NFVO that the requested network resources have been configured.
Step 3	NFVO->VNFM	The NFVO requests the VNFM to perform a VNF instantiation operation for a container-based VNF. The NFVO provides references to VNF external and VNF internal networks (the latter as externally managed VLs).
Step 4	VNFM -> CISM	The VNFM requests the CISM to create MCIOs based on the VNFD/MCIOP.

#	Actor/Role	Description
Step 5	CISM<->CISI	The CISM sends commands to CISI, requesting to create connectivity for the groups of one or more OS container, which are constituents of the container-based VNF instance. The groups of one or more OS container can be connected to these primary and secondary networks in a variety of routing ways, e.g. bridge.
Step 6	CISM->VNFM	Upon completion of the MCIO creation, the CISM returns the VNFM a notification.
Step 7	VNFM->NFVO	Upon completion of the MCIO creation and network connectivity among them, the VNFM completes the VNF instantiation, and sends to the NFVO a notification that the VNF LCM operation occurrence has been "COMPLETED".
Ends when	NFVO	The NFVO is notified that the instantiation of container-based VNF is completed.

6 Potential Solutions

6.1 Overview

Clause 6 documents various potential solutions related to network connectivity for container-based VNFs, including solutions to support multiple networks interfaces per group of one or more OS container with regards to the interworking of NFV-MANO functional blocks. The set of solutions described in clause 6 of the present document can be categorized into three technical areas:

Clauses 6.2 to 6.6 describe various networking technologies that can be used to enable inter-node (CIS cluster node) network connectivity for groups of one or more OS container (e.g. Pods) deployed on different CIS cluster nodes.

- Clause 6.2 describes overlay technologies.
- Clause 6.3 describes routing technologies.
- Clause 6.4 describes L2 underlay technologies.
- Clause 6.5 describes direct mode connectivity between groups of one or more OS container (e.g. Pods) and inter CIS cluster node networks.
- Clause 6.6 describes load balancer technologies.

Clauses 6.7 and 6.8 describe solutions related to the support of multiple networks interface connectivity for groups of one or more OS container.

Finally, clauses 6.9 to 6.11 describe various procedures for setting up the network connectivity for groups of one or more OS container (including multiple networks interface support) considering the interworking of NFV-MANO functional blocks, the CISM and CCM functions and relevant artefacts, descriptors and managed resources, such as virtualised network resources.

6.2 Inter-node network with overlay technologies

As described in the clause 4.4, in a Kubernetes® environment, a Pod, a group of one or more OS container within the same cluster node can share the same network interface. The IP address allocated to the network interface is shared by the OS containers in the Pod. In this context, the OS containers in the Pod communicate with each other using an internal communication method.

As shown in figure 6.2-1, a pair of Pods in a single CIS cluster node are attached to a virtual network (instantiated by a virtual bridge). Pod IP addresses are internally available in a CIS cluster node, so those Pods are reachable via the virtual network.

The Pod IP addresses cannot be used to route traffic between OS containers in different CIS cluster nodes, therefore, another method is used for the OS containers to communicate with each other.

A solution in such a case is to have connectivity between all CIS cluster nodes in a CIS cluster. A tunnel end point in each CIS cluster node (e.g. VXLAN, IP in IP, etc.) is attached to the virtual network. As a result, the overlay tunnel interconnects the virtual networks on each CIS cluster node and it seems as if the all Pods were on the same virtual network. Pod IP addresses are assigned from the subnet mapped on the interconnected virtual network. Pods are running on the different CIS cluster nodes, so an IP address management solution applicable to a distributed address allocation scheme becomes relevant. On a source node, an OS container originated packet destined for another OS container running on a different CIS cluster node in the same CIS cluster is forwarded to the tunnel end point. The packet is then encapsulated into the external IP packet. The IP address assigned to the network interface connecting to the NFVI network is used as the source address in the external header. The IP address of the CIS cluster node hosting the target OS container is used as the destination address.

This way the encapsulated packet is transferred from source CIS cluster node to destination CIS cluster node just like normal node to node traffic.

On the destination CIS cluster node, the packet is de-capsulated at the tunnel end point to restore the original source packet which is then delivered to the destination Pod through virtual bridge interface.

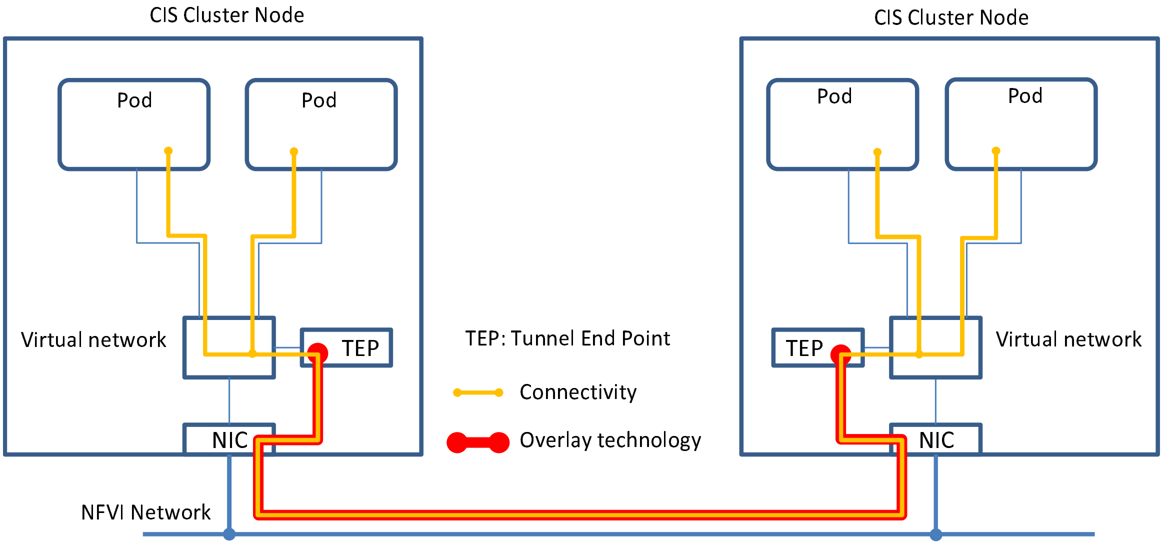


Figure 6.2-1: Connectivity between Pods over overlay technology

6.3 Inter-node network with routing

6.3.1 Overview

OS containers communicate with each other both within a CIS cluster node and between different CIS cluster nodes. In the second case the reachability information should be available at each CIS cluster node. Routing mechanisms can be employed for resolving such connectivity and enable traffic between OS containers on different CIS cluster nodes or even networks.

6.3.2 Route sharing with BGP

On the Internet, BGP [i.5] is a well-known control plane protocol to exchange network reachability information with other BGP systems.

Internal BGP (iBGP) is used by peers in an autonomous system. Each BGP peer is connected with a full-mesh topology and exchanges routing information over node to node network connection. The destination OS container address, therefore, can be resolved by using network reachability information.

However, a full-mesh topology is not scalable. To get rid of the full-mesh topology among iBGP peers, Route Reflectors (RRs) can be used. In an autonomous system, a network node with Route Reflector feature connects to each iBGP peer using a hub & spoke topology. The Route Reflector then transfers network reachability information to each iBGP peer.

On the other hand, External BGP (eBGP) is used by peers among different autonomous systems. A pair of boarder routers are connected at the edges of the autonomous systems to exchange network reachability information over this connection.

6.3.3 Inter-node network with BGP routing

Route sharing with BGP is, for example, used in open source software (e.g. Calico). A CIS cluster node establishes a BGP session with another CIS cluster node. The IP subnets used for interconnecting group of one or more OS container (e.g. Pods) on each CIS cluster node are exchanged with each other so that OS containers running on both CIS cluster nodes become reachable.

A solution in such a case is to exchange network reachability information through NFVI technologies [i.6]. As shown in figure 6.3.3-1, a pair of NFVI environments belonging to different autonomous systems are interconnected. In each NFVI environment, a CIS cluster node is attached to a NFVI network node (e.g. leaf switch), the NFVI network node is further connected to the NFVI network configured by other NFVI network nodes (e.g. spine switches).

On each autonomous system, iBGP runs on both CIS cluster node and NFVI network node to exchange network reachability information. Route Reflector can be instantiated on the NFVI network node if there are many CIS cluster nodes connected to the NFVI network node. The Route Reflector is then used to exchange network reachability information among iBGP peers running on different CIS cluster nodes.

On each autonomous system, eBGP peer runs on NFVI network node. The pair of eBGP peers running on different autonomous system boundaries establish a peering session to exchange network reachability information among such autonomous systems.

By exchanging network reachability information over iBGP/eBGP, the target destination IPs can be resolved in case the groups of one or more OS container (e.g. Pods) to be interconnected are located on different CIS cluster node spread over different autonomous systems.

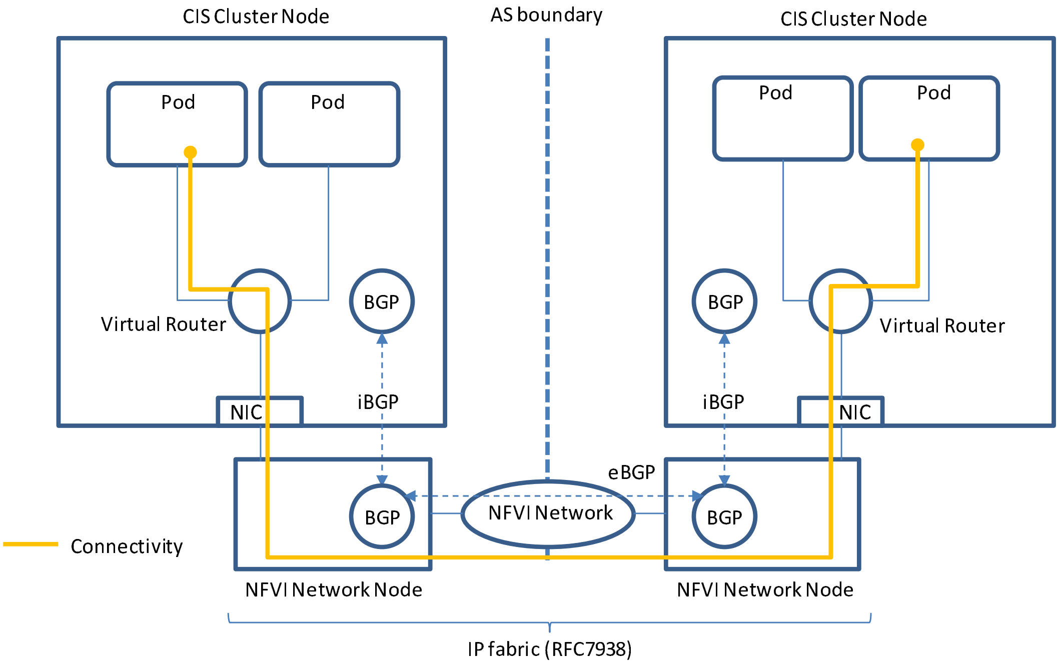


Figure 6.3.3-1: Connectivity between Pods over iBGP/eBGP technologies in NFVI network nodes

6.4 Inter-node network connectivity with L2 underlay

Clause 6.3.3 provides a solution for inter-node network connectivity for groups of one or more OS container (e.g. Pods) by using BGP when the CIS cluster nodes belong to different NFVI environments.

The L2 underlay solution provided in this clause focuses on the case that the CIS cluster nodes are located in the same NFVI environment. As shown in figure 6.4-1, in this case, the 2 CIS cluster nodes are connected in the same L2 network (the NFVI network), e.g. same L2 physical network or same VLAN. The IP address assigned for each Pod on different CIS cluster nodes cannot conflict. Since the CIS cluster nodes are connected in the same L2 network, the L2 reachability information is shared among all the CIS cluster nodes (e.g. through broadcast the ARP/ND request to the whole network). The connectivity among Pods instantiated on different CIS cluster nodes is achieved via the Pod IP address through the virtual network inside the CIS cluster node and the NFVI network among the CIS cluster nodes.

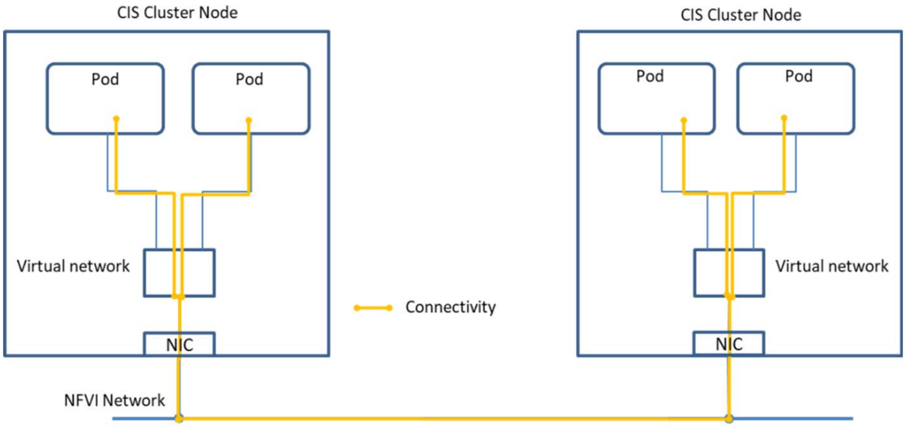


Figure 6.4-1: Connectivity between Pods instantiated on different CIS cluster nodes with L2 underlay technology

6.5 Inter-node network connectivity with direct mode

As described in clause 4.5, multiple network interface can be supported within a single group of one or more OS container (e.g. Pod), for example through CNI™ plugins to create the secondary container cluster external network. As shown in figure 6.5-1, 2 Pods are hosted in a single CIS cluster node. In this use case, additional network interface is created at each Pod to establish connectivity among Pods instantiated on different CIS cluster nodes, shown as yellow line in figure 6.5-1.

The solution provided in this use case is the direct mode that allows to move the specified network device from the host OS network namespace into the OS namespace the Pod is running in. By using the direct mode, a dedicated NIC (physical NIC or virtualised NIC) of the CIS cluster node is only used by one Pod through the additional network interface. The connectivity among Pods instantiated on different CIS cluster nodes is achieved by establishing connection through NFVI network between NICs hosted on different CIS cluster nodes.

The data transmission efficiency is high in this mode, since the Pod is directly connected to the NIC of the CIS cluster node, it does not require data processing through additional network device, e.g. virtual router, vSwitch.

While the number of NICs in a CIS cluster node is limited, this mode does not support large amount of Pods in one CIS cluster node.

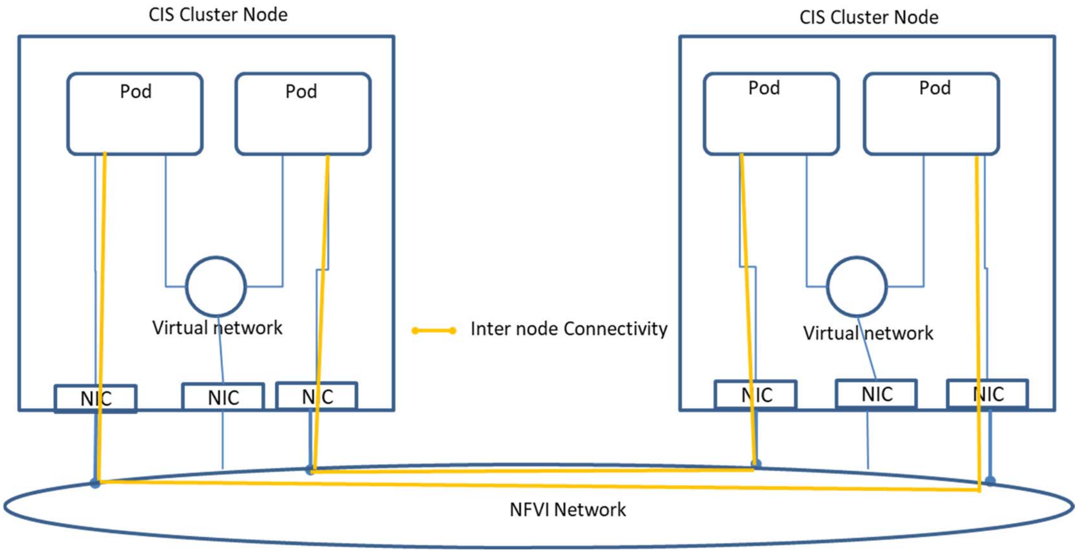


Figure 6.5-1: Connectivity between Pods instantiated on different CIS cluster nodes with direct mode technology

6.6 Network connectivity for VNFs using load balancer

As mentioned in clause 4.6, network connectivity between container-based VNF instances, or container-based VNF instance and VM based VNF instances can be setup using load balancer.

The target VNF is exposed on an external network to access it using a load balancer. The VNF can be exposed externally, for example, using service with LoadBalancer type in Kubernetes® cluster and using self-service or provided network connected with router in an OpenStack® cluster.

In case of Kubernetes® cluster with cloud provider controller, launching a Kubernetes® service with LoadBalancer type automatically creates a load balancer or updates the configuration of an existing load balancer for target endpoints (i.e. target VNFs) as per specifications of the Kubernetes® service.

So, the Kubernetes® service with LoadBalancer type maps to the load balancer and the IP address used by the load balancer is allocated as an external-IP address to this Kubernetes® service.

Refer to figure 6.6-1.

In case of Kubernetes® cluster with cloud provider controller, load balancer related configurations are applied in underlying cloud environment. Load balancer configurations includes flavour specifying resource requirements for load balancer, network related configurations, storage related configurations, load balancer topology, etc.

In case of Kubernetes® cluster with cloud provider controller, load balancer setup can be triggered in one of the following ways:

1) separate load balancer for each service created in Kubernetes®;
2) separate load balancer for each ingress resource created in Kubernetes®.

In both of these cases, load balancer resource requirements are considered as part of grant request while deploying Kubernetes® service or ingress resource.

The case of hardware-based load balancer and associated resources is not analysed in the present document.

NOTE 1: Cloud provider controller is Kubernetes® control plane component which facilitates interoperability between Kubernetes® and underlying cloud provider.

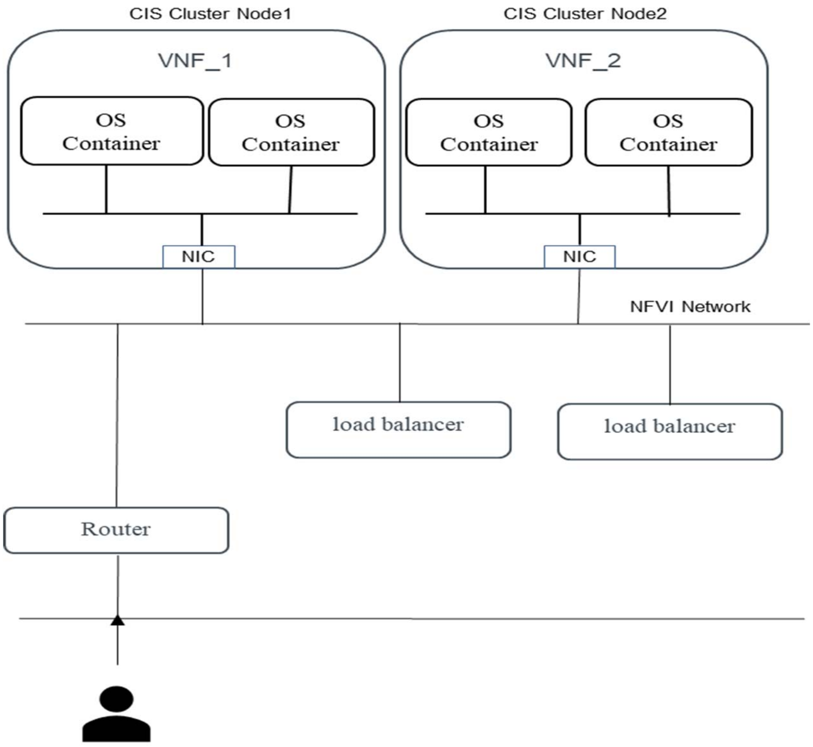


Figure 6.6-1: Example of network connectivity between VNFs using load balancer

In case of a Kubernetes® cluster without cloud provider controller, the load balancer is not launched automatically and the load balancer can be deployed as a separate VNF.

Refer to figure 6.6-2.

Some of the use cases supported by the load balancer deployed as a separate VNF include:

- to load balance traffic among multiple VNFs. It will not be possible to route traffic among different VNFs if the load balancer is not deployed externally as a separate VNF;
- to support Blue-Green deployment.

For different versions of an application deployed as a VNF, the load balancer is deployed as a separate VNF to support load balancing traffic as per Blue-Green deployment requirements (phase in, phase out).

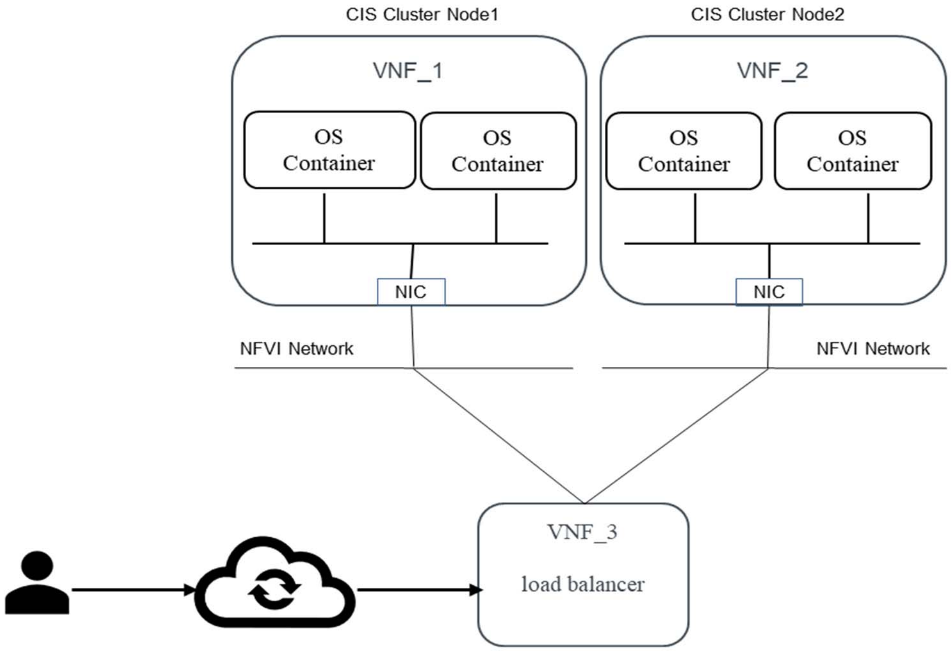


Figure 6.6-2: Network connectivity for load balancer and CIS cluster deployed as separate VNFs

For the actual steps of connectivity, user forwards packets to load balancer using the IP address allocated to the load balancer. The load balancer in turn routes incoming packets to CIS Cluster nodes from where packets are forwarded to specific VNF instances.

NOTE 2: This solution is used for primary container cluster external network.

6.7 Pod networks creation for multiple networks

Clause 5.3 describes how to create network connectivity between groups of one or more OS container and multiple networks.

When the CISM functionality is provided by Kubernetes®, this is achieved by creating Pod networks, which are a virtual networks used for network connectivity among Pods inside or across CIS cluster nodes. The Pod network can be a layer2 (switching) or layer3 (routing) virtual network.

For Kubernetes® networking, Kubernetes® CNI™ plugins can be used for creating the secondary container cluster internal networks and the secondary container cluster external networks. CNI™ plugins include main CNI™ plugins, IPAM CNI™ plugins, and special CNI™ plugins, e.g. flannel.

In case that secondary container cluster internal networks and the secondary container cluster external networks are used for Pod communication, the process of creating the Pod networks is mainly divided into three steps:

- The secondary container cluster internal networks and the secondary container cluster external networks are configured before instantiating the VNF. The network resources of Kubernetes® (e.g. network CRD) are used for setting up the secondary container cluster internal networks and the secondary container cluster external networks. Secondary Network Configuration Profiles (SNCPs) are used to describe the network attributes of secondary networks.

- When a Pod is created, Kubernetes Runtime creates a network namespace and binds it with the Pod.

- According to Pod network configuration information, the CNI™ plug-in creates the network interface for the Pod (e.g. some CNI^{TM} creates a veth pair with one interface on the Pod side and another one a virtual bridge in the CIS cluster node), checks IPAM types and data, triggers the IPAM plug-in to get the idle IP addresses, and assigns the IP addresses from the configured Pod network subnets. With this final step, the Pod is connected to the secondary container cluster internal networks and the secondary container cluster external networks as Pod networks.

NOTE: The solution in the present clause only considers Pod connectivity, and it does not describe usage of Pod connectivity for internal VNF connectivity or connectivity in between VNFs.

6.8 Multiple networks for groups of one or more OS container

As described in clause 4.5, the multiple networks for groups of one or more OS container are used for meeting customer's service requirements to solve the needs of differential network performance and security isolation.

For the case of container-based VNF instance, the configuration of multiple networks for OS containers can be executed as a first step. Then when a group of one or more OS container (e.g. Pod) is created, the secondary container cluster external network can be connected at the same time. After that, the secondary container cluster external network is used for communication with other groups of one or more OS container.

SNCP is introduced to provide information about required additional connectivity of the groups of one or more OS container supporting a VNF instance to secondary container cluster networks. SNCP can contain three parts:

- Information describing the secondary networks needed in a container-based VNF instance.
- Network attributes of secondary container cluster internal network or secondary container cluster external network are used to describe the attribute of secondary container cluster internal network or secondary container cluster external network.
- Network policy of secondary container cluster internal network and secondary container cluster external network is used to describe network rules for secondary container cluster internal network and secondary container cluster external network.

Before a container-based VNF is instantiated, the configuration of multiple networks is requested by the NFVO. Then in a Kubernetes® environment, custom resource definitions (i.e. CRDs) are required to complete the configuration of multiple networks by parsing the information of SNCP(s). The CNI^{TM} plugin referenced in the CRDs is triggered to configure secondary container cluster internal network(s) or secondary container cluster external network(s) based on the SNCP(s).

When a Pod is created, the CISM connects the Pod to the corresponding networks. Meanwhile, the secondary container cluster internal network or the secondary container cluster external network also could be assigned to network policy of secondary container cluster internal network or secondary container cluster external network.

Network policy can define the rule what traffic is allowed to and from the Pod(s). In case that the secondary container cluster internal network is overlay network, the network policy can be used to control the exposure of the secondary container cluster internal network only within the CIS cluster.

SNCP can include the following information:

- name of secondary network (e.g. management network, control network, data network, etc.);
- network attribute of secondary network (e.g. IPAM, DNS, Gateway, route, etc.);
- CNI^{TM} plugins of secondary network (e.g. CNI^{TM} plugin version, name, type, etc.).

The network attributes of secondary container cluster internal network or secondary container cluster external network can include the following parameters:

- Ports of secondary container cluster internal network or secondary container cluster external network (e.g. EHT0, NET1, NET2, etc.).
- Performance of secondary container cluster internal network or secondary container cluster external network (e.g. Qos, bandwidth, delay, jitter, etc.).

The network policy of secondary container cluster internal network or secondary container cluster external network can include the following parameters:

- The rules of secondary container cluster internal network or secondary container cluster external network (e.g. network policy).
- The relationship between the VNFD and the SNCP can be described as follows:
  - The internal VLD can contain information about additional secondary container cluster internal/external network. The VLD can also contain information about the relationship about with SNCP for internal network connectivity of a container-based VNF.
  - The internal CPD (e.g. VduCpd) can contain information about the association of VNFC internal CPs with secondary container cluster internal/external networks based on mapped properties (as declared in respective declarative descriptors) of MCIOs requesting compute/storage/network resources.
  - The external CPD (VnfExtCpd) can contain information about the mapping with VduCpd. In case for OS container, VduCpd is used to describe the network interface of the group of one or more OS container.
  - The external CPD (VnfExtCpd) can contain information about the association of VNF external CP with the secondary container cluster internal/external network based on mapped properties (as declared in respective declarative descriptors) of MCIO requesting compute/storage/network resources.

6.9 Solution for creating internal Network Connectivity for a VNF

6.9.1 General description

The use case in clause 5.3 describes the creation of network connectivity for the groups of one or more OS container during VNF LCM operations. The solution for creating internal/external VNF connectivity for container-based VNFs in this clause leverages the capabilities available in CISM solutions based on e.g. Kubernetes® environment for the group of one or more OS container (i.e. Pod) to connect to a secondary container cluster internal network. The procedure is similar to the creation of externally managed internal virtual link connectivity for VM based VNFs.

For a container-based VNF instance which is composed of VNFC instances that are all realized by groups of one or more OS container, the internal Virtual Link Descriptors in the VNFD, together with the connectivity requirements expressed in the VduCpds associated to the internal VLD, contain the internal network connectivity information that the NFVO will use to determine how to map virtual links to container cluster internal and external networks. In case the NFVO, based on the affinity rules defined in the VNFD, can determine that all containers of the container-based VNF will be in the same cluster, the NFVO will request to the CISM creating the secondary network definition resource for additional secondary container cluster internal network. The additional secondary container cluster internal network can be used for the internal network connection of container-based VNFs. Otherwise, if the NFVO determines that the groups of one or more OS container of the container-based VNF will be deployed in different CIS clusters, an additional secondary container cluster external networks is used.

NOTE: The secondary network definition resource is the managed object in the CIS cluster which has actual state and allocated CIS cluster infrastructure resources. The secondary network definition resource is created based on a declarative descriptor used in the request to the CISM, and parts of the declarative descriptor data can be derived from the SNCP.

EXAMPLE: An example of "secondary network definition resource" in Kubernetes® environment is a "NetworkAttachmentDefinition" resource.

The information of secondary container cluster internal network is described in clause 6.8.

6.9.2 Procedure

1) The sub-flow between the NFVO and the CISM/infrastructure manager (e.g. VIM).

   The NFVO can obtain the network capabilities of the CIS clusters from the CISM/CCM.

NOTE 1: The work split between CISM and CCM is out of scope for the present document.

   The NFVO can determine the need to create and configure cluster secondary networks based on the existence of internal VLs in the VNFD to which VduCpds are associated.

   If needed, the NFVO requests from the infrastructure manager (e.g. VIM) the creation of the network(s) associated to secondary container cluster internal network(s).

NOTE 2: The creation and configuration of network resources for the cluster secondary networks can be regarded as a cluster management activity, and it is assumed to be a pre-condition for the following steps.

   The NFVO requests from the CISM the creation of a secondary network definition resource.

   In the request the NFVO can inject specific attributes of container cluster internal/external network to CISM and indicate the respective SNCP(s).

2) The sub-flow between the NFVO and the VNFM.

   The NFVO includes references to the previously created secondary network definition resource in the instantiation request. The networks are used as externally managed VLs by the VNF.

3) The sub-flow between the VNFM and the CISM.

   The VNFM injects references of secondary container cluster internal/external network definition resources, received from NFVO, to CISM and indicate the respective SNCP(s).

   The CISM populates annotation's placeholders of the declarative descriptors of the MCIO with information of secondary network definition resources used to connect to container cluster internal/external networks.

   During the process of the VNF instantiation operation, according to the MCIO declarative descriptor, when the MCIO is created, the internal/external network interface of the group of one or more OS container can be connected to the secondary container cluster internal/external networks through CNI^{TM} Plugins.

6.10 Solution for creating external Network Connectivity for an NS

6.10.1 General description

The use case in clause 5.1 describes the creation of network connectivity for VNFs during NS LCM and VNF LCM operations. The solution for creating external VNF connectivity for container-based VNFs in this clause leverages the capabilities available in CISM solutions based on e.g. Kubernetes®.

6.10.2 NSD

For the NS instance which is composed of container-based VNF instances, the VLDs in the NSD have the external network connectivity information that the NFVO will use to determine how to map VLs to container cluster external networks in case the container-based VNF instances are in multiple CIS clusters or to container cluster internal networks in case they are in the same CIS cluster. The NFVO can use the VLDs to determine the resource requirements and request to one or multiple CISMs allocating the secondary network definition resources that are used for additional secondary container cluster internal/external networks. The additional secondary container cluster external network can be used for the external network connection of container-based VNFs. In this context, and depending on the final placement of the VNF which are part of the NS, the secondary container cluster external networks realize part or all of the NS VL connectivity, and in particular, these secondary container cluster external networks are used for the external connectivity of those container-based VNFs.

In addition, the VLD can contain information about external network connectivity of container-based VNFs that is used for additional secondary container cluster internal/external networks.

6.10.3 Procedure

1) The sub-flow between the NFVO and the CISM//infrastructure manager (e.g. VIM).

   In order to proceed with the NS instantiation with container-based VNFs, the OSS/BSS sends to the NFVO an "InstantiateNsRequest" to trigger NS instantiation operation.

   Based on the information contained in the NSD with container-based VNFs, the NS to be instantiated can contain NS VL and associated resources.

   As part of the NS instantiation, resource orchestration is performed by the NFVO for the fulfilment of network resources.

   Based on the NS VL connectivity requirements expressed in the NSD (e.g. with additional secondary container cluster external network associated properties), as well as the connectivity requirements expressed in the VnfExtCpds, the NFVO determines the connection information of the constituent container-based VNFs.

   If needed, the NFVO requests from the infrastructure manager (e.g. VIM) the creation of the network(s) associated to secondary container cluster internal/external network(s).

NOTE: The creation and configuration of network resources for the cluster secondary networks can be regarded as a cluster management activity, and it is assumed to be a pre-condition for the following steps.

   The NFVO requests from the CISM the creation of a secondary network definition resource.

   In the request the NFVO can inject specific attributes of container cluster internal/external network to the CISM and indicate the respective SNCP(s).

2) The sub-flow between the NFVO and the VNFM.

   The NFVO requests the instantiation of the VNF and includes references to the previously created secondary network definition resource(s).

3) The sub-flow between the VNFM and the CISM.

   The VNFM injects specific references attributes of secondary network definition resources, received from NFVO, to the CISM and indicate the respective SNCP(s).

   The CISM populates annotations' placeholder of the declarative descriptor of MCIO with information about secondary network definition resources used to connect to container cluster external networks.

   During the process of the VNF instantiation operation for container-based VNFs, according to the MCIO declarative descriptor, when the MCIO is created, the CNI^{TM} Plugin connects the network interface of the group of one or more OS container to the secondary container cluster external networks.

6.11 Solution for Modifying Network Connectivity

6.11.1 General description

The solution for changing external VNF connectivity for container-based VNFs described in the present clause leverages the capabilities available in CISM industry solutions, e.g. Kubernetes®. The procedure is similar to the change of external virtual link connectivity for VM-based VNFs and can be triggered from the OSS/BSS via the NFVO to the VNFM and the VNF. Creation of additional networks when reconnecting a container-based VNF (initially connected to another container-based VNF) to a VM-based VNF is outside the scope of this solution.

A description of this solution in a Kubernetes® environment is provided in clause 6.11.2.

6.11.2 Pre-requisite

Following are the pre-requisites to trigger the change of external network connectivity for container-based VNFs from NFV-MANO:

- The NSD has the model of target external virtual links for container-based VNFs.
- The VNF package for a container-based VNF has MCIOPs that specify the association of the MCIOs with secondary container cluster internal/external networks (see ETSI GS NFV-IFA 036 [i.3] for creation of secondary container cluster networks).

NOTE: In a Kubernetes® environment, this information is contained in annotation placeholders of the pod specifications.

- When the VNF package is delivered by a VNF provider, the annotations field in MCIOPs can have entries for secondary container cluster internal/external networks associating MCIOs with the internal virtual links of the VNF.
- The NSD is on boarded in the NFVO; The NS and the VNF for which change in external connectivity is requested is successfully instantiated.
- Plugin(s) (a part of the CIS cluster characteristics) supporting annotations mechanism for secondary container cluster internal/external networks are installed on the CIS cluster nodes hosting the VNF.

6.11.3 Procedure

The procedure for change of external VNF connectivity for a container-based VNFs in NFV-MANO is described below:

- When the operator invokes the Update NS operation with the updateType parameter set to "ChangeExtVnfConnectivity" from the Os-Ma-nfvo reference point, the ExtVirtualLinkData in ChangeExtVnfConnectivityData can hold information about the target external virtual links to be connected to the VNF. Information about the relevant target secondary container cluster internal/external networks to be associated with VnfExtCps can be passed with the ChangeExtVnfConnectivityData.

The NFVO can inject specific attributes of the target network described in the NSD into the CISM to update the respective multiple networks configuration profiles of the secondary container cluster external or internal network. The NFVO will pass the ChangeExtVnfConnectivityData to the VNFM. The VNFM will perform modification of the association of CP to external virtual links as per the current mechanism for VM based VNFs.

NOTE: The multiple networks configuration profiles of the secondary container cluster internal networks can only be updated in case the secondary container cluster internal networks are used as VNF external link. This originates from the fact that a change of connectivity only applies to the external VNF connectivity as capabilities supported by NFV-MANO.

The VNFM can inject information regarding the target secondary container cluster external or internal networks associated with the VNF external virtual links to the CISM.

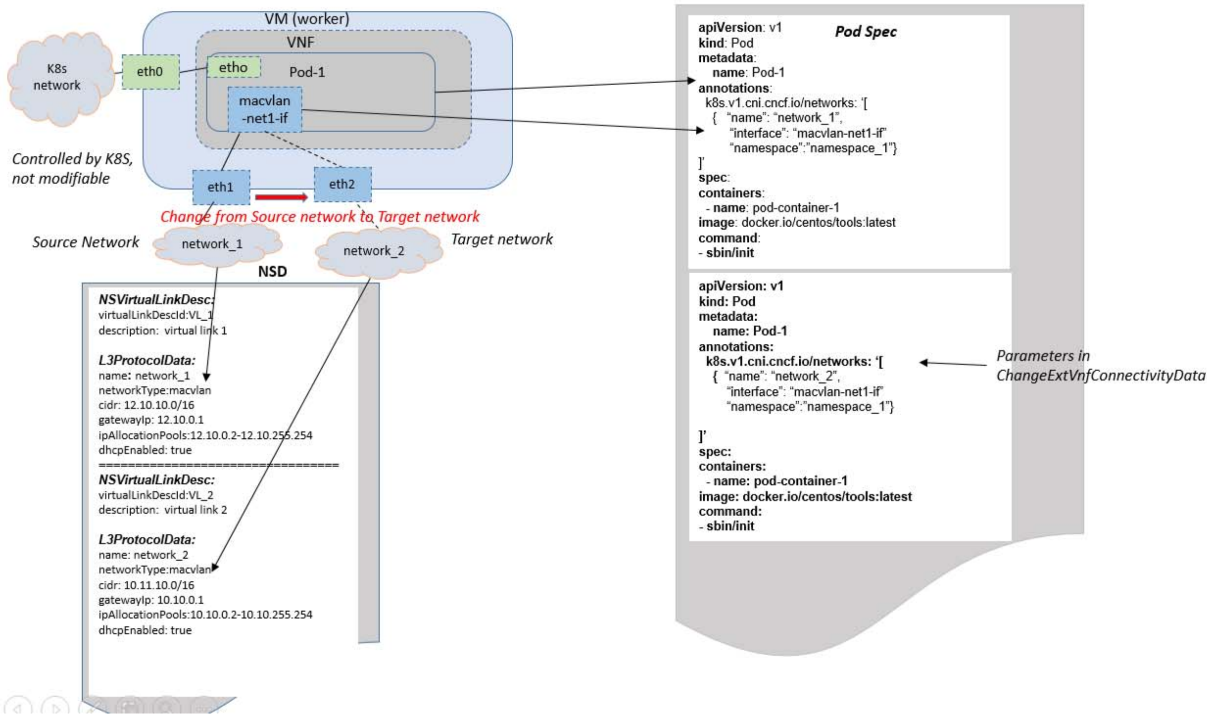
The change in network connectivity will take effect after the group of one or more OS container is restarted.

6.11.4 CISM Implementation based on Kubernetes®

6.11.4.1 Pre-requisite

Figure 6.11.4.1-1 illustrates an example for changing secondary container cluster external network for a group of one or more OS container from network_1 to network_2 in Kubernetes®. Following are the pre-requisites for implementing the solution described in clause 6.11.1 in a Kubernetes® environment for a group of one or more OS container (i.e. Pod) to connect to a secondary container cluster external network at run time:

- The Pod specification has annotations placeholder which can be populated with the network details at run time through configuration files of the MCIOP (i.e. values.yaml of Helm™ chart).
- Attributes of secondary container cluster internal/external network are described in the manifest files of secondary container cluster external network (i.e. Network Attachment Definition (NAD)).


VM (worker)
VNF
K8s network
eth0
etho
Pod-1
macvlan -net1-if
Controlled by K8S, not modifiable
eth1
eth2
Change from Source network to Target network
Source Network
network_1
network_2
Target network
NSD
Pod Spec
Parameters in ChangeExtVnfConnectivityData

```
NSVirtualLinkDesc:
virtualLinkDescId:VL_1
description:  virtual link 1

L3ProtocolData:
name: network_1
networkType:macvlan
cidr: 12.10.10.0/16
gatewayIp: 12.10.0.1
ipAllocationPools:12.10.0.2-12.10.255.254
dhcpEnabled: true
=================================
NSVirtualLinkDesc:
virtualLinkDescId:VL_2
description:  virtual link 2

L3ProtocolData:
name: network_2
networkType:macvlan
cidr: 10.11.10.0/16
gatewayIp: 10.10.0.1
ipAllocationPools:10.10.0.2-10.10.255.254
dhcpEnabled: true
```

```
apiVersion: v1
kind: Pod
metadata:
   name: Pod-1
annotations:
  k8s.v1.cni.cncf.io/networks: '[
   {  "name": "network_1",
      "interface": "macvlan-net1-if"
      "namespace":"namespace_1"}
]'
spec:
containers:
  - name: pod-container-1
image: docker.io/centos/tools:latest
command:
- sbin/init
```

```
apiVersion: v1
kind: Pod
metadata:
   name: Pod-1
annotations:
  k8s.v1.cni.cncf.io/networks: '[
   { "name": "network_2",
     "interface": "macvlan-net1-if"
     "namespace":"namespace_1"}

]'
spec:
containers:
  - name: pod-container-1
image: docker.io/centos/tools:latest
command:
- sbin/init
```

Figure 6.11.4.1-1: Illustration of change of secondary container cluster external network for groups of one or more OS container

6.11.4.2 Procedure

Apart from connectivity of groups of one or more OS container to the default primary container cluster internal/external network (which is not modifiable and controlled by CISM), the group of one or more OS container can be connected to additional secondary container cluster internal/external networks through additional CNI™ plugins. Change of network connectivity for a group of one or more OS container can be achieved at run time by changing the annotations place holder value in the declarative descriptor of the corresponding MCIO for the group of one or more OS container from the source network to the target network. Overriding the source network with the target network is realized by injecting the target network through configuration files (i.e. values.yaml) of the MCIOP.

The target network passed in values.yaml files in the Helm™ install command to the CISM will override the source network in the annotations field of the MCIO declarative descriptor. The change in network will take effect after the group of one or more OS container is restarted.

7 Potential architectural enhancements

7.1 Enhancements related to NFV descriptors and other artefacts

7.1.1 Overview

In the context of the use cases described in clauses 5.1 and 5.3, the NSD and VNFD contain networking related information to describe the primary and secondary container cluster external and internal networks and the relationship between the networks and container-based VNFs or VNFCs. The following clauses identify the proposed enhancements.

NOTE: Clause 7.1 of the present document does only cover enhancements applicable to fully containerized VNFs (i.e. all VNFCs are containerized) and NSs referencing containerized VNFs only.

7.1.2 VNFD

As described in clauses 6.8 and 6.9, for the management of network connectivity of containerized VNFs during VNF LCM operations, the information about secondary container cluster internal/external networks is expected to be contained in internal Virtual Link descriptors in the VNFD, the information about the association of internal CPs with the secondary container cluster internal/external networks is expected to be contained in the CPDs.

As described in clause 6.6, cloud provider controller launches load balancer as a result of deployment of network MCIO (for example, Kubernetes® Service or ingress resources). ETSI GS NFV-IFA 011 [i.7] is proposed to be enhanced to support the definition of resource requirements for load balancer in case of cloud provider controller environment.

NOTE: In case the VNFCs of a VNF are deployed in different CIS clusters, the connectivity between these VNFCs is provided by a primary or secondary container cluster external network.

7.1.3 NSD

As described in clause 6.10, for the management of network connectivity of containerized VNFs during the NS LCM operations, the information about networks used by an NS is expected to be contained in the NS Virtual Link descriptors in the NSD. The information about the association of VNF external CPs with the secondary container cluster internal/external networks is expected to be contained in the NSD.

The NSD may be enhanced with auxiliary network information, for example, to indicate the association of NS Virtual Link descriptors in the NSD with the SNCP(s).

NOTE: In case the VNFs of an NS are in the same CIS cluster, the connectivity between these VNFs is provided by a primary or secondary container cluster internal network.

7.1.4 NSD file structure

Meanwhile, as described in clause 6.10, SNCP(s) for external network connectivity of containerized VNFs are suggested to be included as artefacts in the NSD file structure.

Therefore, the NSD file structure is proposed to be enhanced with the ability to include new artefacts which represent the declarative descriptors of network resources for the secondary container cluster internal and external network(s), i.e. SNCP(s).

7.2 Enhancement related to NFV-MANO functional aspects

7.2.1 Overview

In the context of use cases in clause 5 of the present document, NFV-MANO is proposed to be enhanced with the capabilities to support the management of network connectivity for containerized VNFs, and further according to the solutions in clause 6, especially in clauses 6.8, 6.9, 6.10, 6.11, the NFV-MANO, e.g. NFVO, VNFM and CISM, is proposed to interact with each other to realize the management of network connectivity which is mainly related to the primary and secondary container cluster internal and external networks.

7.2.2 NFV-MANO

- VIM: The described use cases and solutions do not propose enhancements of the VIM. In the NFV solutions on network connectivity of containerized VNFs, the VIM is proposed to provide to the CISM the NFVI virtualised network resources needed by CIS clusters. In addition, during the process of NS and VNF lifecycle management operations, the secondary container cluster internal and external networks are proposed potentially to be created with the virtualised network resources managed by the VIM as needed.

- VNFM: The VNFM is proposed to be enhanced with the capability to request to the CISM the establishment of connectivity for groups of one or more OS container. The VNFM is also proposed to be enhanced with the capability to provide the network information related to the secondary container cluster internal/external networks to the CISM.

  As mentioned in clause 6.6, cloud provider controller launches load balancer as a result of deployment of network MCIO (for example, Kubernetes® Service or ingress resources). Resource definitions/requirements utilized for load balancer in underlying cloud provider environment are expected to be part of grant request from VNFM to NFVO.

- NFVO: The NFVO is proposed to be enhanced with the capability to initiate the management operations on multiple container cluster networks (e.g. configure networks related to the secondary container cluster internal/external networks) towards the CISM/CCM during the NS or VNF lifecycle management through the OS container multiple networks management service interface. The NFVO is also proposed to be enhanced with the capability to provide the network information related to the secondary container cluster internal/external network to the CISM. The NFVO is proposed to be enhanced with the capability to provide references to VNF external and VNF internal networks (the latter as externally managed VLs) to the VNFM.

  As mentioned in clause 6.6, cloud provider launches load balancers as a result of deployment of network MCIO (for example, Kubernetes® Service or ingress resources). Resources utilized for load balancer in underlying cloud provider environment is expected to be part of grant request from VNFM to NFVO.

- CISM: The CISM is proposed to be enhanced with the capability to perform the multiple networks management operations exposed by a CISM service interface, which are initiated by the NFVO to configure the secondary container cluster internal and external networks. The CISM is also proposed to be enhanced to expose the OS container multiple networks management service interface to its consumers, e.g. NFVO, VNFM, or other third parties.

NOTE: The work split between CISM and CCM is out of scope for the present document.

7.3 Enhancement related to NFV-MANO interfaces

7.3.1 Overview

In the context of the solutions described in clause 6 of the present document, especially in clauses 6.9, 6.10 and 6.11, the CISM is proposed to support the capability to configure secondary container cluster internal networks or external networks upon the reception of a request from the NFVO. The NFVO is proposed to invoke an interface produced by the CISM to implement the creation, modification, deletion of OS container multiple networks, including the secondary container cluster external networks and the secondary container cluster internal networks.

7.3.2 Management of OS container multiple networks

The requirements on the OS container network management service interface exposed by the CISM specified in ETSI GS NFV-IFA 040 [i.4], do not contain requirements on supporting the configuration of OS container multiple networks. Therefore, it is recommended to specify a requirement on the CISM OS container network management service to support the configuration of OS container multiple networks.

The following interactions are analysed to identify potential architectural enhancements in NFV-MANO:

- NFVO-CISM interaction:

  The NFVO is proposed to be enhanced to consume the OS container network management service interface exposed by the CISM to manage OS container multiple networks for containerized VNFs. In case of the creation, modification and deletion of OS container cluster external or internal network connectivity (e.g. the secondary container cluster external or internal network) between containerized VNFs, the NFVO is proposed to invoke the OS container network management service interface to request executing the corresponding operations. The CISM is proposed to support the capability of providing information about the current status of container cluster external or internal networks to the NFVO via this interface.

  As described in the solution of clauses 6.9 and 6.10, the NFVO is proposed to support requesting from the CISM to configure secondary container cluster external or internal networks via the OS container network management service interface.

  As described in the solution of clause 6.11, the NFVO is proposed to support requesting from the CISM to modify secondary container cluster external or internal networks for containerized VNF(s) via the OS container network management service interface.

- VNFM-CISM interaction:

  The VNFM is proposed to be enhanced to consume the OS container network management service interface exposed by the CISM to obtain information about OS container multiple networks for containerized VNFs. The CISM is proposed to support the capability of providing information about the current status of container cluster internal or external networks to the VNFM via this interface.

  As described in the solution of clause 6.11, the VNFM is proposed to support requesting to the CISM to modify the association of the MCIO corresponding to an external CP from the source secondary container cluster internal or external network to the target secondary container cluster internal or external network via the OS container network management service interface.

8 Recommendations for future work

8.1 VNF Descriptor and Packaging

The present clause provides recommendations related to VNF descriptor and packaging, on the primary and secondary container cluster internal/external network properties and associated network resource requirements aspects for container-based VNF.

Table 8.1-1 provides the recommendations related to VNF descriptor and packaging.

Table 8.1-1: Recommendations related to VNF descriptor and packaging

Identifier	Recommendation description	Comment/Traceability
It is recommended that a requirement set is specified for the VNF descriptor and packaging to support:		
VNF_PACK.META.001	A description of internal CP including connectivity with one VL realized as one secondary container cluster internal/external network.	Refer to clauses 6.8 and 7.1.
VNF_PACK.META.002	A description of internal VL including additional properties for one or more secondary container cluster internal/external networks (see note 1).	Refer to clauses 6.8 and 7.1.
VNF_PACK.META.003	A description of internal VL including information for associating the VLD with the declarative descriptors of secondary container cluster internal/external networks.	Refer to clauses 6.8 and 7.1.
VNF_PACK.META.004	A description of resource requirements for load balancers associated to network MCIOs in case of cloud provider controller environment.	Refer to clauses 6.6 and 7.1.
NOTE 1: Declarative descriptors of resources related to the secondary container cluster internal/external networks (e.g. network attachment definitions) are not part of the VNF package as they are NFVI/cluster related. NOTE 2: Declarative descriptors of infrastructure resources related to the secondary container cluster internal/external networks (e.g. Network Attachment Definitions) are not part of the VNF package as they are NFVI/CIS cluster related.		

8.2 NS templates

The present clause provides recommendations related to NS templates, on the secondary container cluster internal/external network properties and associated network resource requirements aspects for containerized VNF.

Table 8.2-1 provides the recommendations on network properties related to NS templates.

Table 8.2-1: Recommendations related to NS templates

Identifier	Recommendation description	Comment/Traceability
It is recommended that a requirement set is specified for the NS templates to support:		
NST.VLD.001	A description of VL that enables specifying additional properties for one or more secondary container cluster internal/external networks.	Refer to clauses 6.10 and 7.1.
NST.VLD.002	The association of a VL descriptor with the declarative descriptors of secondary container cluster internal/external networks.	Refer to clauses 6.10 and 7.1.

8.3 NFV-MANO functional aspects

The present clause provides recommendations related to NFV-MANO functional aspects, including the NFVO, VNFM and CISM.

Table 8.3-1 provides the recommendations related to NFV-MANO functional aspects.

Table 8.3-1: Recommendations related to NFV-MANO functional aspects

Identifier	Recommendation description	Comment/Traceability
It is recommended that a requirement set is specified for the NFV-MANO entities to support:		
Nfvo.Oscmnm.001	The ability of the NFVO to initiate the multiple networks management operations during the NS LCM for the fulfillment of NS and VNF connectivity.	The NFVO is enhanced to initiate the multiple networks management operations (e.g. create, modify or delete external networks related to the secondary container cluster internal/external networks). Refer to clauses 6.9 and 7.2.
Nfvo.Oscmnm.002	The ability of the NFVO to handle resource requirement for load balancer as part of granting exchange.	The NFVO is enhanced to incorporate load balancer related resource properties in grant response. Refer to clauses 6.6 and 7.2.
Vnfm.Oscmnm.001	The ability of the VNFM to initiate management operations during the life-cycle management of containerized VNFs that establish the connectivity of the VNFs to multiple networks.	The VNFM is enhanced to provide references to the multiple networks in the OS container network management operations towards the CISM. Refer to clauses 6.9 and 7.2
Vnfm.Oscmnm.002	The ability of the VNFM to handle resource requirement for load balancer as part of grant request.	The VNFM is enhanced to incorporate load balancer related resource requirements in grant request. Refer to clauses 6.6 and 7.2.
Cism.Oscmnm.001	The ability of the CISM to perform the multiple networks management operations.	The CISM is enhanced to perform the multiple networks management operations exposed by a CISM service interface. Refer to clauses 6.9, 6.10 and 7.2.

8.4 NFV-MANO interfaces

Table 8.4-1 provides the recommendations related to NFV-MANO interfaces, i.e. on a service interface related to OS container multiple networks management.

Table 8.4-1: Recommendations related to NFV-MANO interfaces

Identifier	Recommendation description	Comment/Traceability
It is recommended that a requirement set is specified for the OS container network management service interface produced by the CISM to support:		
CismMNetMgt.001	Creating a secondary container cluster internal/external network.	Refer to clauses 6.9, 6.10 and 7.3.
CismMNetMgt.002	Modifying a secondary container cluster internal/external network.	Refer to clauses 6.11 and 7.3.
CismMNetMgt.003	Deleting a secondary container cluster internal/external network.	Refer to clause 7.3.
CismMNetMgt.004	Querying information about the current status of a secondary container cluster internal/external network.	Refer to clauses 6.9, 6.10 and 7.3.
CismMNetMgt.005	Sending notifications in the event of changes for a secondary container cluster internal/external network.	Refer to clauses 6.9, 6.10 and 7.3.

Annex A: Change History

Date	Version	Information about changes
October 2019	0.0.1	First draft, implementing contributions: • NFVIFA(19)000852r1_IFA038_network_connectivity_for_container_based_VNF_Skeleton • NFVIFA(19)000854r2_IFA038-Scope • NFVIFA(19)000896_IFA038_Remove_Annex_on_Authors_and_contributors
December 2019	0.0.2	Implementation of approved contributions from IFA#177 (see NFVIFA(19)0001005r1_Report_IFA_177_F2F_Fukuoka_December_2019): • NFVIFA(19)000975r3_IFA038_Clause_4_1_Background • NFVIFA(19)000954r2_IFA038_Clause_4_2_Introduction • NFVIFA(19)000986r2_IFA038_Clause_4_3_Network_connectivity_for_OS_Containers_among_VNFs
February 2020	0.0.3	Implementation of approved contributions from IFA#184: • NFVIFA(19)000985r8_IFA038_Clause_5_1_network_connectivity_for_OS_Containers_among_VNFs • NFVIFA(19)000987r9_IFA038_Clause_4_4_network_connectivity_for_OS_Containers_among_Container_Cluster_Nodes
April 2020	0.0.4	Implementation of approved contributions from IFA#190: • NFVIFA(19)000988r5_IFA038_Clause_4_5_Network_connectivity_for_OS_containers_attaching_multiple_network_planes • NFVIFA(20)000198r1_IFA038_Clause_5_2_Usecase_for_Change_External_Vnf_Connectivity
June 2020	0.0.5	Implementation of approved contributions from IFA#198 and IFA#200: • NFVIFA(20)000358r2_IFA038_Clause_5_2_network_connectivity_of_OS_Container_for_M • NFVIFA(20)000359r4_IFA038_Clause_5_3_Network_connectivity_between_MCIOs_and_mul
September 2020	0.0.6	Implementation of approved contributions from IFA#203, IFA#206 and IFA#207: • NFVIFA(20)000551r1_IFA038_Route_sharing_with_BGP_ • NFVIFA(20)000550r1_IFA038_Inter-node_network_with_overlay_technologies • NFVIFA(20)000552r1_IFA038_Inter-node_network_with_NFVI_technologies • NFVIFA(20)000436r8_IFA038_Definitions
November 2020	0.0.7	Implementation of approved contributions from IFA#215 and IFA#216: • NFVIFA(20)000601r3_IFA038_6_x_Inter_node_network_with_direct_mode • NFVIFA(20)000602r4_IFA038_6_x_Inter_node_network_with_L2_underlay_ • NFVIFA(20)000631r4_IFA038_Clause_6_x_Pod_networks_creation_for_multiple_network • NFVIFA(20)000632r2_IFA038_Clause_6_x_multiple_networks_for_Pods • NFVIFA(20)000650r2_IFA038_Synchronize_terms_inline_with_IFA036_and_IFA040 • NFVIFA(20)000604r6_IFA038_Clause_6_Solution_ModifyNetworkConnectivity
February 2021	0.0.8	Implementation of approved contributions from IFA#223 and IFA#226: • NFVIFA(21)000018r2_IFA038_Clause_3_1_Terms_Improvement • NFVIFA(21)000019r1_IFA038_Clause_6_7_and_6_8_Synchronize_terms • NFVIFA(21)000066r2_IFA038_Clause_6_x_Creation_of_Internal_Network_Connectivity
March 2021	0.0.9	Implementation of approved contributions from IFA#229: • NFVIFA(21)000064r5_IFA038_Clause_6_x_Creation_of_External_Network_Connectivity • NFVIFA(21)000097r4_IFA038_Clause_6_8_remove_an_Editor_s_Note
April 2021	0.1.0	Implementation of approved contributions from IFA#235: • NFVIFA(21)000265r1_IFA038_Clause_5_clarify_UC_for_Network_Connection_creation • NFVIFA(21)000269r5_IFA038_Clause_7_x_Enhancement_on_NFV_information_models • NFVIFA(21)000271r4_IFA038_Clause_7_x_Enhancement_on_MANO

Date	Version	Information about changes
May 2021	0.2.0	Implementation of approved contributions from IFA#237 and IFA#238: • NFVIFA(21)000270r5_IFA038_Clause_7_x_Enhancement_of_CISM_for_multiple_networks • NFVIFA(21)000363r2_IFA038_Clause_8_x_Recommendation_on_VNF_Descriptor_and_Packa • NFVIFA(21)000364r2_IFA038_Clause_8_x_Recommendation_on_NS_Templates • NFVIFA(21)000365r2_IFA038_Clause_8_x_Recommendation_on_OS_container_multiple_ne • NFVIFA(21)000366r3_IFA038_Clause_8_x_Recommendation_on_MANO
June 2021	0.3.0	Implementation of approved contributions from IFA#240: • NFVIFA(21)000422_IFA038_Remove_hanging_clauses • NFVIFA(21)000423r1_IFA038_Remove_multiple_remaining_editor_s_notes • NFVIFA(21)000425r1_IFA038_Clause_3_1_Editor_s_note_resolution • NFVIFA(21)000426r1_IFA038_Clause_8_1_Editor_s_note_resolution • NFVIFA(21)000427r1_IFA038_Clause_4_3_Editor_s_note_resolution • NFVIFA(21)000428r1_IFA038_Remove_an_Editor_s_Note_in_multiple_clauses
June 2021	0.4.0	Implementation of approved contributions from IFA#242: • NFVIFA(21)000462r1_IFA038-Refine_use_cases_concerning_VNFM_role • NFVIFA(21)000463r1_IFA038-Refine_solutions_concerning_VNFM_role • NFVIFA(21)000464r1_IFA038-Refine_architecture_enhancements_concerning_VNFM_role • NFVIFA(21)000490_IFA038_Clause_3_1_Updating_definitions • NFVIFA(21)000491_IFA038_Clause_4_Updating_concepts • NFVIFA(21)000492_IFA038_Clause_5_Editorials_in_use_case_descriptions • NFVIFA(21)000493r1_IFA038_Clause_4_Network_Connectivity_For_VNFs_Using_load_bal • NFVIFA(21)000498_IFA038_Clause_6_1_Updates_to_solution_modifying_network_conn • NFVIFA(21)000499_IFA038_Clause_6_3_and_6_4_Refactoring_routing_solutions • NFVIFA(21)000500_IFA038_Clause_6_7_Corrections_to_solution_of_pod_networks_fo • NFVIFA(21)000501_IFA038_Clause_6_10_Small_corrections_to_solution_external_ne • NFVIFA(21)000512r1_IFA038_Clause_6_5_Editor_s_note_resolution
June 2021	0.5.0	Implementation of approved contributions from IFA#243: • NFVIFA(21)000494r5_IFA038_Clause_6_Network_connectivity_for_VNFs_using_Load_Bal • NFVIFA(21)000513_IFA038_Clause_5_1_4_4_Editor_s_note_resolution • NFVIFA(21)000514r1_IFA038_Clause_6_10_3_Editor_s_note_resolution • NFVIFA(21)000515_IFA038_Clause_7_1_1_Editor_s_note_resolution
July 2021	0.6.0	Implementation of approved contributions from IFA#245 and IFA246: • NFVIFA(21)000557r1_IFA038_Remove_an_Editor_s_Note_in_multiple_clauses • NFVIFA(21)000550r1_IFA038_Clause_4_5_Editor_s_note_resolution • NFVIFA(21)000581_IFA038_Review_part_1_change_proposals_clause_3_4 • NFVIFA(21)000602_IFA038_Review_part_2_change_proposals_clause_5 • NFVIFA(21)000603r1_IFA038_Review_part_3_change_proposals_clause_6 • NFVIFA(21)000604r1_IFA038_Review_part_4_change_proposals_clause_7_8 • NFVIFA(21)000635r1_IFA038_External_VNF_connectivity
August 2021	0.7.0	Implementation of approved contributions from IFA#248 and IFA249: • NFVIFA(21)000622r3_IFA038_Review_part_6_open_comments_clause_6_1-6_6 • NFVIFA(21)000682r1_IFA038_Clause_6_Resolving_Comments • NFVIFA(21)000605r2_IFA038_Review_part_5_open_comments_clause_4_5 • NFVIFA(21)000640r2_IFA038_Internal_VNF_connectivity • NFVIFA(21)000694r1_Pre-review_of_IFA038_-_Editorial_fixes_and_minor_wording_imp • NFVIFA(21)000705r1_IFA038_Clause_6_1_New_overview_clause
August 2021	0.8.0	Implementation of approved contributions from IFA#250: • NFVIFA(21)000568r4_IFA038_Clause_6_8_Editor_s_note_resolution • NFVIFA(21)000641r3_IFA038_VNFM_role_in_the_management_of_secondary_networks • NFVIFA(21)000685r1_IFA038_Comments_on_7_1 • NFVIFA(21)000693r2_IFA038_External_VNF_connectivity_text_alignment • NFVIFA(21)000721_IFA038_Clause_6_9_2_Editor_s_note_resolution • NFVIFA(21)000725r3_IFA038_Clause_6_load_balancer_resource_management

Date	Version	Information about changes
September 2021	0.9.0	Implementation of approved contributions from IFA#253: • NFVIFA(21)000623r4_IFA038_Review_part_7_open_comments_clause_6_8-6_10 • NFVIFA(21)000624r2_IFA038_Review_part_8_open_comments_clause_7_8 • NFVIFA(21)000707r4 IFA038 pre-review tech comments
October 2021	0.10.0	Implementation of approved contributions from IFA#255: • NFVIFA(21)000755r2_IFA038_MNCP_to_Secondary_network_configuration_profile • NFVIFA(21)000802r1_IFA038_Review_clause_1_2_3_editorial_clean-up • NFVIFA(21)000803r1_IFA038_Review_clause_4_editorial_clean-up • NFVIFA(21)000804_IFA038_Review_consistency_check_on_clause_7 • NFVIFA(21)000813r1_IFA038_Review_clause_5_1_clean-up • NFVIFA(21)000814r1_IFA038_Review_clause_6_clean-up • NFVIFA(21)000815r1_IFA038_Review_clause_8_clean-up • NFVIFA(21)000818r1_IFA038_Review_clause_5_2_and_5_3_clean-up • NFVIFA(21)000819r1_IFA038_Review_clean-up_of_note_proposals • NFVIFA(21)000832r1_IFA038_Multiple_clauses_Terminology_alignment_of_MCIO_vs_OS • NFVIFA(21)000839_IFA038_Clause_6_10_2_Addressing_EN_virtualised_resources

History

Document history		
V4.1.1	November 2021	Publication