CLOUDIFY

Key Challenges in Choosing the Right VNF Manager

hello@cloudify.co | cloudify.co

Table of Contents

Introduction

Cloud technology is seen as the technology of the future and is rapidly becoming mainstream for all service providers and telco operators trying to operate more efficiently as well as offer premium service. Software-based technologies such as Network Functions Virtualization (NFV) and Software Define Network (SDN) are seen as a must in order to transform from legacy networks into software-based and service-oriented networks. Networks based on cloud infrastructure can operate and deploy more efficiently and agilely than any Virtual Network Function (VNF) on its own private cloud or one of the public clouds (e.g. Amazon, Azure...). This gives the possibility for service providers to operate a flexible and dynamic network with the added opportunity for faster, cheaper, and automated deployment of the service.

In order to fully utilize this potential, operators need to introduce new ways of managing their networks through an NFV Management and Orchestration (MANO) function. MANO is seen as one of the most crucial parts of cloud infrastructure in order to implement all of the advantages that come with it. Some of these advantages include increased automation, simplification of infrastructure operation, scalability, and faster time to market when speaking about service deployment.

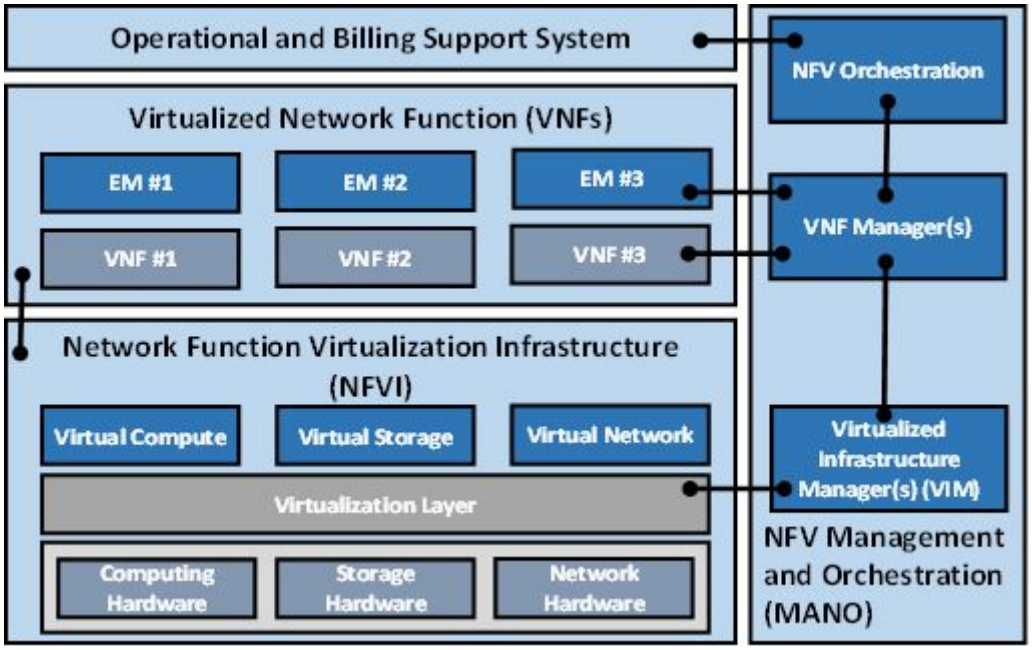
NFV Architecture in Brief

Basically, NFV implies a separation between the software and hardware on which different network functions are running. Instead of running network functions on dedicated hardware, you can deploy them as VMs on a virtualized hypervisor—vendor specific such as VMware or vendor agnostic like OpenStack. In this way, service providers are getting definite flexibility and agility in providing different services to end users, making it easier for them to answer market needs faster.

The basis for the NFV network architecture and design is defined by ETSI NFV Industry Specification Group (a consortium of service providers and vendors) and documented in a set of ETSI publications. The basic reference architecture framework is shown in the figure below and displays some important aspects:

- NFVI – Network Functions Virtualization Infrastructure includes hardware resources and a virtualization layer. Together these two allow for the use of virtual resources in order to deploy different functions over them.
- VNFs – Virtual Network Functions include different network functions that are implemented independently as software components running on a common virtual infrastructure.
- MANO – Management and Orchestration represents a group of functions responsible for managing all resources, functions, and services. It ensures appropriate resource

distribution, configuration, and automation by applying different policies and enabling high flexibility.



NFV network architecture

Such architecture allows service providers to operate a private cloud and utilize one pool of hardware resources for all network functions. All of this is enabled by the use of a virtualization layer and controlled by NFV Management and Orchestration, which enables providers to automate provisioning, deployment, and operation of a network service on top of a virtual infrastructure. Deploying end-to-end network services in an ecosystem like this hides the complexity of the underlying infrastructure, simplifies deployment of VNFs, and simplifies expansion of all resources.

MANO – NFV Management & Orchestration

As already mentioned, management and orchestration is of critical importance to cloud infrastructure. It is responsible for managing both the cloud infrastructure and network functions at the same time, enabling service providers to use their resources optimally while offering the best quality service possible. Here below, we will discuss the three MANO elements: VIM, VNF Manager, and NFV Orchestrator.

VIM (Virtual Infrastructure Manager)

The VIM is responsible for controlling the physical and virtual resources of a cloud infrastructure, orchestrating and optimizing the allocation, expansion, and release of NFVI resources using a precise inventory system. The VIM manager also serves as a repository of NFVI hardware resources (such as compute, storage, and networking) and software resources (like hypervisor), coordinating all necessary physical resources to deliver proper

network function. Providing performance and error notifications is yet another role of the VIM.

Hypervisor can be seen as the glue between the hardware and the software of any cloud infrastructure. The most commonly used hypervisor in the telco world is OpenStack, seen as the solution for multi-vendor environments to make it easier to accomplish multi-vendor provisioning of an underlying infrastructure through APIs. That's why vendors created their own OpenStack implementations, including Red Hat, Mirantis, Huawei, etc.

VNF Manager

The VNF manager is one of the most important parts of virtual infrastructure management since it should standardize virtual network functions and help in the interoperability between software-defined elements. The VNF Manager is responsible for lifecycle management of VNFs that include instantiation and termination of network functions, scaling of network functions, software upgrades of VNFs, configuration of various complexities, and reception of performance measurements and alarms on the VNF level.

Scaling of network functions depends on the given need and can include scaling in/out (meaning increasing and decreasing the number of VNFs) and scaling up/down (meaning increasing or decreasing the number of resources, e.g., memory or vCPU). The VNF Manager can be implemented for the same VNFs, different VNFs, single VNF instances, and multiple ones as well and can also be used for more complex VNFs, such as vIMS or vEPC.

NFV Orchestrator

The NFV Orchestrator has two main purposes: the lifecycle management of a network service and the orchestration of NVFIs across multiple VIMs. It essentially can control the management of network services via onboarding and VNF packages as well as the instantiation and scaling of network services, creating an end-to-end service for providers and operators.

The NFV Orchestrator also manages the topology of network services and different policies that can be applied to a network service and the VNF level. Having connections to the NFVI through the VIMs, it manages NFVI resources that include distribution, reservation, and allocation to a network service or VNF instance. In this way, the NFV Orchestrator has global control over resources across VIM instances, and, based on resource usage and various rules, it can implement a different policy management for network services and VNFs by optimizing the use of these resources.

Challenges Surrounding VNF Onboarding

The goal, and the main gain, when adopting a virtualized infrastructure and software-based network is building up a cloud-native network with a high level of automation and simplified

operation. In principal, this can be potentially reached by choosing the journey to NFV, but there are some challenges that do need to be addressed.

The main focus should be achieving a dynamic orchestration of the virtual infrastructure and choosing the right solution when it comes to VNF lifecycle management. Even though NFV as a technology is standardized by ETSI, there is lack of clarity in the distribution of function in the MANO layer—such as between the NFVO and the VNF Manager. To a certain extent, this creates an environment for innovation and encourages technology to develop further. But it also leads to different interpretations of standards by different vendors, different product implementations, and arbitrary distribution of functions across the orchestration layer.

There are two approaches when choosing a VNF Manager: dedicated per a specific VNF or a generic VNF Manager that can handle the lifecycle management of multiple VNFs (directly increasing the number of northbound integrations towards NFVO and southbound integrations towards VIM). Consequently, the chosen approach affects the level of automation and the complexity of the configuration, especially if there are some vendor-specific functions. Different vendors' interpretations can also result in the duplication of functions in a multi-vendor environment. Using a generic VNF Manager significantly decreases the complexity of MANO given there is no need to manage a number of different VNFMs. However, it can also lead to vendor lock-in, meaning a provider is highly dependant on that one solution.

Additionally, NFV renders the current operating model obsolete and brings a different approach to service delivery in a DevOps format that service providers need to adopt to. This is why it is important to have a management system in place that can deal with frequent changes of VNFs during their lifecycle. It is also important to have a common way of modeling VNFs in order to achieve proper automation. Otherwise, confusion will prevail and mistakes will be made in the operation of VNFs.

The key components to modelling VNFs are VNFD (Virtual Network Function Descriptor) and an automation component. VNFD explains operational instructions, policies, lifecycle automation for a particular VNF, auto-scale options, performance indicators, and rules and triggers, while the automation component constructs an automation routine and executes it. The latter also deals with exceptions and possible fails in the process.

Solving the Complexity of VNF Onboarding

In order to support a cloud-native infrastructure, a typical VNF Manager needs to support automation and a fast deployment of VNFs as well as upgrades and updates. Additionally, it needs to offer simplification of management and easy scaling, depending on service demands—and all of this at a low cost.

Most vendors are guided by these requirements but end up offering different solutions due to the lack of industry standardization. Because of this, it is best to choose a VNF Manager that can be independent and integrated into any VNF. In this way, a solution can perform all generic functions and also deal with automation, configuration, and scaling out of the box. A lack of standardization can also be overcome by using solutions that can collaborate somewhat with other vendors and service providers through different organizations or open-source solutions. Cloud-native functionality can be supported off the shelf in this way, since the VNF Manager has already been tested on different vendor-specific VNFs. ONAP is a good example of such an open-source platform and is supported and pushed by major players.

Since the VNF Manager plays a central role in MANO, interoperability and integration should be carefully analyzed. And with a number of service providers often using a multi-vendor environment, a generic VNF Manager can be a very good solution in order to manage all VNFs within one domain from a single location. This also simplifies the number of integrations that occur on the northbound and southbound interfaces.

Achieving such a universal configuration management, so that VNFs are configured using a common VNF model, means providers can better handle the challenges of modeling and service automation as well. Such a universal configuration can be accomplished with a model-based language such as TOSCA, where VNF topologies, interfaces, lifecycle events, and virtual infrastructure requirements can all be described in templates. Templates are not related to any specific infrastructure or VIM and can be reused numerous times for different VNFs. They can explain service components, relationships between components of the service, and different dependencies and capabilities of each component, making it easier to model each VNF in the same universal manner and allowing for easy configuration of a large number of VNFs.

VNF Managers that support an open-source solution are also a perfect way of overcoming any issue with proprietary vendor deployment, resulting in the easier use of VNF Managers for various VNFs. A perfect example that supports most of the solutions to the complexity of VNF onboarding can be found in Cloudify's Orchestration and their VNF onboarding program.

Conclusion

Software-based network technologies are without a doubt, bringing evolution to NFV. Many challenges are thus on their way, but even more opportunities and benefits will arrive as well. And because of these benefits, it is only a matter of time until all service providers transform their networks and operate their own private clouds to deploy services faster, autonomously, and more agile than possible today. And to achieve all of this, service providers must introduce new capabilities in order to properly manage VNFs.

Most of the challenges in such a transformed and software-based network will be found in how management and orchestration are implemented and deployed. With the current lack of standardization, tools for modeling VNFs, and support for different topologies, the road to being a fully cloud-native and automated network will be very bumpy indeed. Various vendor-specific and open-source implementations of NFV management and orchestration are available. And there are different approaches in how VNF Managers should be deployed (single generic or multiple ones) as well as many different tools for automation.

This variety is why service providers need to choose a more generic approach to simplify management and orchestration and also gain all the potential benefits of a cloud network. This approach will decrease the number of integration points, provide decoupling of physical and virtual infrastructures, be as vendor-independent as possible, and provide a universal model of a network in order to achieve the best possible automation as well.