

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

**Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE**
Domaine : Mathématiques et Informatique
Filière : Informatique

Spécialité : Réseaux, Mobilités et Systèmes Embarqués

Présenté par
**MOUFFOK Nesrine
YACOUBI Nassima**

Thème

**Déploiement d'une architecture
Openstack pour une infrastructure NFV
- Service Orchestration - au sein d'ATM
Mobilis**

Mémoire soutenu publiquement le 24/09/2018 devant le jury composé de :

Présidente : Mme AOUDJIT Rachida
Examineur : Mr RAMDANE Mohamed
Encadreur : Mme BOUHMADOUCHE Imene
Co-encadreur : Mr DAOUI Mehammed

Dédicaces

*Je dédie ce modeste travail à :
Mes très chers parents qui sont la source
de ma réussite.*

*Merci pour vos instructions, votre
soutien, que le tout Puissant vous
Accorde une longue vie papa, maman.*

*Vos prières et vos conseils m'ont
toujours accompagnée et m'ont éclairée
le chemin.*

*A ma grand-mère Sadia que Dieu la
garde pour moi*

*Aux personnes dont j'ai bien aimé la
présence dans ce jour, à
tous mes frères, sœur et belles sœurs qui
n'ont cessé d'être pour moi des exemples
de persévérance, de courage et de
générosité.*

A ma chère binôme Nesrine

A mon promoteur DAOUI Mehammed

A tous mes amies.

Et à tous ceux qui me sont chers.

-Nassima-

Dédicaces

A la mémoire de mes grands-parents, ma tante et mon oncle

Puisse Dieu les accueillir dans son infinie Miséricorde

Je dédie ce mémoire à :

Mes chers parents, que nulle dédicace ne puisse exprimer mes sincères sentiments pour leur patience illimitée, leur encouragement, leur aide, en témoignage de mon profond amour et respect pour leur grand sacrifice.

Que Dieu vous garde pour nous Maman, Papa.

A Mes modèles et exemples, mon sang, Mes chers frères : Yazid, Belkacem et Radouane pour leur grand amour et leur soutien qu'ils trouvent ici l'expression de ma haute gratitude.

A Ma grand-mère qui a toujours été présente pour les bons conseils, son affection et son soutien m'ont été d'un grand secours au cours de ma vie.

A Tous les membres de ma famille petits et grands veuillez trouver dans ce modeste travail l'expression de mon affection.

A ma chère binôme Nassima

A tous mes amis (es) et camarades.

A mon promoteur Mr DAOUI Mehammed dont le professionnalisme n'a d'égal.

-Nesrine-

Remerciement

*Tout d'abord, nous tenons à remercier le **Bon Dieu** tout puissant pour nous avoir donné patience, courage, volonté et l'intelligence nécessaire pour accomplir ce modeste travail et nous avoir permis de le mener à bien.*

*Nous tenons à exprimer notre immense gratitude à l'encadreur de ce travail, Monsieur **DAOUI Mehammed**, Professeur à l'université Mouloud Mammeri Tizi Ouzou, qui est l'exemple de chercheur passionné que l'on souhaite devenir un jour.*

Nos remerciements vont à lui, pour avoir guidé nos premiers pas dans la recherche et pour nous avoir encadrés.

Nous le remercions vivement pour la confiance qu'il nous a accordés, ses conseils et remarques constructives qui nous ont permis d'améliorer la qualité de notre travail.

*Nous souhaitons adresser nos remerciements les plus sincères à Madame **BOUHMADOUCHE Imene** et Mademoiselle **ZENOUNE Safa** pour avoir codirigé ce travail et de suivre la réalisation de ce mémoire au sein d'ATM Mobilis.*

*Nous ne saurons jamais assez remercier Monsieur **Jean Michel MEULIEN** pour le temps qu'il nous a consacré, son aide, ses conseils, et ses orientations. Qu'il trouve ici l'expression de notre profonde gratitude.*

*Nos plus vifs remerciements vont également à Madame **BOUZEFRANE Samia** Maître de conférences au CNAM Paris pour sa présence et ses explications.*

*Nous adressons nos plus sincères remerciement à Madame **AOUDJIT Rachida** et Mr **Ramdane Mohammed** pour l'intérêt qu'ils ont porté à notre travail en examinant ce mémoire et en participant à ce jury.*

Table des matières

Dédicaces

Remerciement

Sommaire I

Liste des abreviations II

Liste des figures..... III

Introduction générale 1

Chapitre I : Virtualisation et Cloud Computing

I.1. Introduction 2

I.2. La virtualisation..... 3

I.2.1. Historique..... 3

I.2.2. Définition de la virtualisation 4

I.2.3. Différents domaines d’application de la virtualisation 5

I.2.3.1. Virtualisation du matériel 5

I.2.3.2. Virtualisation du serveur..... 5

I.2.3.3. Virtualisation du stockage..... 5

I.2.3.4. Virtualisation d’application..... 6

I.2.3.5. Virtualisation de réseau..... 6

I.2.3.6. Virtualisation de postes de travail 6

I.2.4. Méthodes de virtualisation..... 7

I.2.4.1. La paravirtualisation 7

I.2.4.2. La virtualisation complète..... 7

I.2.5. Avantages et inconvénients de la virtualisation..... 7

I.2.5.1. Avantages 7

I.2.5.2. Inconvénients..... 8

I.3. Cloud computing 9

I.3.1. Historique..... 9

I.3.2. Définition du Cloud Computing..... 10

I.3.3. Modèles de déploiements du cloud computing	10
I.3.3.1. Cloud public	10
I.3.3.2. Cloud privé.....	11
I.3.3.3. Cloud hybride	12
I.3.4. Modèles de services de cloud computing.....	12
I.3.4.1. Infrastructure-as-a-service (IaaS).....	12
I.3.4.2. Plate-forme-as-a-service (PaaS).....	13
I.3.4.2. Logiciel-as-a-service (SaaS)	13
I.3.5. Avantages du cloud computing.....	13
I.3.6. Inconvénients du cloud computing	14
I.4. Conclusion.....	16

Chapitre II : Technologies SDN, NFV et Solution Openstack

II.1 Introduction.....	17
II.2. Software Defined Network (SDN).....	17
II.2.1. Définition du SDN	17
II.2.2. Le principe du SDN.....	17
II.2.3. Les différentes couches des réseaux SDN.....	18
II.2.4. Les interfaces de communication :.....	19
II.3. Network Functions Virtualisation (NFV)	19
II.3.1. Définition de la NFV.....	19
II.3.2. L'architecture NFV	20
II.3.2.1. L'infrastructure NFV (NFVI).....	21
II.3.2.2. Les fonctions Réseau Virtuels VNF.....	21
II.3.2.3. NFV MANO	22
II.3.3. Avantages et inconvénients de NFV	23
II.3.3.1. Avantages de NFV	23
II.3.3.2. Inconvénients de NFV.....	26
II.4. Différences entre SDN et NFV.....	27
II.5. Openstack.....	28

II.5.1. Définition d'Openstack	28
II.5.2. Architecture d'openstack	28
II.5.2.1. Les services d'openstack	29
II.5.2.2. Autres Services d'Openstack	30
II.5.2.3. Les Services d'orchestration d'Openstack	30
II.6. OPNFV	31
II.7. Conclusion	31

Chapitre III : NFV-MANO et Tacker

III.1. Introduction.....	32
III.2. NFV MANO Management and Orchestration	32
III.2.1. Définition du NFV MANO	32
III.2.2. Les blocs fonctionnels du cadre architectural NFV-MANO	32
III.2.2.1. Virtualized Infrastructure Manager (VIM)	34
III.2.2.2. Gestionnaire de fonction de réseau virtuel (VNFM).....	34
III.2.2.3. Orchestrateur NFV (NFVO).....	35
III.2.2.4. Groupe de référentiels	36
I.2.2.5. Autres blocs fonctionnels.....	37
III.3. Tacker	39
III.3.1. Historique de Tacker	39
III.3.2. Définition de Tacker	40
III.3.3. Tacker Architecture.....	41
III.3.3.1. Les composants de Tacker	42
III.3.4. Cas d'utilisation de Tacker	45
III.4. Conclusion :	45

Chapitre IV : Implémentation et mise en œuvre

IV.1. Introduction	46
IV.2. Préparation de l'environnement de travail.....	46

IV.3. Déploiement d'Openstack	47
IV.4. Définition de Devstack	47
IV.5. Installation d'Openstack	47
IV.5.1. Pré-requis du système	47
IV.5.2. Création d'un utilisateur stack.....	48
IV.5.3. Téléchargement de Devstack	49
IV.5.4. Configuration des fichiers de DEVSTACK	49
IV.5.5. Création du fichier de configuration local.conf.....	50
IV.5.6. Activer les plug-ins Devstack liés à Tacker dans le fichier local.conf	51
IV.5.7. Démarrage de l'installation	52
IV.6 Orchestration	55
IV.6.1. Création d'un VIM Management	55
IV.6.2. Création et configuration du VNF	56
IV.6.2.1. Configuration du réseau :	56
IV.6.2.2. Création de VNF Catalog.....	60
IV.6.2.3. Création de VNF Manager	64
IV.7.Conclusion :	69
Conclusion générale	70
Références Bibliographiques	
Annexes	

A

API Application Programming Interface

B

BSS Business Support System

C

CMS Conversational Monitor System

COTS Commercial Off The Shelf

CP Connexion Point

CRM Customer Relationship Management

D

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

E

EC2 Elastic Compute Cloud

EM Element Management

ESX Vmware Elastic Sky X

ETSI Européen
Telecommunications Standards
Institute

ETSI ISG NFV European
Telecommunications Standards Institute
Industry Specification Group

ESXI Elastic Sky X Integrated

F

FCAPS Fault, Configuration,
Accounting, Performance, Security

I

IaaS Infrastructure as a Service

IBM International Business Machines

K

KVM Keyboard -Video – Mouse

M

MANO Management and
Orchestration

M2M Machine to Machine

N

NAS Network Attached Storage

NFV Network Functions Virtualisation

NFVI Network Function Virtualisation
Infrastructure

NFVO Network Function Virtualisation Orchestrator

NS Network Service

NSD Network Service Descriptor

O

OSS Operations Support System

OPNFV Open Platform for Network Function Virtualization

P

PaaS Platform as a Service

PC Personal Computer

PoP Point Of Presence

Q

Qemu Quick Emulator

QoS Qualité of service

R

RAM Random Access Memory

REST Representational State Transfer

ROM Read-Only Memory

S

SaaS Software as a Service

SAN Storage Area Network

SDN Software Defined Network

S3 Simple Storage Service

T

TOSCA Topology and Orchestration Specification for Cloud Applications

V

VcE Virtual Customer Edge

vCPE Virtual Customer Premises

VDI Virtual Device Interface

VDU Virtual Deployment Unit

VIM Virtualized Infrastructure Manager

VL Virtual Link

VLAN Virtual Local Area Network

VM Virtual Machine

VNF Virtualised Network Function

VNFD Virtual Network Function Descriptor

VNFFGD Virtual Network Function Forwarding Graph Descriptor

Equipement

VNFM Virtual Network Function Manager

VPE Virtual Provider Edge

Liste des figures

Figure I.1. Différence entre architecture standard et virtualisée	4
Figure I.2. Cloud Public.....	11
Figure I.3. Cloud Privé	11
Figure I.4. Cloud Hybride.....	12
Figure II.1. Architecture SDN	18
Figure II.2. Architecture NFV	20
Figure II.3. NFV Infrastructure (NFVI)	21
Figure II.4. Chainage de service réseau virtuel	22
Figure II.5. Architecture Openstack	28
Figure III.1. Le cadre architectural NFV-MANO	33
Figure III.2. Tacker project evolution in OpenStack.....	40
Figure III.3. Tacker Architecture.....	42
Figure IV.1. Environnement de travail.....	46
Figure IV.2. Dashboard d'Openstack	54
Figure IV.3. Vue d'ensemble du Dashboard d'Openstack.....	55
Figure IV.4. Etapes de création d'une VIM	56
Figure IV.5. Graphique de la topologie du Réseau	56
Figure IV.6. Topologie du réseau	56
Figure IV.7. Etapes de création d'un réseau public.....	58
Figure IV.8. Etapes de création d'un réseau privé	59
Figure IV.9. Etapes de création d'un Routeur	60
Figure IV.10. Unité de déploiement virtuelle (VDU)	61
Figure IV.11. Lien Virtuel (VL)	61
Figure IV.12. Point de connexion (CP)	62
Figure IV.13. Création du VNFD Firewall.....	63
Figure IV.14. Création du VNFD Dhcp	64
Figure IV.15. Création du VNFD DNS	64
Figure IV.16. Déploiement du VNF Firewall.....	65
Figure IV.17. Déploiement du VNF DHCP	66

Figure IV.18. Déploiement du VNF DNS	67
Figure IV.19. Lancement de VNF	68
Figure IV.20. Instances créés.....	68
Figure IV.21. Graphe de la topologie du réseau VNF.....	69

Introduction générale

Introduction générale

Nous sommes dans un monde de plus en plus connecté, de plus en plus relié aux outils informatiques. L'infrastructure réseau doit prendre en charge tous ces nouveaux périphériques en fournissant une connectivité adéquate et des services basés sur les applications. Les fournisseurs de services doivent investir davantage dans l'infrastructure réseau pour répondre aux besoins croissants des clients. Pour cela, un nouveau concept de partage d'infrastructure entre les fournisseurs de services a vu le jour pour réduire les coûts d'investissement excessifs liés au déploiement de l'infrastructure.

De nouvelles architectures basées sur la virtualisation du réseau émergent pour permettre le partage de réseau, gérer l'explosion du Big Data à partir des différents périphériques et simplifier les tâches de gestion

Notre projet de fin d'étude est réalisé dans le cadre du mémoire de Master en Réseaux Mobilités et Systèmes Embarqués « RMSE » ayant comme objectif principal : Déploiement d'une architecture Openstack pour infrastructure NFV – Service Orchestration au profit d'ATM Mobilis. Principalement, mettre en place une solution de gestion et d'orchestration de NFV.

Notre mémoire est organisé en quatre chapitres répartis comme suit :

Après une brève introduction, dans laquelle nous cernons la thématique, de manière globale, le but du projet et les solutions envisageables, nous entamons notre mémoire par le premier chapitre qui est consacré à la présentation des notions générales sur la virtualisation et le Cloud Computing.

Nous enchainons notre mémoire par le deuxième chapitre qui donne un aperçu sur les technologies SDN « *Software Defined Networking* », NFV « *Network Functions Virtualization* » ainsi que OpenStack, la solution sur laquelle nous avons déployé notre projet.

Le troisième chapitre est dédié à la présentation de NFV-MANO et le projet Tacker d'Openstack, leurs architectures et leurs différents composants

Enfin, le quatrième chapitre est consacré à la partie pratique de notre travail. Il présente l'installation et la configuration de notre application, ainsi que les tests et les résultats obtenus.

Chapitre I

Virtualisation et Cloud Computing

I.1. Introduction

Alors que les données informatiques augmentent de façon exponentielle, et que les entreprises font de plus en plus appel au processus informatique pour gagner en productivité et en compétitivité, la possible réduction des coûts de gestion des infrastructures informatiques est une des principales priorités des entreprises

Ces dernières années, plusieurs moyens sont apparus pour aborder cette réduction des coûts parmi lesquelles, la virtualisation, et le cloud computing.

La virtualisation et le cloud computing sont deux concepts différents, mais pourtant complémentaires. Le cloud computing est le fruit des évolutions récentes des technologies de l'information, notamment la virtualisation, et de l'augmentation de la capacité des réseaux. Si le Cloud utilise en général les technologies de virtualisation, il incarne surtout une nouvelle approche des infrastructures matérielles et logicielles.

Au niveau de ce chapitre, nous allons présenter les notions fondamentales de la virtualisation et du Cloud Computing. Nous allons étudier dans un premier lieu la virtualisation, son historique puis sa définition, ses différents domaines d'application, ses méthodes ainsi que ses avantages et inconvénients. Nous aborderons par la suite le cloud computing, son historique et sa définition, suivi de ses modèles de déploiement, les différents services qu'il fournit, ainsi que ses avantages et inconvénients.

I.2. La virtualisation :

I.2.1. Historique [1] :

La virtualisation remonte aux années 1960. A l'époque, c'est la firme IBM qui crée le premier système de virtualisation de serveur. Dans ce contexte, l'informatique est peu présente et les rares sociétés qui possèdent des systèmes informatiques sont équipées de gros calculateurs, les Mainframe.

Déjà à cette époque, les soucis d'optimisation des ressources matérielles d'une machine se posaient. En effet, les supers calculateurs sont souvent sous utilisés. IBM développe alors un produit VM/CMS (Virtual Machine / Conversational Monitor System), un système de virtualisation de serveurs.

Au cours des années 80-90 apparaît l'architecture x86 et les PC se déploient auprès d'un grand nombre d'utilisateurs. Le besoin de virtualiser pour optimiser les machines se faisait moins sentir.

Mais, dans les années 90-2000, VMware réussit à virtualiser un poste x86. Ceci ouvre la porte à plus de possibilité et relance l'envie pour les sociétés informatiques de développer de nouvelles fonctionnalités pour optimiser et offrir plus de flexibilité.

A l'heure actuelle, la virtualisation est très connue. On entend parler de virtualisation de serveur, de Virtual box, de baremetal, mais aussi de virtualisation de poste de travail, de VDI, et de virtualisation dans les jeux-vidéos avec les émulateurs.

En 2012, trois grandes sociétés se partagent le marché de la virtualisation en entreprise :

- ❖ VMware qui est le leader
- ❖ Citrix, très fort dans la virtualisation de poste de travail
- ❖ Microsoft, qui s'aligne sur la concurrence

I.2.2. Définition de la virtualisation [2] :

Le principe de virtualisation consiste à faire fonctionner sur une seule machine physique, plusieurs systèmes d'exploitation ou plusieurs applications séparément les uns(es) des autres, comme s'ils fonctionnaient sur des machines physiques distinctes.

La virtualisation fait référence à la création d'une ressource virtuelle telle qu'un serveur, un ordinateur de bureau, un système d'exploitation, un fichier, un stockage ou un réseau.

Sans virtualisation, un seul système peut être opérationnel sur une machine physique alors qu'avec la virtualisation, il est possible d'en faire fonctionner plusieurs.

La couche de virtualisation appelée hyperviseur masque les ressources physiques d'un équipement matériel pour proposer à un système d'exploitation des ressources différentes de ce qu'elles sont en réalité. Elle permet en outre de rendre totalement indépendant un système d'exploitation du matériel sur lequel il est installé, ce qui ouvre de grandes possibilités.

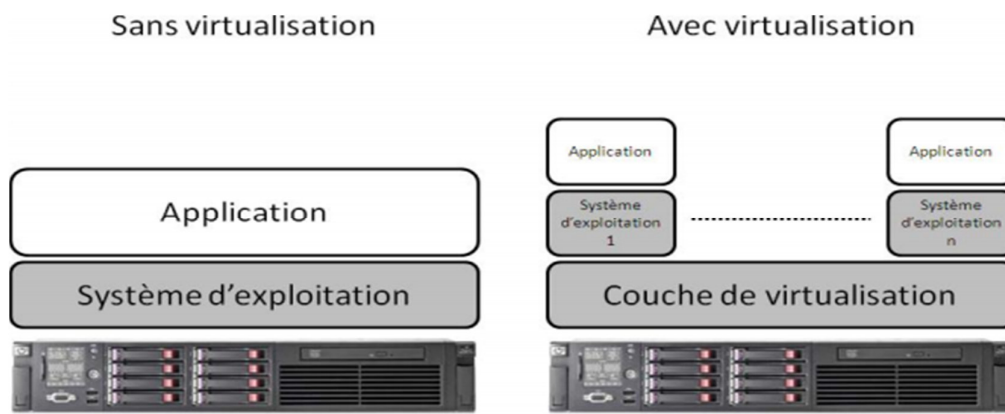


Figure I.1. Différence entre architecture standard et virtualisée [2]

L'objectif principal de la virtualisation est de gérer les charges de travail en transformant radicalement l'informatique traditionnelle pour la rendre plus évolutive.

I.2.3. Différents domaines d'application de la virtualisation :

Initialement orientée sur les serveurs, la virtualisation couvre plusieurs domaines de l'infrastructure informatique d'une entreprise. On distingue aujourd'hui les domaines d'application suivants :

- Virtualisation du matériel
- Virtualisation du serveur
- Virtualisation du stockage
- Virtualisation d'application
- Virtualisation des réseaux
- Virtualisation des postes de travail

I.2.3.1. Virtualisation du matériel

C'est le type de virtualisation le plus courant aujourd'hui. La virtualisation du matériel est rendue possible par un gestionnaire de machine virtuelle (VM) appelé «hyperviseur». L'hyperviseur crée des versions virtuelles d'ordinateurs et de systèmes d'exploitation et les consolide en un seul grand serveur physique, de sorte que toutes les ressources matérielles peuvent être utilisées plus efficacement. Il permet également aux utilisateurs d'exécuter différents systèmes d'exploitation sur la même machine en même temps. [2]

I.2.3.2. Virtualisation du serveur

La virtualisation de serveurs est le fait de créer plusieurs serveurs virtuels sur un serveur physique. Ces derniers tournent sur la même machine physique, tout en ayant les mêmes propriétés que s'ils sont sur une machine physique. Le but de cette manipulation est d'optimiser les performances d'un serveur physique tout en permettant à l'entreprise de faire des économies en investissement en infrastructures physiques et en maintenance. [2]

I.2.3.3. Virtualisation du stockage

La virtualisation du stockage est la mise en commun de stockage physique de multiples périphériques de stockage réseau dans ce qui semble être un dispositif de stockage unique, qui est géré depuis une console centrale.

La virtualisation du stockage est couramment utilisé dans un réseau de stockage (SAN/NAS). [3]

I.2.3.4. Virtualisation d'application

La virtualisation d'application permet de dissocier l'application du système d'exploitation hôte et des autres applications présentes afin d'éviter les conflits. Il s'agit d'un processus dans lequel les applications sont virtualisées et livrées depuis un serveur vers l'appareil de l'utilisateur final, tel qu'un ordinateur portable, un smartphone ou une tablette. Ce type de virtualisation est populaire pour les entreprises qui nécessitent l'utilisation de leurs applications en déplacement. [4]

I.2.3.5. Virtualisation de réseau

La virtualisation de réseau est une méthode qui combine tout l'équipement de réseau physique en une seule ressource. C'est le processus de division de la bande passante en plusieurs canaux indépendants, chacun pouvant être assigné aux serveurs et aux dispositifs en temps réel. Les entreprises qui bénéficieraient de la virtualisation de réseau sont celles qui ont un grand nombre d'utilisateurs et qui ont besoin de maintenir leurs systèmes opérationnels à tout moment. Avec les canaux distribués, la vitesse de réseau augmentera considérablement, ce qui permet de fournir des services et des applications plus rapidement qu'avant. [4]

I.2.3.6. Virtualisation de postes de travail

Similaire à la virtualisation d'application mentionnée ci-dessus, la virtualisation de postes de travail sépare l'environnement de bureau du périphérique physique et est configurée comme une «infrastructure de bureau virtuel» (VDI). Les principaux avantages de la virtualisation de postes de travail sont que les utilisateurs peuvent accéder à tous leurs fichiers et applications personnels depuis n'importe quel endroit et sur n'importe quel PC, ce qui signifie qu'ils peuvent travailler n'importe où sans avoir à apporter leur ordinateur de travail. Il réduit également le coût des licences pour l'installation de logiciels sur les postes de travail et la gestion des correctifs (vulnérabilités, mise à jour...) est très simple, puisque tous les postes de travail virtuels sont hébergés au même endroit. [1]

I.2.4. Méthodes de virtualisation [5]

I.2.4.1. La paravirtualisation

La paravirtualisation est une technique de virtualisation qui présente à la machine invitée une interface logicielle similaire mais non identique au matériel réel. Ainsi, elle permet aux systèmes d'exploitation invités d'interagir directement avec le système d'exploitation hôte et donc ils seront conscients de la virtualisation.

I.2.4.2. La virtualisation complète

La virtualisation complète (en anglais full-virtualization) est une technique de virtualisation qui permet de créer un environnement virtuel complet. En utilisant cette technique, le système d'exploitation invité n'interagit pas directement avec le système d'exploitation hôte et donc il croit s'exécuter sur une véritable machine physique.

Cette technique de virtualisation ne permet de virtualiser que des SE de même architecture matérielle que l'hôte.

I.2.5. Avantages et inconvénients de la virtualisation[7]

I.2.5.1. Avantages :

Depuis de nombreuses années, les performances des équipements informatiques n'ont cessées d'évoluer pour atteindre aujourd'hui une puissance extraordinaire. Les applications proposées de nos jours ont besoin de beaucoup de ressources mais paradoxalement n'utilisent qu'une fraction du potentiel de certains serveurs. Selon Microsoft, il est souvent possible de regrouper jusqu'à 5 serveurs sur une seule machine sans perte de performances. La virtualisation apporte donc de nombreux avantages :

- ❖ Consolidation et rationalisation d'un parc de serveurs en entreprise : les entreprises ne sont plus obligées d'acheter un serveur physique pour chaque application,
- ❖ Rationalisation des coûts de matériels informatiques,
- ❖ Possibilité d'installer plusieurs systèmes (Windows, linux) sur une même machine,
- ❖ Portabilité des serveurs : une machine virtuelle peut être déplacée d'un serveur physique vers un autre (lorsque celle-ci a, par exemple, besoin de plus de ressources),
- ❖ Accélération des déploiements de systèmes et d'applications en entreprise,
- ❖ Administration simplifiée de l'ensemble des serveurs,

- ❖ Réduction de la consommation électrique en diminuant le nombre de machines physiques.

I.2.5.2. Inconvénients

- ❖ Coûts importants : pour faire fonctionner convenablement une architecture virtualisée, l'entreprise doit investir dans un serveur physique disposant de plusieurs processeurs et de beaucoup de mémoire.
- ❖ Pannes généralisées : si le serveur physique tombe en panne, les machines virtuelles tombent également en panne.
- ❖ Vulnérabilité généralisée : si l'hyperviseur est bogué ou exposé à une faille de sécurité, les machines virtuelles peuvent l'être également et ne sont plus protégées. En augmentant les couches logicielles, la virtualisation a pour conséquence d'augmenter la surface d'attaque de l'entreprise.

I.3. Cloud computing

I.3.1. Historique : [6]

La bulle Internet atteint son apogée le 10 mars 2000, puis éclate progressivement au cours des semaines suivantes, avec la vente massive d'actions par des grands noms de la technologie de pointe, tels que Dell et Cisco. Pour continuer à survivre, les entreprises doivent repenser ou ajuster leur modèle commercial et leurs offres destinés aux clients. Parmi les plus récentes, nombreuses sont celles qui décident de proposer des services basés sur Internet, plutôt que de l'utiliser comme moyen de passer commande ou de communiquer avec les clients. La fin des années 1990 et le début des années 2000 représentent une période propice pour créer une entreprise en ligne ou investir dans une telle activité. Avec le développement de l'architecture multi-tenante, l'omniprésence du haut débit et la mise en place de normes d'interopérabilité universelles entre les logiciels, c'est le cadre idéal pour permettre au Cloud Computing de décoller. Salesforce.com est lancé en 1999. Il est le premier site à proposer des applications d'entreprise à partir d'un simple site Web standard, accessible via un navigateur Web : c'est ce qu'on appelle aujourd'hui le Cloud Computing. Amazon.com lance Amazon Web Services en 2002. Ce nouveau service permet aux utilisateurs de stocker des données et tire profit des compétences d'un très grand nombre de personnes pour de très petites tâches (par exemple, sur Amazon Mechanical Turk). Facebook est fondé en 2004 et révolutionne la façon dont les utilisateurs communiquent et stockent leurs propres données (photos et vidéos), en faisant involontairement du Cloud un service personnel. En 2006, Amazon développe ses services Cloud. Le premier à voir le jour est Elastic Compute Cloud (EC2), qui permet aux utilisateurs d'accéder à des ordinateurs et d'y exécuter leurs propres applications, le tout sur le Cloud. Le deuxième service lancé est Simple Storage Service (S3). Il permet d'introduire le modèle de paiement à l'utilisation auprès des clients et du secteur en général, modèle qui représente désormais une pratique courante. Salesforce.com lance ensuite Force.com en 2007. Cette plate-forme en tant que service (PaaS) permet aux développeurs de concevoir, de stocker et d'exécuter toutes les applications et tous les sites Web nécessaires à leurs activités sur le Cloud. Google Apps arrive en 2009 et permet à ses utilisateurs de créer et de stocker des documents entièrement sur le Cloud. Plus récemment, les entreprises de Cloud Computing ont cherché à accroître davantage l'intégration de leurs produits. En 2010, salesforce.com lance sa base de données Cloud avec Database.com pour les développeurs, marquant ainsi le développement des services de Cloud Computing utilisables sur n'importe quel terminal, exécutables sur n'importe quelle plate-forme et écrits dans n'importe quel langage de programmation.

I.3.2. Définition du Cloud Computing[8]

En termes simples, le cloud computing est la fourniture de services informatiques (serveurs, stockage, bases de données, réseaux, logiciels) sur Internet. Les entreprises offrant ces services informatiques sont appelées «fournisseurs de cloud» et facturent généralement les services d'informatique en fonction de leur utilisation, de la même manière que la facturation de l'eau ou de l'électricité à domicile.

Les premiers services de cloud computing ont à peine dix ans, mais déjà une variété d'organisation : des petites start-up aux multinationales, des agences gouvernementales aux organismes sans but lucratif adoptent la technologie pour toutes sortes de raisons. Voici quelques-unes des choses que nous pouvons faire avec le nuage :

- ❖ Créer de nouvelles applications et de nouveaux services
- ❖ Stocker, sauvegarder et récupérer des données
- ❖ Héberger des sites Web et des blogs
- ❖ Distribuer les flux audio et vidéo
- ❖ Livrer des logiciels à la demande
- ❖ Analyser les données pour les modèles et faire des prédictions

I.3.3. Modèles de déploiements du cloud computing [9]

Il existe trois manières différentes de déployer des ressources de cloud computing : cloud public, cloud privé et cloud hybride.

I.3.3.1. Cloud public

Les Clouds publics sont détenus et exploités par un fournisseur de services de cloud tiers, qui fournit leurs ressources informatiques comme les serveurs et le stockage sur Internet. Un navigateur web permet aux utilisateurs d'accéder aux services du cloud et à la gestion de leur compte. La mise en place de ce type de Cloud est gérée par des entreprises tierces (Amazon, Google, etc...) et il est accessible selon le modèle pay-as-you-go (payer selon la consommation).

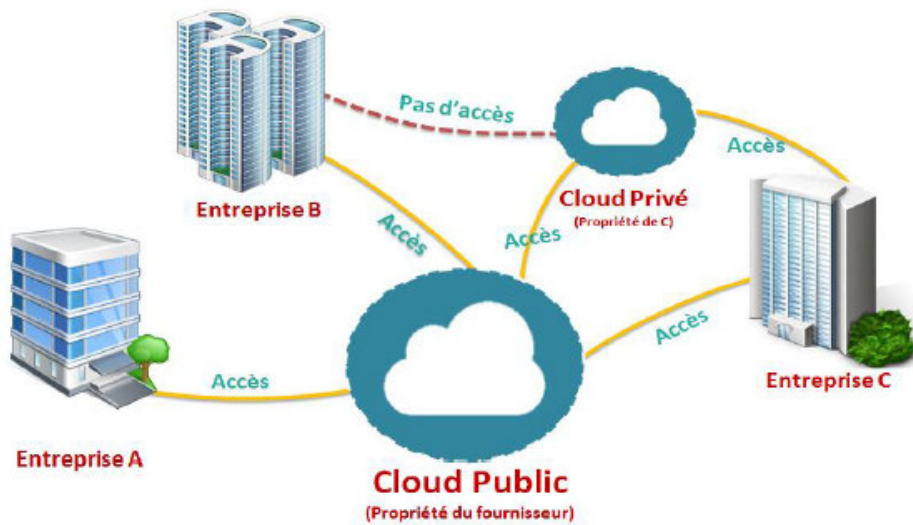


Figure I.2. Cloud Public

I.3.3.2. Cloud privé

Un cloud privé fait référence aux ressources de cloud computing utilisées exclusivement par une seule entreprise ou organisation. Un cloud privé peut être physiquement situé sur le centre de données sur le site de l'entreprise. Certaines entreprises paient également des fournisseurs de services tiers pour héberger leur cloud privé. Un cloud privé est un nuage dans lequel les services et l'infrastructure sont maintenus sur un réseau privé.

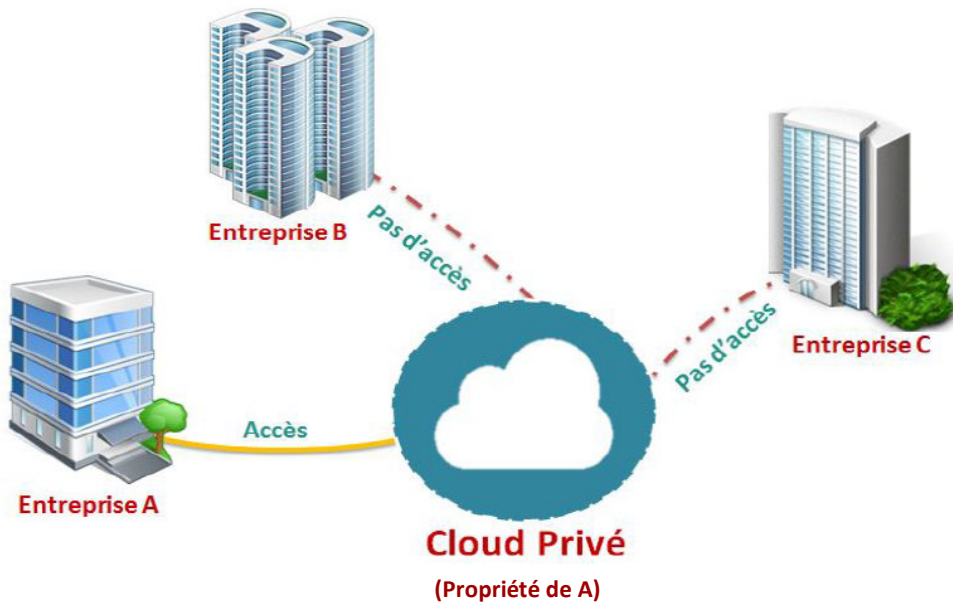


Figure I.3. Cloud Privé

I.3.3.3. Cloud hybride

Les nuages hybrides combinent des nuages publics et privés, reliés entre eux par une technologie qui permet le partage de données et d'applications. En laissant déplacer ces dernières entre les clouds privés et publics, le cloud hybride offre aux entreprises une plus grande flexibilité et des options de déploiement avantageux.

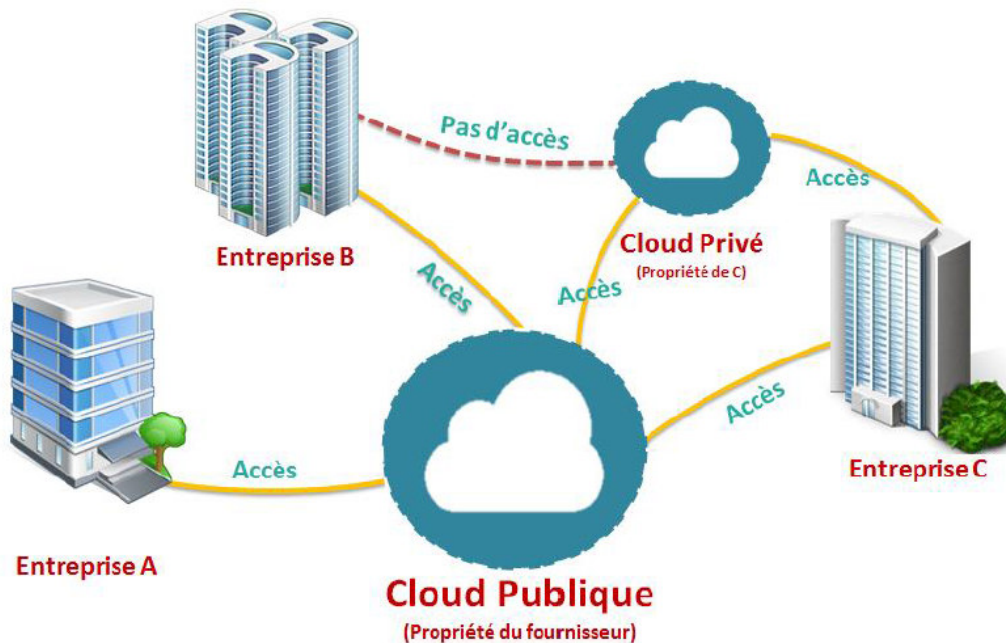


Figure I.4. Cloud Hybride

I.3.4. Modèles de services de cloud computing [8]

La plupart des services d'informatique en nuage se répartissent en trois grandes catégories : infrastructure en tant que service (IaaS), plate-forme en tant que service (PaaS) et logiciel en tant que service (SaaS). On les appelle parfois la pile de cloud computing, car ils se superposent.

I.3.4.1. Infrastructure-as-a-service (IaaS)

C'est le service de plus bas niveau du Cloud computing et la catégorie la plus élémentaire des services d'informatique en nuage. Il permet de louer une infrastructure informatique pouvant héberger et exécuter des applications, des services ou encore stocker des données à partir d'un fournisseur de cloud plutôt que d'acheter des serveurs, des logiciels, et de l'espace dans un centre de traitement de données.

Parmi les prestataires d'IaaS, on peut citer : Amazon avec EC2 ou Orange Business Services avec Flexible Computing.

I.3.4.2. Plate-forme-as-a-service (PaaS)

La plate-forme en tant que service (PaaS), située juste au-dessus de l'IaaS, fait référence aux services d'informatique en nuage qui fournissent un environnement à la demande pour développer, tester, livrer et gérer des applications logicielles. PaaS est conçu pour faciliter la création rapide d'applications Web ou mobiles par les développeurs, sans se préoccuper de la configuration ou de la gestion de l'infrastructure sous-jacente (caché) des serveurs, du stockage, du réseau et des bases de données nécessaires au développement.

Les principaux fournisseurs de PaaS sont : Microsoft avec AZURE, Google avec Google App Engine et Orange Business Services.

I.3.4.2. Logiciel-as-a-service (SaaS)

Le logiciel en tant que service (SaaS) est le service du plus haut niveau du Cloud computing. C'est une méthode de distribution d'applications logicielles sur Internet, à la demande et généralement sur abonnement.

Avec SaaS, les fournisseurs de cloud hébergent et gèrent l'application logicielle, l'infrastructure sous-jacente (caché) et toute maintenance, comme les mises à niveau logicielles et les correctifs de sécurité. Les utilisateurs se connectent à l'application via Internet, généralement avec un navigateur Web sur leur téléphone, tablette ou PC.

Les prestataires de solutions SaaS les plus connus sont Microsoft-offre Office365 (outils collaboratifs) Google- offre Google Apps (messagerie et bureautique).

I.3.5. Avantages du cloud computing [10]

Le cloud computing est un grand changement par rapport à la manière traditionnelle dont les entreprises pensent aux ressources informatiques.

Voici 6 raisons courantes pour lesquelles les entreprises se tournent vers les services de cloud computing.

❖ Coût :

Le cloud computing élimine les dépenses liées à l'achat de matériel et de logiciels et à la mise en place de centres de données sur site.

❖ Vitesse :

La plupart des services de cloud computing sont fournis en libre-service et à la demande, de sorte que même de grandes quantités de ressources informatiques peuvent être provisionnées en quelques minutes, généralement en quelques clics de souris.

❖ Échelle mondiale :

Les avantages des services d'informatique en nuage comprennent la possibilité d'évoluer de manière élastique. Dans le cloud, cela signifie fournir la quantité nécessaire de ressources informatiques.

Par exemple, plus ou moins de puissance de calcul, de stockage, de bande passante au moment voulu et à partir du bon emplacement géographique.

❖ Performance :

Les plus grands services de cloud computing fonctionnent sur un réseau mondial de Datacenter sécurisés, régulièrement mis à niveau vers la dernière génération de matériel informatique rapide et efficace. Cela offre plusieurs avantages par rapport à un seul centre de données d'entreprise, notamment une latence réseau réduite pour les applications et de plus grandes économies d'échelle.

❖ Fiabilité :

Le cloud rend la sauvegarde des données, la reprise après sinistre et la continuité des activités plus facile et moins coûteuse, car les données peuvent être mises en miroir sur plusieurs sites redondants sur le réseau du fournisseur de cloud.

I.3.6. Inconvénients du cloud computing [10]**❖ Temps d'arrêt**

Les fournisseurs de services de cloud computing prenant en charge un certain nombre de clients chaque jour, ils peuvent être dépassés et se heurter à des pannes techniques. Cela peut entraîner la suspension temporaire de vos processus métier. De plus, si la connexion Internet est hors ligne, l'accès aux applications, serveurs ou données du cloud devient impossible.

❖ Sécurité

Bien que les fournisseurs de services cloud mettent en œuvre les meilleures normes de sécurité et certifications industrielles, le stockage de données et de fichiers importants sur des fournisseurs de services externes ouvre toujours des risques. L'utilisation de technologies basées sur le cloud signifie que vous devez fournir à votre fournisseur de services un accès aux données professionnelles importantes. Dans le même temps, être un service public ouvre les fournisseurs de services cloud à des problèmes de sécurité de manière routinière. La facilité d'acquisition et d'accès aux services cloud peut également permettre aux utilisateurs malveillants d'analyser, d'identifier et d'exploiter les failles et les

vulnérabilités d'un système. Par exemple, dans une architecture de cloud multi-locataire où plusieurs utilisateurs sont hébergés sur le même serveur, un pirate informatique peut tenter de pirater les données d'autres utilisateurs hébergés et stockés sur le même serveur. Cependant, de tels exploits et échappatoires ne sont pas susceptibles de faire surface.

❖ Verrouillage du fournisseur

Bien que les fournisseurs de services de cloud computing promettent que le cloud sera flexible à utiliser et à intégrer, le changement des services de cloud computing est quelque chose qui n'a pas encore complètement évolué. Les organisations peuvent trouver difficile de migrer leurs services d'un fournisseur à un autre. L'hébergement et l'intégration d'applications cloud actuelles sur une autre plate-forme peuvent engendrer des problèmes d'interopérabilité (capacité du système) et de support. Par exemple, les applications développées sur Microsoft Development Framework (.Net) peuvent ne pas fonctionner correctement sur la plate-forme Linux.

❖ Contrôle limité

Étant donné que l'infrastructure cloud est entièrement détenue, gérée et surveillée par le fournisseur de services, elle transfère un contrôle minimal au client. Le client ne peut que contrôler et gérer les applications, les données et les services exploités en plus de l'infrastructure principale. Les principales tâches administratives telles que l'accès au shell du serveur, la mise à jour et la gestion du microprogramme peuvent ne pas être transmises au client ou à l'utilisateur final.

Il est facile de voir comment les avantages du cloud computing compensent largement les inconvénients. La réduction des coûts, la réduction des temps d'arrêt et la réduction des efforts de gestion sont des avantages qui parlent d'eux-mêmes.

I.4. Conclusion

À travers ce chapitre nous avons pu développer le vaste sujet sur la généralité en rapport avec le cloud computing et la virtualisation.

Le cloud computing est donc un moyen de délivrer un service informatique ciblé et quantifié à une clientèle précise sans que cette dernière n'ait à investir dans un système d'information dédié à ce service. Selon cette définition, nous avons vu les modèles tels que le PaaS, SaaS, IaaS.

Ces modèles reposent sur des technologies de virtualisation qui permet d'atteindre la flexibilité requise à la réalisation de ces modèles. La virtualisation est une technologie clé du Cloud Computing.

La virtualisation permet d'assurer une très grande souplesse au niveau de ressources annuelles à une solution ou à un client. L'indépendance des solutions matérielles et logiciels permet de donner toute la puissance requise à la bonne exécution du service.

Le Cloud est une technologie jeune et innovante dont l'attractivité est en hausse. Ce procédé peut changer les méthodes de travail, réduire les coûts, modifier des interactions client-fournisseur ainsi que changer le modèle économique.

Chapitre II

Les technologies SDN, NFV et Solution Openstack

II.1 Introduction

Alors que le déploiement d'applications est toujours plus aisé et dynamique, notamment grâce au Cloud, le réseau reste parfois perçu comme un frein. Ce dernier ne semble pas avoir évolué à cause de l'enfermement des utilisateurs dans des solutions propriétaires. En plus chaque modification nécessite des interventions manuelles sans orchestration dans la plupart des organisations et c'est la raison pour laquelle le SDN « *Software Defined Network* » et NFV « *Network Function Virtualization* » sont pris au sérieux aujourd'hui.

Dans ce chapitre, nous présenterons les technologies SDN et NFV ainsi que la solution « Openstack » sur laquelle nous allons déployer notre projet.

II.2. Software Defined Network (SDN)

II.2.1. Définition du SDN

Le Software Defined Network ou les réseaux définis par logiciels (SDN) est un principe d'architecture des fonctions réseau et des systèmes qui les pilotent, qui facilite la gestion de ces fonctions réseau virtualisées permettant de tirer un meilleur parti des réseaux et de mettre en œuvre des architectures dynamiques.

II.2.2. Le principe du SDN

Les réseaux définis par logiciels (Software Defined Network (SDN)) ont introduit une séparation entre le plan de données et le plan de contrôle pour rendre le réseau programmable [Das et al, 2012].

Le plan de données définit les équipements du réseau et les connexions entre eux tandis que le plan de contrôle définit le comportement de ce réseau. [11]

Le plan de contrôle est placé dans un contrôleur centralisé qui a une vision sur la topologie du réseau. Le plan de données réside sur le commutateur ou le routeur. [11]

Le SDN se traduit par une architecture émergente dynamique, maintenable, économique et adaptable, ce qui la rend idéale pour les applications modernes à forte bande passante et très dynamiques.

La figure II.1 illustre l'architecture des réseaux SDN.

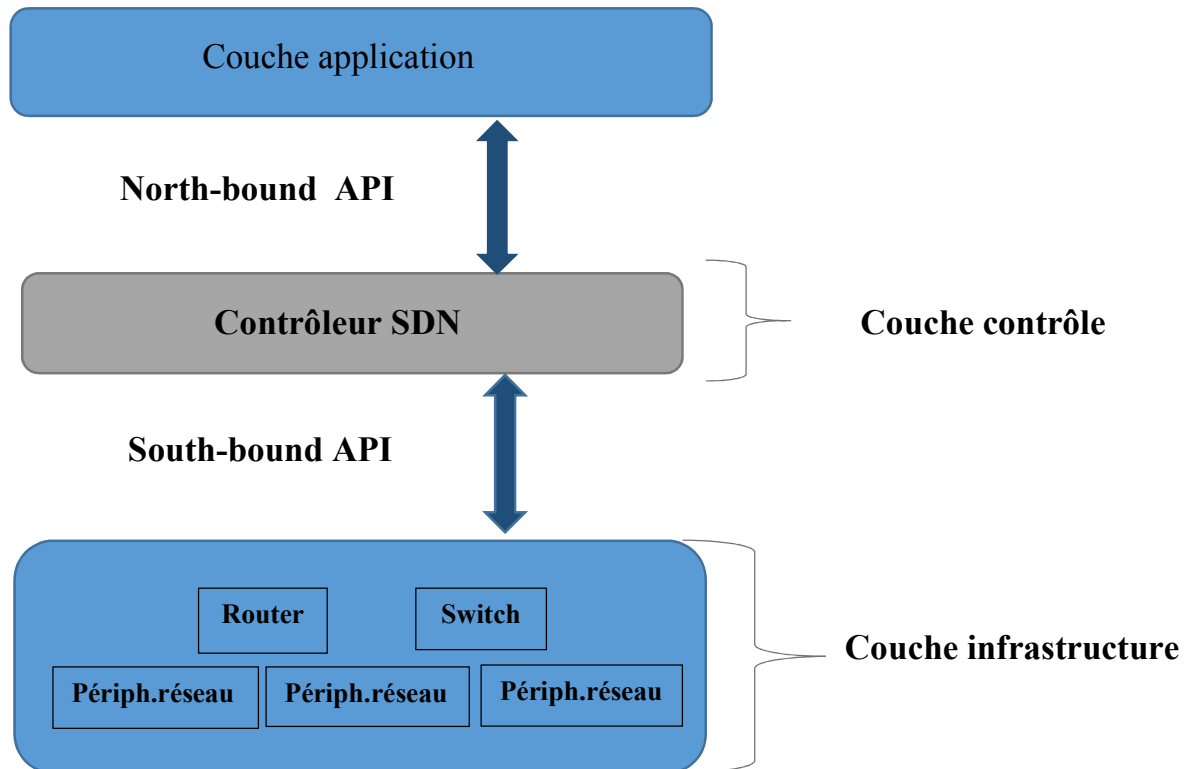


Figure II.1 Architecture SDN

II.2.3. Les différentes couches des réseaux SDN

La couche infrastructure ou couche plan de données : elle est composée principalement des équipements d'acheminement tels que les commutateurs et les routeurs, etc[12]. Son rôle principal est de transmettre les données, surveiller les informations locales et collecter les statistiques.[13]

La couche contrôle : est constituée principalement d'un contrôleur SDN qui permet d'héberger la logique de contrôle du réseau. Ce contrôleur met en œuvre cette logique en accédant au plan des données à travers une interface unifiée appelée '**south-bound**' [12]. Il permet de traduire les requêtes d'une application en une suite d'ordre auprès des équipements du réseau concerné.

La couche application : représente les programmes qui définissent la logique de contrôle du réseau. Ces programmes sont construits moyennant une interface de programmation appelée '**north-bound**' qui est offerte par le contrôleur [12]

II.2.4. Les interfaces de communication :

L'interface north-bound : elle permet la communication entre les composants de niveau supérieur [14]. Elle permet aux applications et aux systèmes d'orchestration de programmer le réseau et d'en demander des services. [15]

L'interface south-bound : dans SDN, les interfaces south-bound sont la spécification de protocole **OpenFlow** qui permet la communication entre les contrôleurs et les commutateurs et les autres nœuds de réseau, ce qui est le cas pour les composants de niveau inférieur. Cela permet en outre au routeur d'identifier la topologie du réseau, de déterminer les flux du réseau et d'implémenter la requête qui lui est envoyée via des interfaces north-bound. [14]

OpenFlow est un protocole de communication entre le contrôleur et le commutateur dans un réseau SDN [McKeown et al. 2008]. Il décrit l'interfaçage entre le plan de données et le plan de contrôle dans un équipement réseau. OpenFlow permet l'exécution des opérations du plan de contrôle dans un contrôleur OpenFlow. Il permet également d'ajuster les règles de routage selon les besoins dynamiques du réseau. Il s'agit d'un nouveau standard dans les réseaux SDN qui donne plus de pouvoir aux administrateurs dans la gestion du réseau. Par exemple, dans un data center, des milliers de serveurs physiques et virtuels sont interconnectés à travers des VLANs, avec SDN et OpenFlow comme protocole, la gestion devient plus simple. Les administrateurs auront seulement à utiliser un logiciel de contrôleur pour programmer les VLAN et suivre l'ensemble du réseau OpenFlow. [16]

II.3. Network Functions Virtualisation (NFV)

II.3.1. Définition de la NFV

La virtualisation des fonctions réseau (NFV) est une initiative fédérée à la base par les plus grands fournisseurs de services de télécommunication du monde avec l'European Telecommunications Standards Institute (ETSI) qui a généré un grand intérêt chez les industriels. L'initiative a été créée pour aborder les principaux défis opérationnels et les coûts de gestion des applications réseau propriétaires et fermées qui sont actuellement déployées.

La NFV est un moyen qui nous permet de dissocier des fonctions réseaux telles que le pare-feu ou le chiffrement de tout matériel dédié en les déplaçant vers des serveurs virtuels. Cela permet de réunir plusieurs fonctions en un seul serveur physique, elle permet de réduire les coûts et le nombre d'interventions sur le terrain et d'accélérer le déploiement de services pour les opérateurs de réseau.

Autrement dit, la NFV est la stratégie de virtualisation des fonctions réseaux, passant de plusieurs équipements matériels propriétaires à des logiciels, exécutés sur des serveurs virtuels en utilisant du matériel standard.

Par exemple, un client qui dispose d'un service réseau entre son siège social et l'une de ses succursales, il utilise un pare-feu, un DNS et un chiffrement pour sa connexion.

Avant la NFV, un appareil séparé pour chaque service aurait été installé sur le site du client.

Avec la NFV, le prestataire de service peut installer un serveur générique dans les locaux du client, puis utiliser une plateforme de virtualisation informatique standard comme openstack, pour exécuter des machines virtuelles pour chaque fonction réseau. [17]

II.3.2. L'architecture NFV

Selon l'ETSI, l'architecture NFV est composée de trois éléments clés :

Infrastructure (NFVI), VNF et NFV MANO.

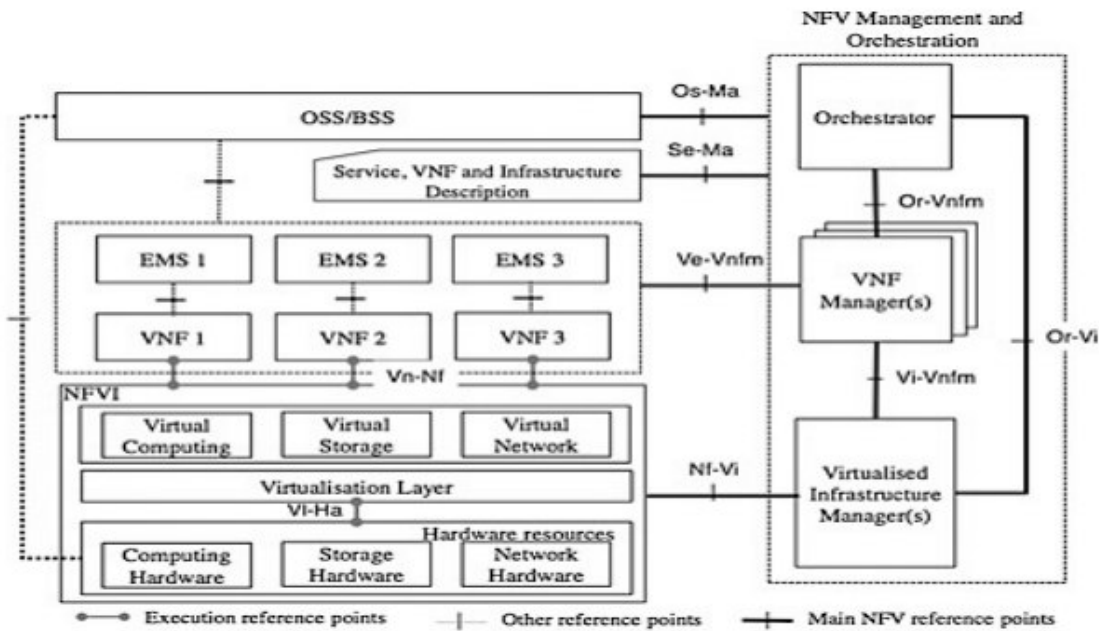


Figure II.2. Architecture NFV [18]

II.3.2.1. L'infrastructure NFV (NFVI) [19]

NFVI (Network Function Virtualisation Infrastructure) fournit les ressources matérielles (serveurs, COTS – Commercial Off The Shelf, cartes électroniques, ...) et le logiciel de virtualisation. Le NFVI se compose donc :

- ❖ D'une interface matérielle (stockage, réseau, calcul)
- ❖ D'une interface virtuelle (stockage, réseau, calcul)
- ❖ D'une couche de virtualisation entre le matériel et le logiciel

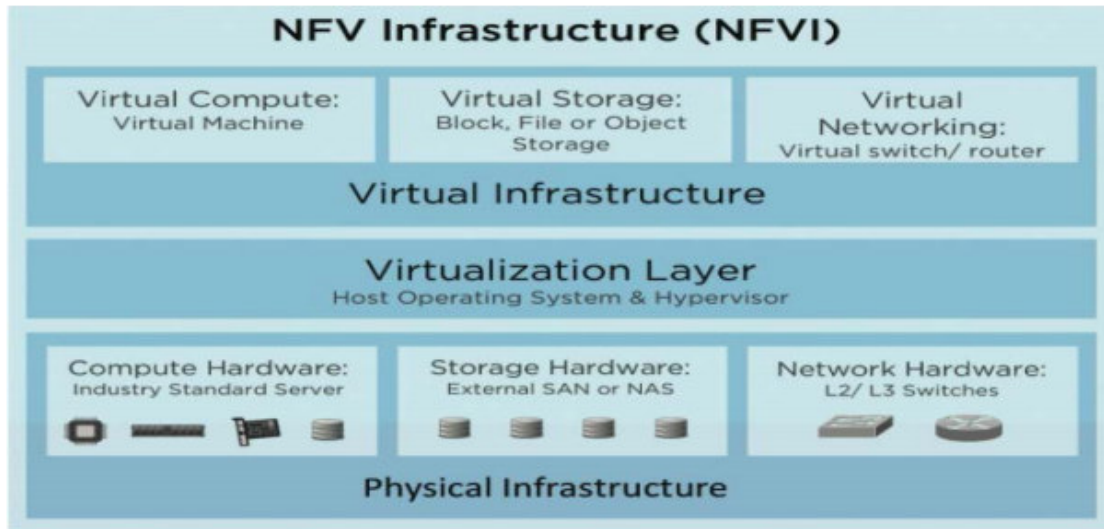


Figure II.3. NFV Infrastructure (NFVI) [20]

II.3.2.2. Les fonctions Réseau Virtuels VNF [19]

Le VNF (Virtualised Network Function) correspond aux fonctions réseaux virtualisées pouvant être exécutées sur les équipements du NFVI.

Les fonctions réseaux virtualisées sont déployées dans des conteneurs au-dessus de la couche de virtualisation. Un conteneur est une instance complète de système d'exploitation, avec son système de fichiers, ses comptes utilisateurs, ses processus, etc. Ainsi, les conteneurs logiciels sont considérés comme des applications pour le serveur.

Enfin, les VNF sont interconnectées entre elles pour constituer un service réseau (NS).

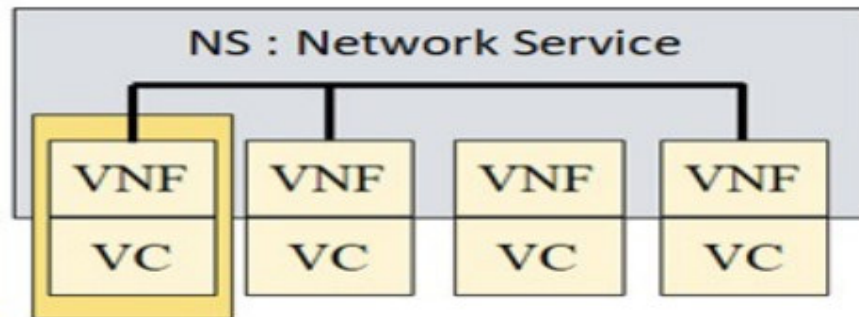


Figure II.4 Chainage de service réseau virtuel

II.3.2.3. NFV MANO : [19]

MANO signifie "Management and Orchestration" et c'est le bloc fonctionnel qui a été défini par ETSI NFV dans le cadre du NFV Architectural Framework.

Le NFV MANO couvre la gestion de l'orchestration et du cycle de vie des ressources physiques et / ou logicielles supportant la virtualisation de l'infrastructure et la gestion du cycle de vie des VNF. NFV Management and Orchestration se concentre sur toutes les tâches de gestion spécifiques à la virtualisation nécessaires dans le cadre NFV. Il est composé de trois blocs de construction :

- **Le NFVO, NFV Orchestrator** : se charge de l'orchestration et de la gestion de l'infrastructure NFV et des ressources logicielles, et de la réalisation de services réseau sur NFVI.
- **Le (s) VNFM (s), VNF Manager** : supervise la gestion du cycle de vie du VNF (par exemple, instanciation, mise à jour, interrogation, mise à l'échelle, terminaison). Ils peuvent être plusieurs VNF Manager en charge d'un ou plusieurs VNF, ou un ensemble de VNF Manager générique qui peut être configuré pour gérer plusieurs VNF, ou un seul VNF Manager générique qui serait configuré pour gérer le cycle de vie de tous les VNF.
- **Le VIM, Virtualized Infrastructure Manager** : contrôle et gère les ressources de calcul, de stockage et de réseau du NFVI, c'est typiquement là que l'on trouve des éléments comme OpenStack.

OSS fournit des tâches de gestion de réseau aux opérateurs. En retour, BSS est orienté métier car il est responsable de la facturation et de la gestion de la relation client (CRM).

II.3.3. Avantages et inconvénients de NFV [21]

II.3.3.1. Avantages de NFV

❖ Flexibilité du matériel :

Parce que NFV utilise du matériel standard, les opérateurs de réseau ont la liberté de choisir et de construire le matériel de la manière la plus efficace pour répondre à leurs besoins et exigences.

Le matériel offert par les fournisseurs de réseaux traditionnels a des options très limitées pour ses capacités de calcul, de mémoire, de stockage et de mise en réseau, et toute modification entraîne une mise à niveau matérielle qui coûte du temps et de l'argent aux opérateurs. Avec NFV, les fournisseurs peuvent désormais choisir entre plusieurs fournisseurs différents et avoir la possibilité de sélectionner les capacités matérielles optimales pour leur architecture et leur planification réseau. Par exemple, si la passerelle Internet utilisée manque de capacité pour stocker la table de routage complète et nécessite une mise à niveau de la mémoire, dans la plupart des implémentations actuelles, elle ne peut y parvenir que via une mise à niveau du contrôleur ou une mise à niveau complète. En NFV, le fournisseur peut allouer plus de mémoire à la VM hébergeant ce VNF.

❖ Cycle de vie du service plus rapide

Les nouveaux services ou fonctionnalités réseau peuvent désormais être déployés plus rapidement, sur demande et en fonction des besoins, ce qui présente des avantages pour les utilisateurs finaux et les fournisseurs de réseau.

Contrairement au matériel physique, les VNF peuvent être créés et supprimés à la volée. Le cycle de vie des VNF peut être beaucoup plus court et dynamique que celui des périphériques physiques, puisque ces fonctions peuvent être ajoutées en cas de besoin, provisionnées facilement à l'aide d'outils logiciels automatisés n'exigeant aucune activité sur site, puis détruites pour libérer des ressources. Comme le besoin est fini. Cela contraste avec l'effort de déploiement nécessaire lorsqu'une nouvelle fonction doit être ajoutée à un réseau existant, ce qui aurait nécessité une installation physique sur site, ce qui peut prendre du temps et être coûteux. La capacité à ajouter rapidement de nouvelles fonctions réseau (agilité de déploiement) est l'un des plus grands avantages de NFV. Les services peuvent maintenant être mis en service ou désaffectés en appuyant sur un bouton sans avoir besoin d'un camion de livraison.

❖ Agilité

La possibilité de déployer, de terminer, de reconfigurer ou de modifier rapidement l'emplacement topologique d'une VNF est communément appelée agilité de déploiement.

❖ Évolutivité et élasticité

De nouveaux services et des applications gourmandes en capacité dans les réseaux actuels maintiennent les opérateurs de réseau, en particulier les fournisseurs de Cloud, à l'affût des demandes croissantes des consommateurs. Les fournisseurs de services ont rattrapé ces exigences, car l'augmentation de la capacité de l'équipement de réseau traditionnel demande du temps, de la planification et de l'argent. Ce problème est résolu par NFV, qui permet des changements de capacité en offrant un moyen d'étendre et de réduire les ressources utilisées par les VNF. Par exemple, si l'un des VNF requiert un processeur, un stockage ou une bande passante supplémentaire, il peut être demandé au VIM et attribué au VNF à partir du pool de matériel. Dans un dispositif réseau traditionnel, il faudrait soit un remplacement complet de l'équipement, soit une mise à niveau matérielle pour modifier l'un de ces paramètres. Mais puisque les VNF ne sont pas limités par les limitations du matériel physique personnalisé, ils peuvent offrir cette élasticité. Par conséquent, les réseaux n'ont pas besoin d'être substantiellement surprovisionnés pour s'adapter aux changements de capacité.

Une autre façon pour le NFV de mettre en œuvre l'élasticité est de décharger la charge de travail d'un VNF et de faire tourner une nouvelle instance pour implémenter la même fonction réseau et diviser la charge avec un VNF existant. Cela n'est pas non plus possible avec un équipement de réseau traditionnel.

L'élasticité est un mot très couramment utilisé dans le contexte NFV pour se référer à la capacité d'un VNF à étendre et étirer les ressources ou à se dérober et à les réduire, en fonction des besoins. En outre, ce terme est utilisé pour faire référence au scénario lorsque nous créons ou supprimons des VNF supplémentaires pour partager la charge de travail d'un VNF existant.

❖ Tirer parti des outils existants

Comme NFV utilise la même infrastructure que les centres de données, il peut réutiliser et exploiter les outils de déploiement et de gestion déjà utilisés dans les centres de données. L'utilisation d'un seul panneau de verre centralisé pour la gestion des serveurs virtuels offre l'avantage d'une adaptation plus rapide pour de nouveaux déploiements sans avoir besoin de développer de nouveaux outils et donc de réduire les coûts de déploiement, de familiarisation et d'utilisation des nouveaux outils.

❖ Développement rapide et indépendance des fournisseurs

Parce que NFV fournit les moyens de déployer facilement une solution de fournisseur différente sans les coûts élevés associés au remplacement du déploiement d'un fournisseur existant, elle empêche les opérateurs de réseau d'être enfermés dans un fournisseur particulier. Les opérateurs peuvent mélanger et assortir les fournisseurs et les fonctions, et choisir entre eux en fonction de la disponibilité des fonctionnalités, du coût de la licence du logiciel, du modèle de support post-déploiement, etc.

De nouvelles solutions et fonctionnalités peuvent être mises en production rapidement, sans attendre que le fournisseur déployé existant les développe et les supporte. Un tel déploiement rapide est en outre facilité par le soutien inhérent de NFV à l'utilisation d'outils et de logiciels open source.

❖ Validation de nouvelles solutions

Les fournisseurs de services préfèrent souvent valider de nouvelles solutions, services et fonctions en les déployant dans des configurations de test, avant de les introduire dans leurs réseaux de production. Traditionnellement, ils ont dû répliquer un sous-ensemble de leur environnement de production pour des tests internes, ce qui a augmenté leur budget opérationnel. Avec NFV, la construction et la gestion de telles configurations de test sont devenues beaucoup plus rentables. Les configurations de test basées sur NFV peuvent être dynamiques et donc mises à l'échelle et modifiées pour répondre aux scénarios de test et de validation.

❖ Efficacité opérationnelle

Avec un matériel commun hébergeant différents VNF, les tâches associées à l'exploitation de l'entreprise, telles que la gestion des stocks, le processus d'approvisionnement, peuvent être centralisées. Cela réduit la charge

opérationnelle par rapport aux déploiements distincts de différents services réseau utilisant plusieurs périphériques matériels.

NFV est essentiellement compatible avec l'automatisation, et peut augmenter les avantages qui peuvent être obtenus grâce à l'utilisation d'outils Machine to Machine (M2M). Par exemple, il est possible pour un outil d'automatisation de surveiller un appareil pour déterminer le besoin de plus de mémoire dans une fonction réseau. Avec NFV, cet outil peut aller de l'avant et demander l'attribution de cette mémoire sans impliquer aucune intervention humaine.

Les activations liées à la maintenance réseau peuvent également bénéficier de manière significative de NFV en réduisant les temps d'arrêt possibles. NFV permet de faire tourner un nouveau VNF, de déplacer temporairement la charge de travail vers ce VNF et de libérer du VNF existant pour les activités de maintenance. Cela permet d'obtenir des réseaux d'autoréparation 24 heures sur 24, 7 jours sur 7 et en service, et de minimiser la perte opérationnelle de revenus due aux pannes de réseau.

II.3.3.2. Inconvénients de NFV

❖ Problèmes de sécurité

NFV crée plusieurs nouveaux défis en matière de sécurité. Selon un document technique d'Alcatel Lucent, il existe quatre problèmes de sécurité majeurs spécifiques à NFV dont les opérateurs doivent être conscients :

- **L'introduction de nouveaux composants logiciels qui n'existaient pas dans le modèle de réseau traditionnel :** l'hyperviseur et divers éléments de gestion / orchestration, ce qui crée une «chaîne de confiance plus longue».
- **Isolation réduite :** dans la NFV, presque tous les éléments du réseau sont capables de communiquer directement les uns avec les autres, au moins au niveau physique (par opposition aux réseaux traditionnels dans lesquels différents segments de réseau sont souvent séparés physiquement et incapables de communiquer).
- **Partage des risques entre plusieurs composants non liés en raison de la mise en pool des ressources :** une attaque sur une fonction de réseau virtuel (VNF) donnée peut affecter d'autres VNF s'exécutant sur la même machine virtuelle ou le même serveur physique.

- **Le problème de "l'engagement de clé"** : comment partager efficacement les clés et les identifiants de sécurité entre les fonctions du réseau hébergé de manière à empêcher l'accès par des attaquants.

❖ Problèmes de maturité et de complexité dans OpenStack

Une majorité des implémentations NFV repose en partie ou en totalité sur OpenStack, qui fournit un gestionnaire d'infrastructure virtuelle (VIM) et sur certaines couches MANO au-dessus de l'hyperviseur et des ressources virtualisées. Mais OpenStack, la norme de facto actuelle pour la création de Cloud privés, est également difficile à apprendre, à déployer et à utiliser.

II.4. Différences entre SDN et NFV [22]

NFV et SDN sont deux technologies étroitement liées qui existent souvent ensemble, mais pas toujours. NFV et SDN sont deux mouvements vers la virtualisation de réseau et l'automatisation, mais les deux technologies ont des objectifs différents. SDN et NFV s'appuient tous deux sur la couche logicielle pour administrer virtuellement le réseau. Mais chaque solution intervient à des niveaux différents. Ils diffèrent dans leur manière de séparer les ressources et les fonctions virtuelles.

Catégorie	SDN	NFV
Idee de base	Séparation du contrôle et des données, centralisation du contrôle et programmabilité du réseau	transfère les fonctions réseau des appliances dédiées aux serveurs génériques.
Zone d'opération	campus, centre de données et / ou de cloud	Réseau de fournisseurs de services
Appareils cibles	Serveurs de base et commutateurs	Serveurs de base et commutateurs
Cible initiale de l'application	Orchestration du Cloud et mise en réseau	Routeurs, firewalls, passerelles, CDN, accélérateurs WAN, assurance SLA
Protocoles	OpenFlow	Aucun
Organisation de soutien	Open Networking Foundation (ONF)	Groupe de travail ETSI NFV

II.5. Openstack

II.5.1. Définition d'Openstack

Openstack est un logiciel open source qui permet la construction de Cloud privé et public de type Infrastructure-as-a-Service (IaaS). Il contient de vastes pools de ressources de calcul, de stockage et de réseaux dans un centre de données, tous gérés via un Dashboard permettant aux administrateurs de contrôler leurs ressources via une interface web.[23]

Il s'installe sur un système d'exploitation libre comme Ubuntu ou Debian et se configure entièrement en ligne de commande.

Openstack joue le rôle d'une couche de management de Cloud qui assure la communication entre la couche physique où se trouvent des serveurs physiques occupés par des hyperviseurs différents (Vmware ESX, Citrix Xen, KVM, qemu...) et la couche applicative (Applications, utilisateurs, administrateurs...)[9]

II.5.2. Architecture d'openstack

Openstack est composé d'une série de logiciels et de projets au code source libre qui communiquent via des APIs et qui sont maintenus par la communauté.

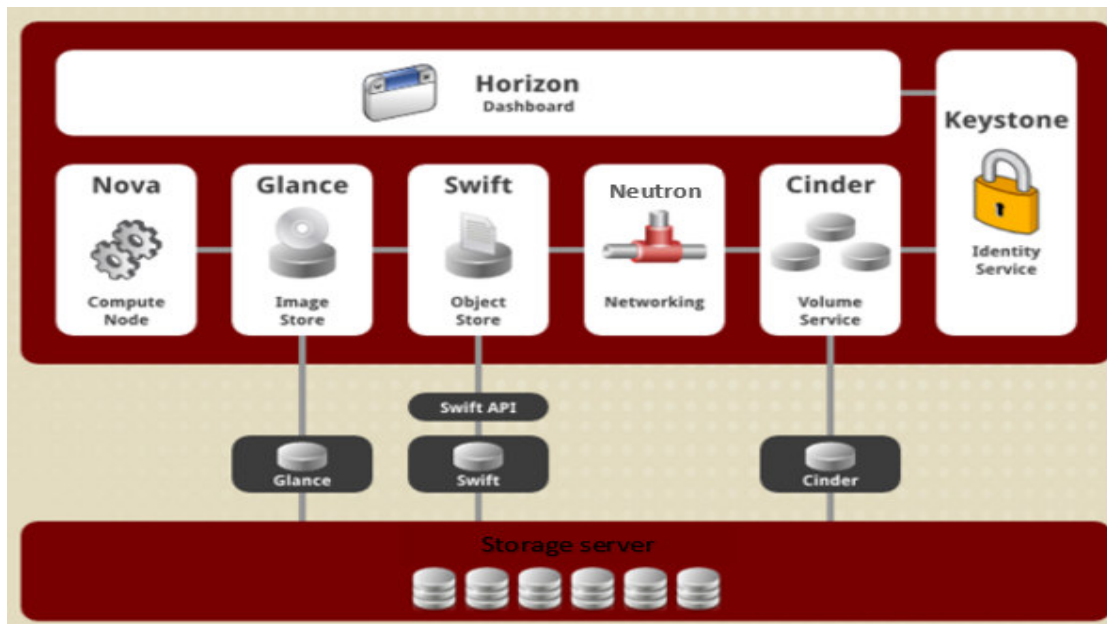


Figure II.5. Architecture Openstack [24]

II.5.2.1. Les services d'openstack

L'architecture s'articule autour des composants suivants :

II.5.2.1.1 Openstack Nova (OpenStack Compute)

NOVA est le cœur d'Openstack et est par conséquent l'un des composants le plus complexes. Il permet la gestion de larges réseaux de machines virtuelles et d'une architecture redondante et évolutive. Elle fournit une interface d'administration et l'API nécessaire à l'orchestration du Cloud. Elle inclue : les gestions des instances serveurs la gestion du réseau et les contrôle d'accès.[25]

II.5.2.1.2 Openstack Glance (Imaging Service)

Glance est le module gérant le catalogue d'image, il fournit les services de stockages, de découvertes, d'enregistrements et de distributions pour les images disques de machines virtuelles. Il fournit également une API compatible REST permettant d'effectuer des requêtes pour obtenir des informations sur les images hébergées par les différents magasins de stockages.[25]

II.5.2.1.3. Openstack Horizon (Openstack Dashboard)

Horizon est le tableau de bord web d'Openstack. Il fournit une interface web aux utilisateurs et aux administrateurs qui permet d'agir sur les différents services d'Openstack, y compris Nova, Swift, Keystone etc.

Avec cette interface Web nous pouvons créer des machines virtuelles, assigner des adresses IP ou gérer le contrôle d'accès.[25]

II.5.2.1.4. Openstack Keystone (Openstack Identity)

C'est le principal service d'authentification et d'autorisation qui gère les utilisateurs, les services et les points de terminaison (endpoints). Keystone utilise des jetons d'authentification (tokens) pour autoriser l'accès aux ressources et maintient l'état des sessions.[11]

II.5.2.1.5. Openstack Cinder (Openstack Block Storage)[9]

Cinder est le service de stockage en mode block, Il gère les opérations de création, d'attachement et de détachement d'un volume sur une VM.

Ce service était inclus dans Nova à l'origine (sous le nom nova-volume) dans les versions précédente de OpenStack.

II.5.2.1.6. OpenStack Neutron (OpenStack Network)

À l'origine ce composant s'appelait Quantum, il permet aux utilisateurs de créer des réseaux à la demande et d'y attacher des machines virtuelles.[25]

OpenStack Network fournit des fonctionnalités réseaux avancées tels que : tunneling, QoS, Réseaux virtuels et équilibrage de charge, etc.[9]

II.5.1.2.7. OpenStack Swift (Object Storage)

Object Storage sert à la création d'espace de stockage redondant et évolutif pour le stockage de plusieurs pétaoctets de données. Il ne s'agit pas réellement d'un système de fichier mais est surtout conçu pour le stockage à long terme de gros volumes.[26]

II.5.2.2. Autres Services d'Openstack

II.5.2.2.1. OpenStack Ceilometer (OpenStack Telemetry)

Il permet de collecter différentes métriques sur l'utilisation du cloud. Par exemple il permet de récolter le nombre d'instances lancées dans un projet et depuis combien de temps.[27]

II.5.2.2.2 OpenStack Mistral [28]

Mistral est le service de workflow OpenStack. Ce projet vise à fournir un mécanisme permettant de définir des tâches et des flux de travail sans écrire de code, les gérer et les exécuter dans l'environnement cloud.

II.5.2.2.3 OpenStack barbican[29]

Barbican est une API REST conçue pour le stockage sécurisé, l'approvisionnement et la gestion de secrets tels que les mots de passe et les clés de chiffrement. Il est destiné à être utile pour tous les environnements, y compris les grands clouds éphémères.

II.5.2.3. Les Services d'orchestration d'Openstack

II.5.2.3.1. OpenStack Heat (OpenStack Orchestration)

Heat est le composant d'orchestration d'OpenStack. Il permet par exemple de demander à Nova de démarrer une machine virtuelle supplémentaire en cas de charge importante de façon automatique.[9]

II.5.2.3.2. OpenStack Tacker «OpenStack NFV Orchestration»

Tacker est un projet OpenStack officiel qui crée un gestionnaire VNF générique (VNFM) et un orchestrateur NFV (NFVO) pour déployer et exploiter les services réseau et les fonctions de réseau virtuel (VNF) sur une plate-forme d'infrastructure NFV comme OpenStack. Il est basé sur le Framework architectural ETSI MANO et fournit une pile fonctionnelle pour orchestrer les services réseau de bout en bout à l'aide de VNF. [30]

II.6. OPNFV

OPNFV (Open Platform for Network Function Virtualization) est un projet lancé en septembre 2014 par la Linux Foundation. Elle facilite le développement et l'évolution des composantes NFV à travers différents écosystèmes open source. Grâce à l'intégration au niveau du système, le déploiement et les tests, OPNFV crée une plate-forme NFV de référence pour accélérer la transformation des réseaux de fournisseurs d'entreprise et de services. En tant que projet open source, OPNFV est particulièrement bien placée pour réunir le travail des organismes de normalisation, les communautés open source et les fournisseurs commerciaux pour offrir une plate-forme de facto de NFV open source standard pour l'industrie.[31]

II.7. Conclusion

Dans ce chapitre, nous avons abordé les réseaux définis par les logiciels SDN et la virtualisation des fonctions réseau NFV qui sont de nouvelles façons de concevoir, construire et exploiter les réseaux tout en détaillant leurs architectures, ainsi que la solution OpenStack et les éléments qui la composent.

Dans le prochain chapitre, nous allons mettre l'accent sur NFV-MANO et le projet Tacker d'Openstack.

Chapitre III

NFV-MANO et Tacker

III.1. Introduction :

Comme nous l'avons mentionné dans les avantages de NFV, nous pouvons facilement augmenter et réduire les VNF, pour cela il doit y avoir une entité qui lance et gère ces VNF. À cette fin, le groupe de spécification de l'industrie de l'Institut européen des normes de télécommunications (ETSI ISG NFV) a défini un cadre pour la gestion et l'orchestration de la NFV « NFV-MANO ».

Dans ce chapitre, nous allons présenter NFV-MANO, son architecture et ses différents blocs fonctionnels, par la suite nous allons mettre un point sur le projet Tacker d'OpenStack, ses composants et ses différents cas d'utilisation.

III.2. NFV MANO Management and Orchestration

III.2.1. Définition du NFV MANO

La norme ETSI NFV MANO décrit un framework pour mettre en service, gérer et orchestrer les fonctions réseau virtualisées, y compris les définitions des opérations nécessaires pour gérer leur fonctionnement, leur cycle de vie, leur configuration et l'infrastructure réseau sur laquelle ils s'exécutent. L'objectif principal de cette norme est de répondre au besoin d'un framework partagé pour définir une architecture NFV, avec des interfaces et des concepts bien définis, capables d'inter fonctionner avec les systèmes de gestion et les infrastructures existantes.

III.2.2. Les blocs fonctionnels du cadre architectural NFV-MANO

NFV-MANO définit plusieurs blocs fonctionnels, chacun avec un ensemble de responsabilités. Chacun de ceux-ci applique des opérations de gestion et d'orchestration sur des entités bien définies, en tirant parti des services offerts par les autres blocs fonctionnels.

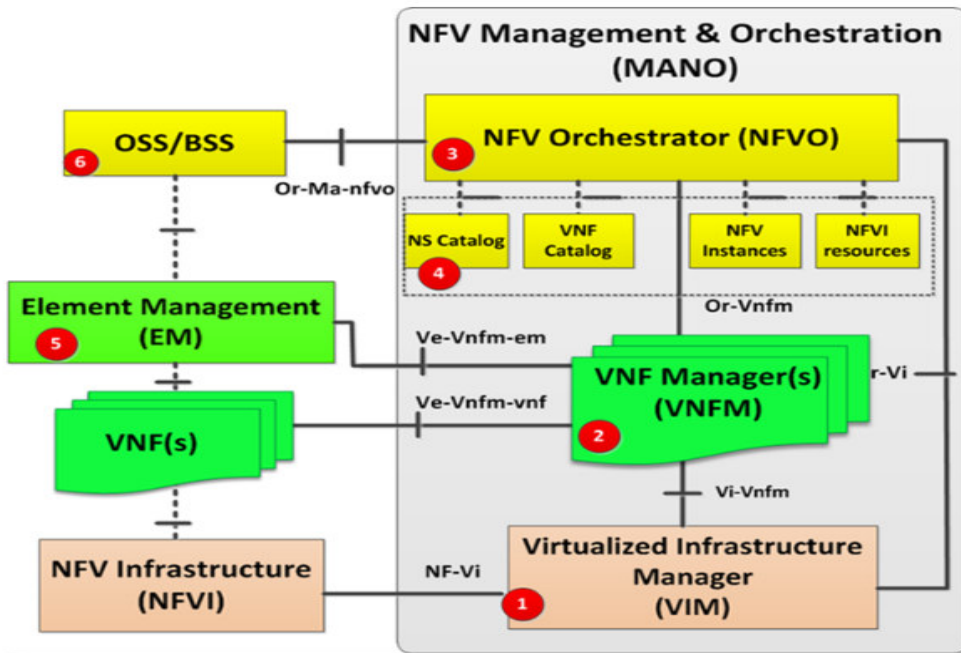


Figure III.1. Le cadre architectural NFV-MANO [32]

Le cadre architectural NFV-MANO comprend trois blocs fonctionnels de base :

- Virtualized Infrastructure Manager (VIM)
- VNF Manager (VNFM)
- NFV Orchestrator (NFVO)
- Un groupe de référentiels (bloc 4)

En plus des quatre blocs à l'intérieur du MANO, il y a deux blocs à l'extérieur, à savoir la gestion traditionnelle des éléments (EM) et l'OSS / BSS. Alors que les deux derniers blocs ne font pas directement partie du MANO, ils échangent des informations avec MANO, de sorte que l'apprenant doit les positionner correctement contre les blocs MANO .

Ci-dessous, nous avons décrit une description de chacun de ces six blocs en commençant par Virtualized Infrastructure Manager (VIM).

III.2.2.1. Virtualized Infrastructure Manager (VIM) [33]

Le gestionnaire d'infrastructure virtualisé (VIM) est la fonction NFV-MANO responsable du contrôle et de la gestion des ressources contenues dans une infrastructure NFV, y compris les capacités de calcul, de stockage et de réseau fournies dans l'ensemble de ses points de présence NFVI

Un VIM peut se spécialiser dans la gestion d'un certain type de ressource NFVI (par exemple, calcul uniquement, stockage uniquement ou mise en réseau uniquement) ou capable de gérer plusieurs types de ressources NFVI simultanément, exposant une interface en direction nord aux autres fonctions pour les gérer. Les tâches typiques livrées par le VIM sont mentionnées ci-dessous :

- Gère le cycle de vie des ressources virtuelles dans un domaine NFVI. En d'autres termes, il crée, gère et arrête des machines virtuelles (VM) à partir de ressources physiques dans un domaine NFVI.
- Conserve l'inventaire des machines virtuelles (VM) associées aux ressources physiques.
- Gestion des performances et des défaillances du matériel, des logiciels et des ressources virtuelles.
- Conserve les API liées au nord et expose ainsi les ressources physiques et virtuelles à d'autres systèmes de gestion.

III.2.2.2. Gestionnaire de fonction de réseau virtuel (VNFM) [33]

Un gestionnaire (VNFM) se charge du cycle de vie des instances VNFs en fonction des données contenues dans le descripteur. Il démarre les instances VNF, les gère, les adapte et récupère des indicateurs de supervision. Le gestionnaire VNFM utilise donc le descripteur VNFD durant la procédure d'instanciation de la fonction réseau virtualisée VNF et pendant toute la durée de vie de la VNF ;

Une seule instance VNF est associée de manière unique à un gestionnaire VNF donné; Ce gestionnaire peut gérer plusieurs autres instances, de types identiques ou différents. Alors qu'un VNFM doit prendre en charge les exigences des VNF qui lui sont associés, la plupart des fonctions du gestionnaire VNF sont génériques et ne dépendent d'aucun type particulier de VNF.

Plus précisément, VNFM effectue les opérations suivantes:

- VNFM gère le cycle de vie des VNF. Cela signifie qu'il crée, maintient et termine les instances VNF. (Qui sont installés sur les machines virtuelles (VM) que le VIM crée et gère)
- Il est responsable de la gestion des défauts, de la configuration, de la comptabilité, des performances et de la sécurité des VNFs.
- Il augmente ou réduit les VNF, ce qui se traduit par une augmentation et une réduction de l'utilisation du processeur.

Il peut y avoir plusieurs VNFM gérant des VNF distincts ou un seul VNFM gérant plusieurs VNF.

III.2.2.3. Orchestrateur NFV (NFVO)[33]

L'entité d'orchestration est responsable du cycle de vie des services réseau tant au niveau logiciel que matériel sur plusieurs domaines en contrôlant les VIM de chaque domaine ; le NFVO permet de contrôler l'intégration de nouveaux services réseau et de nouveaux VNF dans une infrastructure virtuelle ainsi qu'il valide et autorise les demandes de ressources NFVI.

Les orchestrateurs NFV se composent de deux couches :

- L'orchestration des ressources
- L'orchestration de service

L'orchestration des ressources [34]

NFVO coordonne, autorise, libère et engage les ressources NFVI telles que les ressources de calcul, de stockage et de réseau entre différents PoPs ou dans un PoP. Cela se fait en s'interagissant directement avec les VIM via leurs APIs liées au nord au lieu de s'interagir directement avec les ressources NFVI.

Certaines des fonctionnalités fournies par cet aspect sont les suivantes :

- Validation et autorisation des demandes de ressources NFVI provenant de VNF Manager, pour contrôler la manière dont l'allocation des ressources demandées interagit au sein d'un même NFVI-PoP ou entre plusieurs points NFVI.
- Gestion des ressources NFVI, y compris la distribution, la réservation et attribution de ressources NFVI aux instances NS et VNF ; ce sont soit extraites d'un référentiel de ressources NFVI déjà connues, ou demandé à partir d'un VIMs si nécessaire. Le NFVO résout également le lieu des VIM, en les fournissant aux VNFM si nécessaire.

- Gestion de la relation entre une instance de VNF et les ressources NFVI qui lui sont allouées, en utilisant les référentiels des ressources NFVI et les informations reçues des VIM.
- Collecte d'informations concernant l'utilisation par simple ou multiple Instances VNF de ressources NFVI.

L'orchestration de service [34]

L'orchestration de service permet de gérer et de coordonner la création d'un service de bout en bout en impliquant des VNF provenant de différents domaines VNFM

Voici ci-dessous quelques fonctionnalités de cet aspect :

- Service Orchestration crée un service de bout en bout entre différents VNF. Cela se fait en se coordonnant avec les différents VNFM afin de ne pas avoir à parler directement avec les VNF. Un exemple serait la création d'un service entre les VNF de la station de base d'un fournisseur et les nœuds VNF du nœud principal d'un autre fournisseur.
- Service Orchestration peut instancier des VNFM, le cas échéant.
- Il gère la topologie des instances de services réseau (également appelées graphiques de transfert VNF).

III.2.2.4. Groupe de référentiels [33]

Ce type de référentiel contient de différentes informations sur NFV-MANO, il est utilisé pour stocker des descripteurs et des informations sur les instances VNF et NS. Il existe quatre types de référentiels :

❖ VNF Catalog :

Le catalogue VNF représente le référentiel de tous les packages VNF intégrés, prenant en charge la création et la gestion du package VNF (VNF Descriptor (VNFD), images logicielles, fichiers de manifeste, etc.) via des opérations d'interface exposées par le NFVO. Le NFVO et le VNFM peuvent interroger le catalogue VNF pour trouver et récupérer un VNFD, pour prendre en charge différentes opérations (par exemple, validation, vérification de la faisabilité de l'instanciation).

❖ Network Services (NS) Catalog:

Le catalogue NS représente le référentiel de tous les services réseau intégrés, supportant la création et la gestion des modèles de déploiement NS (descripteur de service réseau (NSD), descripteur de lien virtuel (VLD) et VNF Forwarding Graph Descriptor (VNFFGD) via les opérations d'interface exposées par le NFVO

❖ Référentiel d'instances NFV :

Le référentiel d'instances NFV contient des informations sur toutes les instances VNF et les instances de service réseau.

Chaque instance VNF est représentée par un enregistrement VNF et chaque instance NS est représentée par un enregistrement NS. Ces disques sont mis à jour pendant le cycle de vie des instances respectives, reflétant les modifications résultant de l'exécution des opérations de gestion du cycle de vie NS et / ou des opérations de gestion du cycle de vie VNF. Cela prend en charge les responsabilités de NFVO et de VNFM dans le maintien de l'intégrité et de la visibilité des instances NS, respectivement des instances VNF, et de la relation entre elles.

❖ Référentiel de ressources NFVI :

Ce type de référentiel est utilisé dans le but d'établir des services NFV, il contient des informations sur les ressources NFVI disponibles réservées ou allouées telles qu'elles ont été extraites par le VIM à travers les domaines d'infrastructure de l'opérateur, prenant ainsi en charge les informations utiles pour la réservation de ressources, fins d'allocation et de surveillance. A ce titre, le référentiel de ressources NFVI joue un rôle important en soutenant la gouvernance et l'orchestration de NFVO, en permettant aux ressources réservées ou allouées par NFVI d'être suivies par rapport aux instances NS et VNF qui leurs sont associées (par exemple, le nombre de machines virtuelles utilisées par une certaine instance VNF à tout moment au cours de son cycle de vie).

I.2.2.5. Autres blocs fonctionnels [33]

Les deux systèmes de gestion suivants ne font pas partie de NFV MANO mais sont décrits car ils échangent des informations avec les blocs fonctionnels NFV-MANO.

❖ **Element Management (EM):**

La gestion des éléments est un autre bloc fonctionnel défini dans la structure ETSI et est destiné à faciliter la mise en œuvre des fonctions de gestion FCAPS¹ d'un ou de plusieurs VNF. Ceci comprend :

- Configuration pour les fonctions réseau fournies par le VNF.
- Gestion des pannes pour les fonctions réseau fournies par le VNF.
- Comptabilisation de l'utilisation des fonctions VNF.
- Collecter des résultats de mesure de performance pour les fonctions fournies par le VNF.
- Gestion de la sécurité pour les fonctions VNF.

L'EM peut être au courant de la virtualisation et de la collaboration avec le gestionnaire VNF pour effectuer ces fonctions qui nécessitent des échanges d'informations concernant les ressources NFVI associées au VNF.

❖ **OSS / BSS:**

Les OSS / BSS sont les combinaison des autres opérations de l'opérateur et des fonctions de support métier qui ne sont pas autrement explicitement capturé dans le cadre architectural actuel, mais devrait avoir des échanges d'informations avec des blocs fonctionnels dans le cadre architectural NFV-MANO. Les fonctions OSS / BSS peuvent assurer la gestion et orchestration des systèmes existants et peut avoir une visibilité de bout en bout des services fournis par les fonctions réseaux existantes dans le réseau d'un opérateur.

❖ **Points de référence NFV [34] :**

La structure ETSI définit des points de référence pour identifier la communication qui doit avoir lieu entre les blocs fonctionnels. Il est important de les identifier et de les définir pour garantir la cohérence du flux d'informations au sein de l'implémentation du fournisseur pour les blocs fonctionnels. Cela permet également d'établir un moyen ouvert et commun d'échanger des informations entre les blocs fonctionnels.

La liste suivante décrit ces points de référence plus en détail.

¹ **FCAPS** : est un mode réseau de gestion des télécommunications ISO et est l'abréviation des cinq principaux paramètres de gestion : défaut, configuration, performance, comptabilité et sécurité.

- **Os-Ma-nfvo:** Il s'agissait à l'origine du nom d'Os-Ma et vise à définir la communication entre OSS / BSS et NFVO. C'est le seul point de référence entre l'OSS / BSS et le bloc de gestion de NFV (MANO).
- **Ve-Vnfm-vnf:** Ce point de référence définit la communication entre VNFM et VNF. Il est utilisé par VNFM pour la gestion du cycle de vie VNF et pour échanger des informations de configuration et d'état avec le VNF.
- **Ve-Vnfm-em:** à l'origine défini avec Vn-Vnfm-vnf (conjointement appelé Ve-Vnfm), il est maintenant défini séparément pour la communication entre les blocs fonctionnels VNFM et EM. Il prend en charge la gestion du cycle de vie VNF, la gestion des pannes et de la configuration et d'autres fonctions, et il n'est utilisé que si le gestionnaire d'exploitation connaît la virtualisation.
- **Nf-Vi:** Ce point de référence définit l'échange d'informations entre VIM et les blocs fonctionnels dans NFVI. VIM l'utilise pour allouer, gérer et contrôler les ressources NFVI.
- **Or-Vnfm:** la communication entre NFVO et VNFM se produit via ce point de référence, tel que l'instanciation VNF et d'autres flux d'informations liés au cycle de vie VNF.
- **Or-Vi:** l'orchestrateur NFV (NFVO) est conçu pour communiquer directement avec VIM afin d'influencer la gestion des ressources d'infrastructure, telles que la réservation de ressources pour les machines virtuelles ou l'ajout de logiciels VNF.
- **Vi-Vnfm:** ce point de référence est destiné à définir les normes d'échange d'informations entre VIM et VNFM, telles que la demande de mise à jour de ressources pour une VM exécutant une VNF.
- **Vn-Nf:** C'est le seul point de référence qui ne possède pas de bloc fonctionnel de gestion. Ce point de référence est destiné à communiquer les besoins de performance et de portabilité du VNF au bloc d'infrastructure.

Le tableau cité dans l'annexe B résume ces définitions de points de référence

III.3. Tacker

III.3.1. Historique de Tacker

OpenStack Tacker existe depuis quelques années maintenant. Initialement appelé ServiceVM, il a été rebaptisé Tacker et promu au OpenStack Summit de

Vancouver en 2015. Initialement défini par une poignée de personnes, il était assez silencieux jusqu'à ce qu'OPNFV et d'autres projets commencent à

considérer ce code comme un outil pour exercer l'infrastructure pour d'autres projets sur lesquels ils travaillaient, à savoir SFC (Service Function Chaining) dans OPNFV, et mapper à ETSI NFV les fonctions VNFM.

Tacker est géré sous le parapluie OpenStack, donc il suit les directives de projet de la communauté OpenStack et le modèle de gouvernance. Pas à pas Tacker est passé d'un projet indépendant dans OpenStack à la grande tente et fait maintenant partie de la version Mitaka.

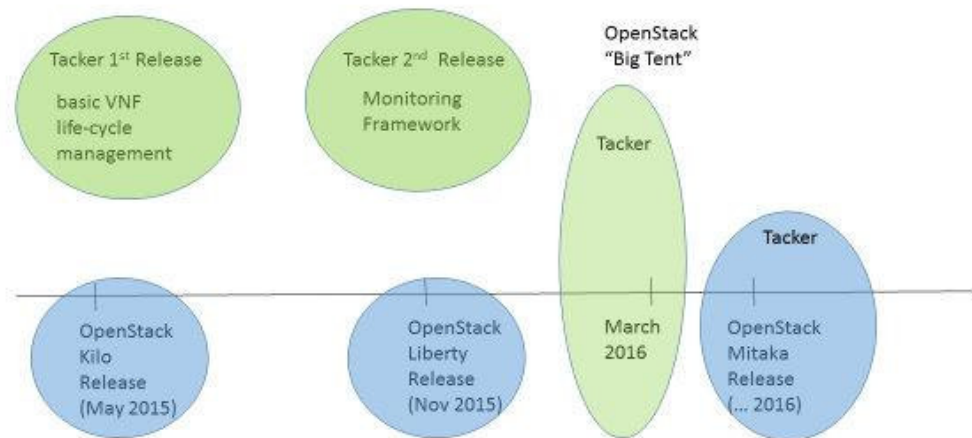


Figure III.2. Tacker project evolution in OpenStack

III.3.2. Définition de Tacker [35]

Tacker est un projet openstack visant à créer un générique VNF Manager (VNFM) et un NFV Orchestrator (NFVO) pour déployer et exploiter des fonctions réseaux virtuelles(VNF) et services réseau (NS) sur une plate-forme d'infrastructure NFV comme OpenStack. Il traite des cas d'utilisation de l'orchestration NFV et du gestionnaire VNF en utilisant l'architecture ETSI MANO Architectural Framework et fournit une pile fonctionnelle pour orchestrer des services réseaux de bout en bout en utilisant des VNFs.

III.3.3. Tacker Architecture [35]

La figure 3 présente l'architecture Tacker, qui fournit une API spécifique pour la gestion du cycle de vie des VNF ainsi que des fonctionnalités telles que la surveillance VNF et la mise à l'échelle automatique.

Le catalogue NFV stocke :

- Le Descripteur VNF (VNFD)
- Le Descripteur de service réseau (NSD)
- Le Descripteur Virtual Network Function Forwarding Graph (VNFFGD).

Tacker utilise Openstack comme Virtualized Infrastructure Manager (VIM), en particulier les services Nova, Neutron, Heat et Keystone.

Il utilise aussi la Spécification de Topologie et d'Orchestration pour le Cloud Applications(TOSCA) afin de définir le descripteur VNF (VNFD) et le descripteur de service réseau (NSD).

Pour fournir un moyen visuel pour gérer les VNFs, le projet Tacker modifie le service Horizon d'Openstack de sorte à ce que les interactions avec les VNFs soient effectuées facilement via le tableau de bord.

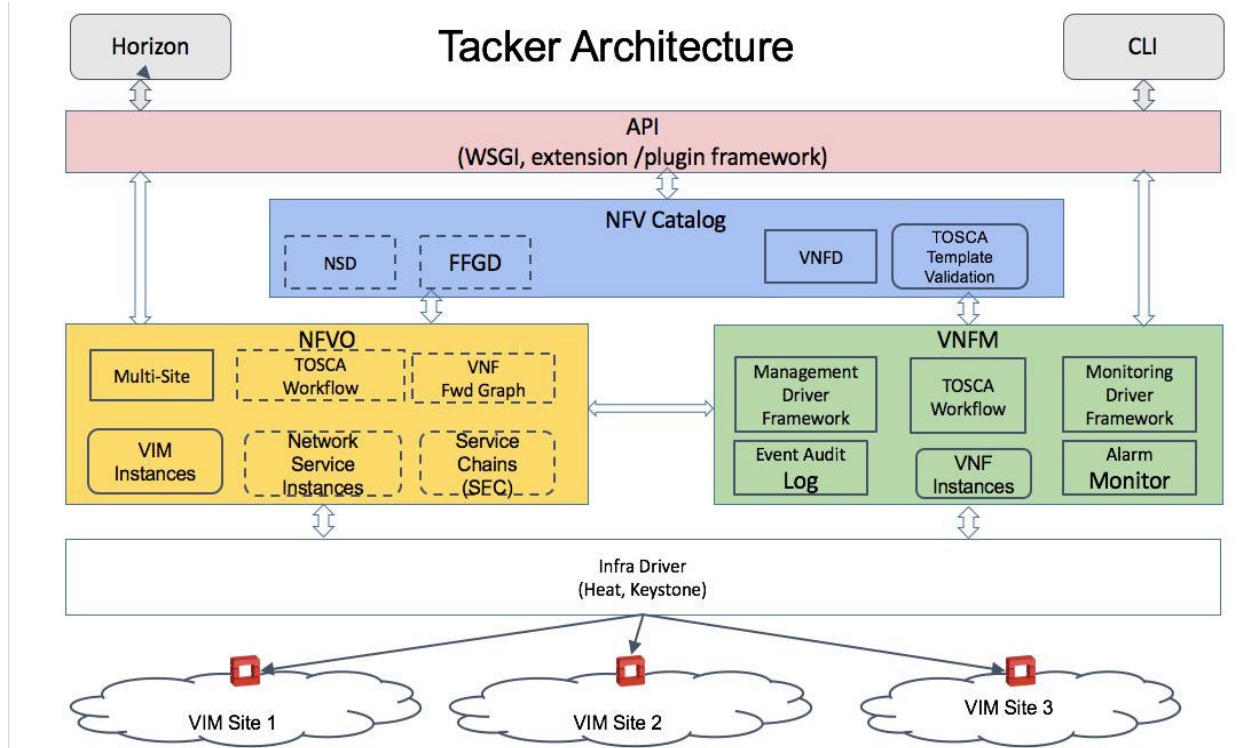


Figure III.3. Tacker Architecture

III.3.3.1. Les composants de Tacker[35]

Tacker est constitué de 3 composants :

1) Network Function Virtualization Catalog

Le catalogue NFV est la collection de templates qui peuvent être utilisés pour déployer divers services réseau. Le format de modèle standard est TOSCA² (Topology and Orchestration Specification for Cloud Applications).

Les templates qui peuvent être intégrés dans Tacker sont :

²

TOSCA est une norme ouverte OASIS qui définit la description interoperable des services et applications hébergés sur le cloud et ailleurs; y compris leurs composants, relations, dépendances, exigences et capacités, permettant ainsi la portabilité et la gestion automatisée à travers les fournisseurs de cloud sans tenir compte de la plate-forme ou de l'infrastructure sous-jacente; élargissant ainsi le choix des clients, améliorant la fiabilité et réduisant les coûts et les délais de mise en valeur.

i) VNF Descriptors (VNFD) :

C'est un référentiel de descripteur de fonction de réseau virtuel (VNFD) qui à son tour est un fichier template qui spécifie le déploiement et les exigences du comportement d'un VNF. Il permet la prise en charge de plusieurs machines virtuelles par VNF. Le VNFD est stocké dans Tacker DB.

ii) Network Services Descriptors (NSD) :

NSD (Network Services Descriptor) est une template utilisée pour composer dynamiquement un service réseau complet. En utilisant un seul NSD, plusieurs VNF peuvent être créés.

Autrement dit, le NSD contient les attributs pour un groupe de fonctions réseaux qui constituent ensemble un service réseau. Ces attributs contiennent aussi les exigences pour chaîner les VNF ensemble et fournir le service réseau.

iii) VNF Forwarding Graph Descriptors :

C'est une template intégrée dans Tacker. La template VNFFG décrit la stratégie de gestion du trafic réseau via les VNFs déployés. Elle contient des metadata concernant le graphe d'acheminement des fonctions réseaux virtualisées VNF, ainsi que les références aux conteneurs de l'infrastructure, aux instances VNFs et au lien entre les VNFs. Les métadonnées contiennent en plus, des règles de politiques pour les fonctions d'acheminement (règle de routage et de commutation). Autrement dit, le VNF -FG montre le graphique des liens logiques reliant les nœuds VNF dans le but de décrire le flux de trafic entre ces VNFs.

Le VNFFG correspond à la fonction SFC (Service Function Chaining) pour le NFV.

2) VNF Manager :

Un gestionnaire (VNFM) s'en charge du cycle de vie des instances VNFs en fonction des données contenues dans le descripteur. Il démarre les instances VNF, les gère, les adapte et récupère des indicateurs de supervision. Le gestionnaire VNFM utilise donc le descripteur VNFD durant la procédure d'instanciation de la fonction réseau virtualisée VNF et pendant toute la durée de vie de la VNF.

3) NFV Orchestration :

NFVO aide à fournir un placement efficace de VNF. NFVO permet de connecter des VNF individuels en utilisant le chainage de fonction de service à l'aide d'un descripteur de graphe VNF. Dans un modèle de changement de fonction de service, les VNF décomposées sont réunies pour fournir une solution de réseau de bout en bout

III.3.4. Cas d'utilisation de Tacker

- **VCE (Virtual Customer Edge) :**

L'API Tacker peut être utilisée par l'OSS / BSS de SP (service provider) ou un orchestrateur NFV pour déployer des VNF dans le réseau de SP afin de fournir des services de réseau souples aux réseaux distants des clients.

- **VCPE (Virtual Customer Premises Equipement) :**

API Tacker peut être utilisé par l'OSS / BSS de SP ou un orchestrateur NFV pour gérer des équipements CPE distants compatibles OpenStack afin de déployer des VNF pour fournir des services réseau locaux sur le site du client.

- **VPE (virtual provider edge) :**

L'API Tacker peut être utilisée par l'OSS / BSS de SP ou un orchestrateur NFV pour déployer des VNF dans le réseau de SP afin de virtualiser les services réseau existants dans une fonction virtuelle.

III.4. Conclusion :

Dans ce chapitre, nous avons abordé NFV-MANO et le projet tacker d'Openstack qui se base sur l'architecture ETSI MANO Architectural Framework afin de traiter des cas d'utilisation de l'orchestration NFV et du gestionnaire VNF.

Toutes ces notions théoriques seront mises en pratique dans le prochain chapitre.

Chapitre IV

Implémentation et mise en œuvre

IV.1. Introduction

Après avoir étudié les notions théoriques dans les chapitres précédents, nous allons mettre en pratique toutes ces notions et retourner les résultats des tests implémentés. Pour cela nous avons devisé notre travail en trois principales parties :

- ❖ Installation et configuration de VIM en tant qu'infrastructure NFV (OpenStack) et l'intégration de MANO(Tacker) via Devstack.
- ❖ Installation et configuration de quelques fonctions NFV (Firewall, DHCP, et DNS).
- ❖ Orchestration des fonctions NFV (création et déploiement).

IV.2. Préparation de l'environnement de travail

Notre infrastructure se compose de deux serveurs puissants de 128 Go de RAM et de 1 Tera Octets d'espace mémoire chacun. Sur chaque serveur un hyperviseur de type 1 ESXI est installé. Sur l'un des serveurs 4VMs ont été installées et 2 autres VMs sont installées sur l'autre (donc en tout 6 VMs).

16 Go de RAM et 100GO de mémoire ont été alloués pour chaque VM. Les VMs sont installées à partir d'Ubuntu server 16.04. Sur ces dernières, nous avons procédé à l'installation OpenStack via Devstack avec la dernière version qui est Stein afin de déployer l'infrastructure NFV.

Pour pouvoir accéder à cette infrastructure, ATM mobilis ont mis à notre disposition un accès distant via VPN à partir de nos propres machines et ceci en utilisant l'outil forticlient qui est une suite de sécurité proposée par l'éditeur Fortinet pour protéger les ordinateurs et l'émulateur de terminal Linux pour Windows mobaXterm qui nous offre la possibilité de se connecter à distance au serveurs d'ATM Mobilis via le protocole SSH.

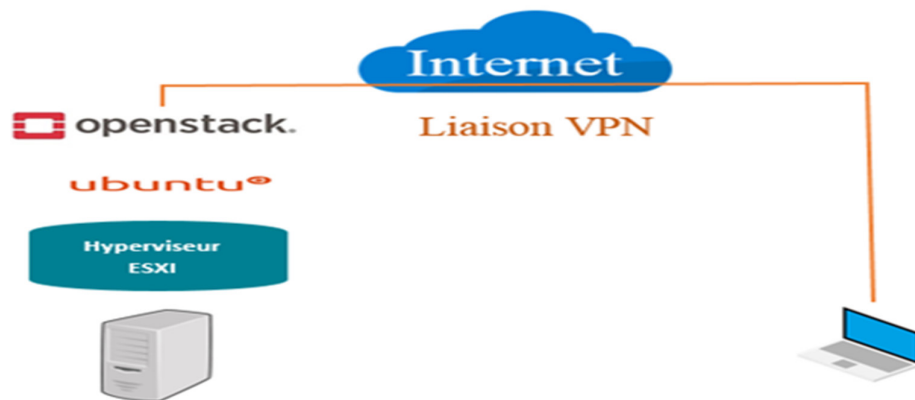


Figure IV.1. Environnement de travail

IV.3. Déploiement d'Openstack

OpenStack peut être déployé dans une configuration à un seul nœud (All-in-one), à deux nœuds (Two-nodes) ou à plusieurs nœuds (multi-nodes).

Single node : Tous les services ainsi que toutes les instances sont hébergés au sein du même serveur. Cette solution permet d'effectuer des tests sur le Cloud pour des fins purement techniques

Two nodes : Tous les services sont hébergés au sein d'un serveur Cloud controller Node, à l'exception du service Nova Compute qui est installé sur un serveur dédié Compute Node.

Multiple nodes : Un nœud supplémentaire sera ajouté à l'architecture précédente et qui va héberger les fichiers de configuration tels que nova.conf. D'autres services tels que le Volume Controller et le Network Controller peuvent être installés sur d'autres serveurs, ce qui va rendre l'infrastructure plus complexe.

Dans notre contexte, nous l'avons déployé sur un nœud simple (single-node) en se basant sur un script nommé « Devstack ».

IV.4. Définition de Devstack

Devstack est un ensemble de scripts et d'utilitaires permettant de déployer rapidement un Cloud OpenStack basé sur les dernières versions de git master. Il est utilisé de manière interactive comme environnement de développement et comme base de la plupart des tests fonctionnels du projet OpenStack.

IV.5. Installation d'Openstack

IV.5.1. Pré-requis du système

1. Afin de pouvoir installer OpenStack, nous devons d'abord modifier l'adresse IP de la carte réseau en utilisant la commande

```
root@stage6:~# sudo nano /etc/network/interfaces
GNU nano 2.5.3 File: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens160
iface ens160 inet static
    address 10.234.208.21
    netmask 255.255.255.0
    network 10.234.208.0
    broadcast 10.234.208.255
    gateway 10.234.208.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 8.8.8.8
    dns-search 8.8.8.8

auto ens192
iface ens192 inet static
    address 10.234.208.22
    netmask 255.255.255.0
    network 10.234.208.0
    gateway 10.234.208.1
```

2. Mettre le système à jour

```
root@stage6:~# apt-get update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [107 kB]
Ign:2 http://ubuntu-cloud.archive.canonical.com/ubuntu xenial-updates/queens InRelease
Get:3 http://ubuntu-cloud.archive.canonical.com/ubuntu xenial-updates/queens Release [7,890 B]
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:5 http://ubuntu-cloud.archive.canonical.com/ubuntu xenial-updates/queens Release.gpg [543 B]
Get:6 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:7 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [554 kB]
Get:8 https://download.docker.com/linux/ubuntu xenial InRelease [66.2 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:10 http://security.ubuntu.com/ubuntu xenial-security/main i386 Packages [479 kB]
Get:11 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [370 kB]
Get:12 http://security.ubuntu.com/ubuntu xenial-security/universe i386 Packages [317 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [849 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu xenial-updates/main i386 Packages [763 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [683 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages [624 kB]
Fetched 5,037 kB in 2s (1,978 kB/s)
Reading package lists... Done
```

IV.5.2. Création d'un utilisateur stack

DevStack doit être exécuté par un utilisateur non root mais ayant les permissions sudo.

Il faut donc créer un utilisateur, appelé ici « stack ».

```
root@stage6:~# sudo useradd -s /bin/bash -d /opt/stack -m stack
```

Comme l'utilisateur exécute DevStack et que ce dernier apporte des changements majeurs au système il faut le doter des permissions sudo.

```
root@stage6:~# echo "pile ALL = (TOUS) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/ stack
root@stage6:~# sudo su - stack
stack@stage6:~$ █
```

IV.5.3. Téléchargement de Devstack

Après avoir installé toutes les mises à jours nécessaires et vérifié que Python et pip sont à jour, nous allons maintenant procéder au clonage DEVSTACK qui nous permet de récupérer les fichiers de DEVSTACK contenant le script d'installation

```
stack@stage3:/$ sudo git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
remote: Counting objects: 42286, done.
remote: Compressing objects: 100% (21146/21146), done.
remote: Total 42286 (delta 29983), reused 32265 (delta 20476)
Receiving objects: 100% (42286/42286), 8.76 MiB | 4.13 MiB/s, done.
Resolving deltas: 100% (29983/29983), done.
Checking connectivity... done.
stack@stage6:~$ cd devstack
stack@stage6:~/devstack$ █
```

IV.5.4. Configuration des fichiers de DEVSTACK

Pour le bon déroulement de l'installation d'Openstack, nous devons d'abord modifier la configuration de certains fichiers qui se trouvent dans le dossier DEVSTACK.

1. Fichier stackrc

Comme le protocole git est bloqué dans notre environnement de travail, la solution consiste à modifier le fichier **stackrc** dans le dossier d'installation de devstack pour utiliser le protocole **https** au lieu du protocole **git**.

Paramètre par défaut dans le fichier **stackrc** :

```
# Base GIT Repo URL
# Another option is https://git.openstack.org
GIT_BASE=${GIT_BASE:-git://git.openstack.org} █
```

Paramètre modifié qui contourne les restrictions git :

```
# Base GIT Repo URL
# Another option is https://git.openstack.org
GIT_BASE=${GIT_BASE:-https://git.openstack.org} █
```

1. Fichier fonctions-common

Afin d'éviter un arrêt brusque de l'installation d'openstack à cause d'une lente connexion, nous devons modifier le délai d'attente (time out) 300 pour 1000

Paramètre par défaut dans le fichier **functions-common** :

```
local proxies="http_proxy=${http_proxy:-} https_proxy=${https_proxy:-} no_proxy=${no_proxy:-} "  
local update_cmd="$sudo $proxies apt-get update"  
if ! timeout 300 sh -c "while ! $update_cmd; do sleep 30; done"; then  
die $LINENO "Failed to update apt repos, we're dead now"
```

Paramètre modifié :

```
local proxies="http_proxy=${http_proxy:-} https_proxy=${https_proxy:-} no_proxy=${no_proxy:-} "  
local update_cmd="$sudo $proxies apt-get update"  
if ! timeout 1000 sh -c "while ! $update_cmd; do sleep 30; done"; then  
die $LINENO "Failed to update apt repos, we're dead now"
```

IV.5.5. Création du fichier de configuration local.conf

Créer un fichier local.conf à déposer à la racine du dossier devstack avec la commande suivante : `stack@stage6:~/devstack$ sudo nano local.conf`

Le fichier local.conf est un fichier de configuration pour l'installation de Devstack, il contient la configuration nécessaire pour lancer le script.

Tous les mots de passe, les détails des services OpenStack, la configuration des services OpenStack, sont configurés ici. Devstack utilise ce fichier de configuration pour installer et configurer les composants OpenStack.

```
[[local|localrc]]
#####
# Customize the following HOST_IP based on your installation
#####
HOST_IP=10.234.208.21

#FLOATING_RANGE=192.168.1.224/27
FIXED_RANGE=10.234.208.0/24
FIXED_NETWORK_SIZE=256
FLAT_INTERFACE=ens160
ADMIN_PASSWORD=sdn
DATABASE_PASSWORD=sdn
RABBIT_PASSWORD=sdn
SERVICE_PASSWORD=sdn
SERVICE_TOKEN=devstack
```

HOST_IP : c'est l'adresse IP de la machine hôte 10.234.208.21 dans notre cas (VM6)

FIXED_RANGE, **FIXED_NETWORK** : afin de configurer l'espace d'adressage interne utilisé par les instances

FLOATING_RANGE : Définir `floating_range` sur une plage non utilisée sur le réseau local, à savoir 192.168.1.224/27. Cela configure les adresses IP se terminant par 225-254 à utiliser comme adresses IP flottantes.

FLAT_INTERFACE : c'est l'interface Ethernet qui connecte l'hôte au réseau local. C'est l'interface qui doit être configurée avec l'adresse IP statique mentionnée ci-dessus.

Par la suite nous avons définis les mots de passe des différents modules de DevStack.

IV.5.6. Activer les plug-ins Devstack liés à Tacker dans le fichier `local.conf`

Nous avons ajoutés les services Openstack qui permettent l'orchestration NFV, à savoir Tacker et Heat

```
# Tacker
enable_plugin tacker https://git.openstack.org/openstack/tacker master
```

```
# Heat
enable_plugin heat https://git.openstack.org/openstack/heat master
```

Autres services tel que Ceilometer, Mistral, Barbican, Watcher et Networking-sfc pour le chainage.

```
# Enable heat, networking-sfc, barbican and mistral
enable_plugin heat https://git.openstack.org/openstack/heat master
enable_plugin networking-sfc https://git.openstack.org/openstack/networking-sfc master
enable_plugin barbican https://git.openstack.org/openstack/barbican master
enable_plugin mistral https://git.openstack.org/openstack/mistral master
```

```
enable_plugin watcher https://git.openstack.org/openstack/watcher
```

IV.5.7. Démarrage de l'installation

Pour lancer l'installation il faut démarrer le script se trouvant dans le dossier devstack avec la commande suivante :

```
stack@stage6:~/devstack$ ./stack.sh
```

Devstack a installé Keystone, Glance, Nova, Cinder, Neutron et Horizon. Des IP flottantes seront disponibles, les invités auront accès au monde extérieur. Dans cette installation, **Tacker Heat, Ceilometer, Barbican et Mistral** sont également installés.

A la fin de l'installation nous auront la figure suivante :

```
=====
DevStack Component Timing
(times are in seconds)
=====
run_process      52
test_with_retry  4
apt-get-update   12
pip_install      319
osc              473
wait_for_service 26
dbsync           30
apt-get          50
-----
Unaccounted time 666
-----
Total runtime    1632

This is your host IP address: 10.234.208.21
This is your host IPv6 address: ::1
Horizon is now available at http://10.234.208.21/dashboard
Keystone is serving at http://10.234.208.21/identity/
The default users are: admin and demo
The password: sdn

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future
With the removal of screen support, tail_log is deprecated and will be removed after Queens
With the removal of screen support, tail_log is deprecated and will be removed after Queens
With the removal of screen support, tail_log is deprecated and will be removed after Queens
With the removal of screen support, tail_log is deprecated and will be removed after Queens

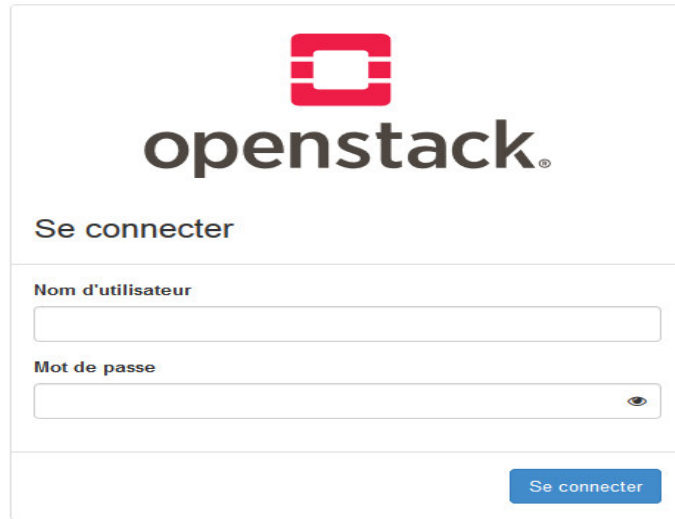
Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: stein
Change: 393c95fdef2bbb43876481bd026a7d6a336a68af Merge "Update DEVSTACK_SERIES to stein" 2018-08-29 15:34:44 +0000
OS Version: Ubuntu 16.04 xenial
```

Grace aux informations retournées tel que l'URL (host IP address **10.234.208.21**) de notre interface ainsi que le nom d'utilisateur et le mot de passe, nous pouvons accéder à horizon pour découvrir l'interface Web d'OpenStack.

La figure ci-dessous représente Horizon, qui est l'interface graphique utilisateur pour l'environnement OpenStack. Horizon fournit de nombreuses options de contrôle aux utilisateurs pour lancer des instances, ajouter des images et gérer des projets. La figure ci-dessous montre le tableau de bord OpenStack après l'installation complète

10.234.208.21/dashboard/auth/login/?next=/dashboard/



The image shows the OpenStack login page. At the top center is the OpenStack logo, a red square with a white 'O' inside. Below the logo is the text 'openstack®'. Underneath is the heading 'Se connecter'. There are two input fields: 'Nom d'utilisateur' and 'Mot de passe'. The password field has an eye icon to toggle visibility. At the bottom right is a blue button labeled 'Se connecter'.

Figure IV.2. Dashboard d’Openstack

Une fois authentifié avec l'utilisateur par défaut admin, ou bien demo, nous accédons au dashboard d'Openstack, avec différents onglets.

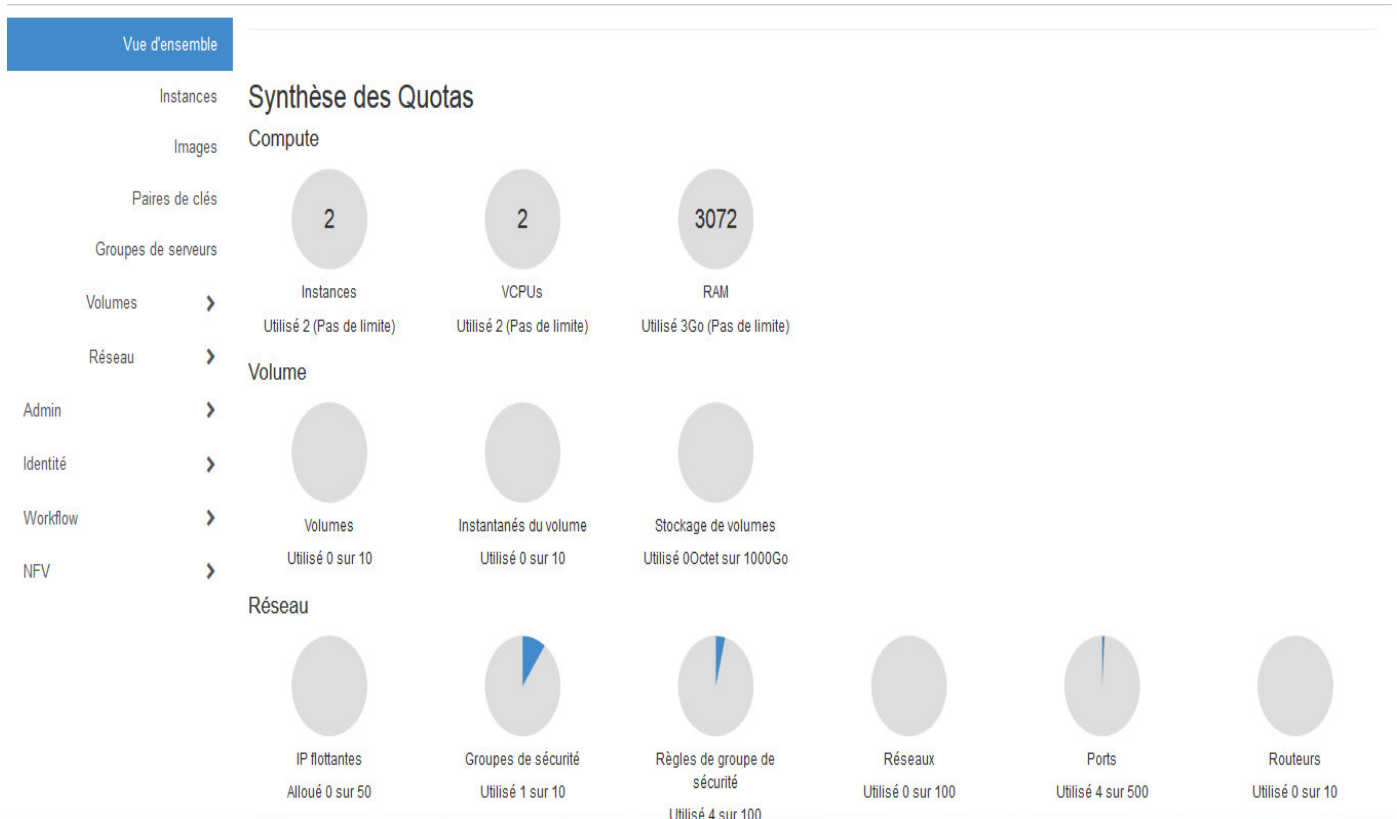


Figure IV.3. Vue d’ensemble du Dashboard d’Openstack

IV.6 Orchestration

IV.6.1. Création d'un VIM Management

Dans cette partie nous allons créer le VIM Management.

Register VIM

VIM Type *

Description :
Registers a VIM.

Nom *

Description

Auth URL *

Username *

Mot de passe *

Cert Verify *

True
 Faux

Nom de projet *

Nom de Domaine ?

Default

OpenStack sert de VIM pour Tacker.

VIM Management

Affichage de 1 élément

<input checked="" type="checkbox"/>	Nom	Description	VIM Id	Default	Auth URL	Regions	Utilisateur	Projet	Statut	VIM Type
<input checked="" type="checkbox"/>	VIM1		a0dd19cc-9e2a-438b-b98e-dd12f1993173	True	http://10.234.208.19/identity/v3	RegionOne	demo	demo	REACHABLE	openstack

Affichage de 1 élément

Figure IV.4. Etapes de création d'une VIM

IV.6.2. Création et configuration du VNF

Pour pouvoir créer un VNF une configuration réseau est nécessaire

IV.6.2.1. Configuration du réseau :

Dans cette partie, nous allons configurer notre topologie réseau qui contient des Réseaux et un Routeur avec lequel le VNF va se connecter comme le montre la figure qui suit.

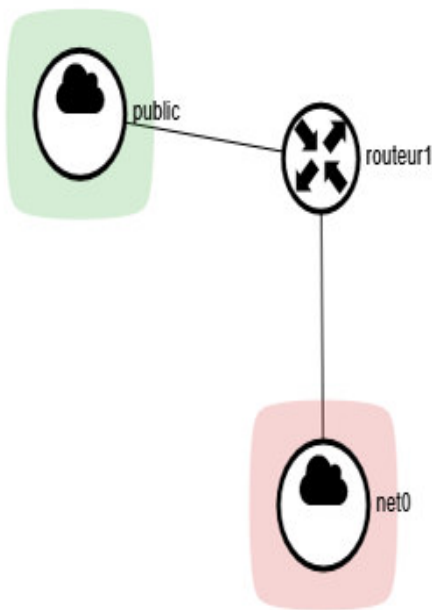


Figure IV.5. Graphique de la topologie du Réseau

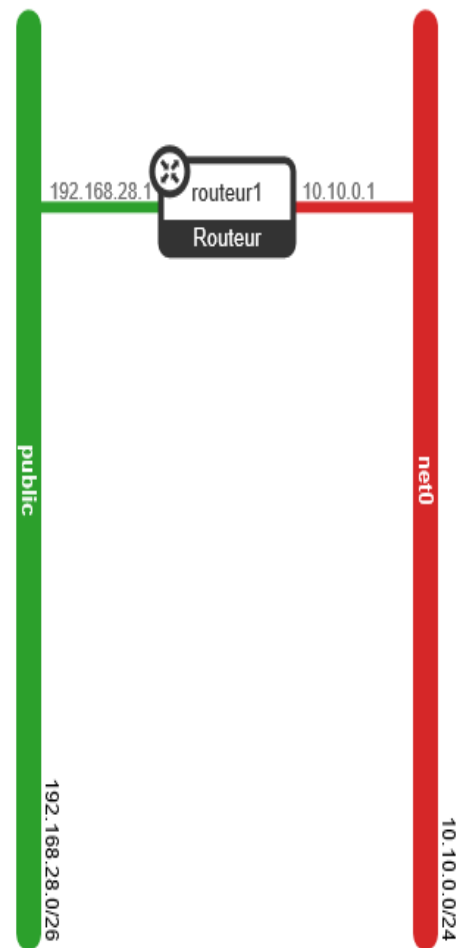


Figure IV.6. Topologie du réseau

IV.6.2.1.1. Création du réseau Public

1

Créer un réseau

Réseau
Sous-réseau
Détails du sous-réseau

Nom du réseau

Créez un nouveau réseau. En plus, un sous-réseau associé à ce réseau pourra être créé dans les étapes suivantes de cet assistant.

État Administratif Actif ⓘ

Partagé

Créer un sous-réseau

Indications de zone de disponibilité ⓘ

2

Créer un réseau

Réseau
Sous-réseau
Détails du sous-réseau

Nom du sous-réseau

Adresse réseau Source

Adresse réseau ⓘ

Version IP

Adresse IP de la passerelle ⓘ

Désactiver la passerelle

Crée un sous-réseau associé à un réseau. Vous devez entrer une "Adresse réseau" et une "Adresse IP de la passerelle" valide. Si vous n'entrez pas d'"Adresse IP de la passerelle", la première valeur (IP) de votre réseau sera assignée par défaut. Si vous ne souhaitez pas de passerelle, veuillez cocher "Désactiver la passerelle". Cliquez sur l'onglet "Détails Sous-réseaux" pour configurer des options avancées.

3

Créer un réseau

Réseau
Sous-réseau
Détails du sous-réseau

Activer DHCP

Pools d'allocation ⓘ

Serveurs DNS ⓘ

Routes d'hôte ⓘ

Spécifier les attributs additionnels pour le sous-réseau.

Figure IV.7. Etapes de création d'un réseau public

IV.6.2.1.2. Création du réseau Privé

Créer un réseau 1

Réseau | Sous-réseau | Détails du sous-réseau

Nom du réseau

État Administratif Actif ⓘ

Partagé

Créer un sous-réseau

Indications de zone de disponibilité ⓘ

Créez un nouveau réseau. En plus, un sous-réseau associé à ce réseau pourra être créé dans les étapes suivantes de cet assistant.

Créer un réseau 2

Réseau | **Sous-réseau** | Détails du sous-réseau

Nom du sous-réseau

Adresse réseau Source

Adresse réseau ⓘ

Versión IP

Adresse IP de la passerelle ⓘ

Désactiver la passerelle

Crée un sous-réseau associé à un réseau. Vous devez entrer une "Adresse réseau" et une "Adresse IP de la passerelle" valide. Si vous n'entrez pas d'"Adresse IP de la passerelle", la première valeur (IP) de votre réseau sera assignée par défaut. Si vous ne souhaitez pas de passerelle, veuillez cocher "Désactiver la passerelle". Cliquez sur l'onglet "Détails Sous-réseaux" pour configurer des options avancées.

Créer un réseau 3

Réseau | Sous-réseau | **Détails du sous-réseau**

Activer DHCP

Spécifier les attributs additionnels pour le sous-réseau.

Pools d'allocation ⓘ

Serveurs DNS ⓘ

Routes d'hôte ⓘ

Figure IV.8. Etapes de création d'un réseau privé

IV.6.2.1.3. Création d'un routeur

Créer un routeur

Nom du routeur

État Administratif Actif

Indications de zone de disponibilité

Description :

Crée un routeur avec les paramètres spécifiés.

Activer le SNAT n'aura un effet que si un réseau externe est défini.

Annuler

Créer un routeur

Ajouter une interface

Sous-réseau *

Adresse IP (facultatif)

Description :

Vous pouvez connecter un sous-réseau spécifique au routeur.

Si aucune adresse IP n'est spécifiée ici, l'adresse IP de la passerelle du sous-réseau sera utilisée pour la nouvelle interface créée dans le routeur. Si l'IP de la passerelle est déjà utilisée, vous devez spécifier une adresse différente qui appartient au sous-réseau sélectionné.

Annuler

Envoyer

Ajouter une interface

Sous-réseau *

Adresse IP (facultatif)

Description :

Vous pouvez connecter un sous-réseau spécifique au routeur.

Si aucune adresse IP n'est spécifiée ici, l'adresse IP de la passerelle du sous-réseau sera utilisée pour la nouvelle interface créée dans le routeur. Si l'IP de la passerelle est déjà utilisée, vous devez spécifier une adresse différente qui appartient au sous-réseau sélectionné.

Annuler

Envoyer

Projet / Réseau / Routeurs / routeur1

routeur1

Définir la passerelle

[Vue d'ensemble](#)

[Interfaces](#)

[Routes Statiques](#)

[+ Ajouter une interface](#)

[Supprimer les Interfaces](#)

Affichage de 2 éléments

<input type="checkbox"/>	Nom	IP fixes	Statut	Type	État Administrateur	Actions
<input type="checkbox"/>	(3f76f8fa-c285)	• 10.10.0.1	Active	Interface interne	Actif	Supprimer l'Interface
<input type="checkbox"/>	(b035eb40-ab08)	• 192.168.28.1	Active	Interface interne	Actif	Supprimer l'Interface

Affichage de 2 éléments

Figure IV.9. Etapes de création d'un Routeur

IV.6.2.2. Création de VNF Catalog

Le déploiement d'un VNF nécessite un VNFD. Le modèle VNFD comprend la configuration requise pour configurer un VNF avec OpenWRT.

Définition OpenWRT :

OpenWRT est un projet open source destiné à apporter des fonctionnalités de routage aux systèmes Linux. Traditionnellement, la plupart des routeurs sont livrés avec un micrologiciel installé sur le matériel propriétaire.

Cette approche sous-traite le contrôle de routage à l'entreprise qui fabrique le matériel. OpenWRT vise à fournir le micrologiciel de routage et à placer le contrôle de routage entre les mains de l'utilisateur. OpenWRT fonctionne actuellement sur certains systèmes matériels et est compatible avec Linux

L'ouverture de OpenWRT apporte de nombreuses applications, telles que :

- Mise en forme du trafic et qualité de service
- Capturer et analyser le trafic réseau

Le VNFD a 3 sections :

❖ Unité de déploiement virtuelle (VDU, Virtual Deployment Unit)

Met en évidence l'instance de calcul, image à utiliser, flavor à déployer.

La configuration de l'expérience comprenait la configuration d'un OpenNRT VNF.

Les propriétés de VDU pour VNF étaient 1 GO de mémoire et 5 Go de stockage. Le nom de l'image est spécifié en tant que OpenWRT. D'autres propriétés de configuration VDU peuvent être vues dans le Figure ci-dessous.

```
topology_template:
  node_templates:

    VDU1:
      type: toasca.nodes.nfv.VDU.Tacker
      capabilities:
        nfv_compute:
          properties:
            num_cpus: 1
            mem_size: 1 GB
            disk_size: 5 GB
      properties:
        image: OpenWRT
        config: |
          param0: key1
          param1: key2
        mgmt_driver: openwrt
        monitoring_policy:
          name: ping
          parameters:
            count: 3
            interval: 10
        actions:
          failure: respawn
```

Figure IV.10. Unité de déploiement virtuelle (VDU)

❖ Lien virtuel (VL)

le lien virtuel décrit le lien qui serait configuré avec le VNF. Le nom du réseau est décrit dans cette section. Cette configuration se compose de trois sous-réseaux bleus, vert et orange, visibles sur la figure ci-dessous.

```
VL1:
  type: toasca.nodes.nfv.VL
  properties:
    network_name: net_mgmt
    vendor: Tacker

VL2:
  type: toasca.nodes.nfv.VL
  properties:
    network_name: net0
    vendor: Tacker

VL3:
  type: toasca.nodes.nfv.VL
  properties:
    network_name: net1
    vendor: Tacker
```

Figure IV.11. Lien Virtuel (VL)

❖ Point de connexion (CP)

Il s'agit de la section où les liens virtuels et les unités de visualisation sont connectés ensemble. Nous pouvons avoir plusieurs points de connexion dans un modèle. Pour cette configuration, nous avons 3 points de connexion qui lient trois liens virtuels à VDU.

```
CP1:
  type: toasca.nodes.nfv.CP.Tacker
  properties:
    management: true
    order: 0
    anti_spoofing_protection: false
  requirements:
    - virtualLink:
        node: VL1
    - virtualBinding:
        node: VDU1

CP2:
  type: toasca.nodes.nfv.CP.Tacker
  properties:
    order: 1
    anti_spoofing_protection: false
  requirements:
    - virtualLink:
        node: VL2
    - virtualBinding:
        node: VDU1

CP3:
  type: toasca.nodes.nfv.CP.Tacker
  properties:
    order: 2
    anti_spoofing_protection: false
  requirements:
    - virtualLink:
        node: VL3
    - virtualBinding:
        node: VDU1
```

Figure IV.12. Point de connexion (CP)

❖ **Création de VNFD :**

Nous fournissons le VNFD suivant pour configurer le VNF. Et ce en téléchargeant le fichier toasca-vnfd-openwrt écrit en YAML¹ (Voir annexe2) comme le montre la figure ci-dessous.

• **Création du VNFD Firewall :**

The screenshot shows a web form titled "OnBoard VNF" with a close button (X) in the top right corner. The form contains the following elements:

- Nom ***: A text input field containing "VNFDFirewall".
- Description**: A text area containing "Onboards a VNF.".
- TOSCA Template Source**: A dropdown menu with "TOSCA Template File" selected.
- TOSCA Template File**: A text input field containing the file path "C:\Users\User\Desktop\template\tosca-vn" and a "Parcourir..." button.

At the bottom right of the form, there are two buttons: "Annuler" (white) and "OnBoard VNF" (blue).

Figure IV.13. Création du VNFD Firewall

• **Création du VNFD Dhcp :**

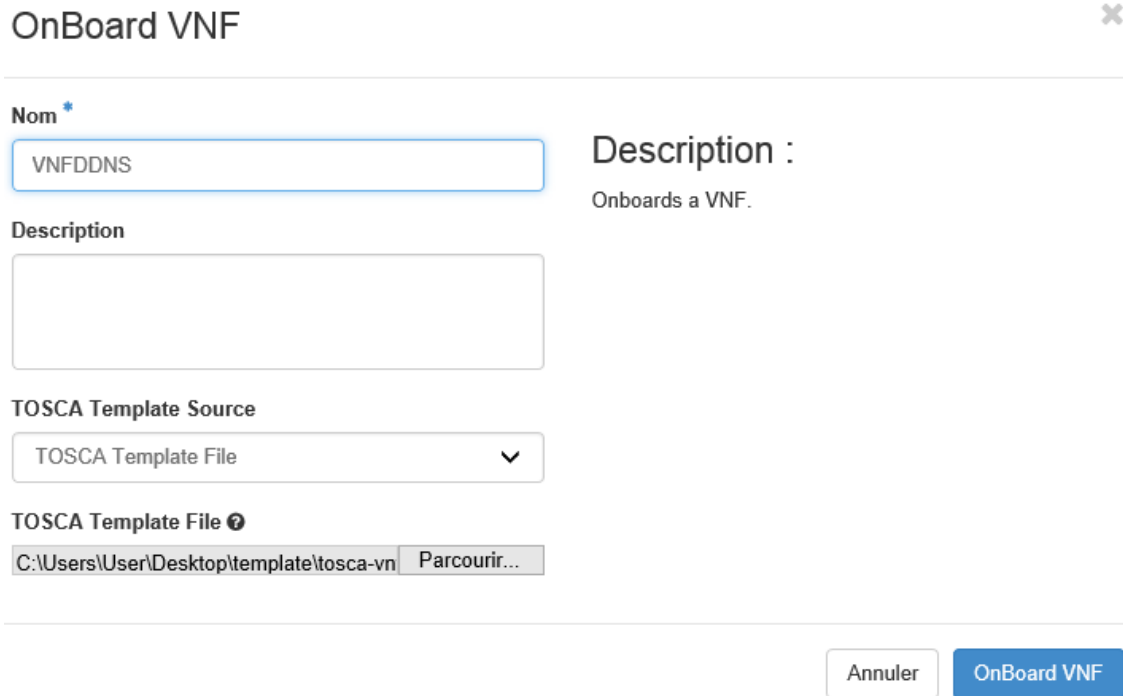
The screenshot shows a web form titled "OnBoard VNF" with a close button (X) in the top right corner. The form contains the following elements:

- Nom ***: A text input field containing "VNFD Dhcp".
- Description**: A text area containing "Onboards a VNF.".
- TOSCA Template Source**: A dropdown menu with "TOSCA Template File" selected.
- TOSCA Template File**: A text input field containing the file path "C:\Users\User\Desktop\template\tosca-vn" and a "Parcourir..." button.

At the bottom right of the form, there are two buttons: "Annuler" (white) and "OnBoard VNF" (blue).

Figure IV.14. Création du VNFD Dhcp

¹ YAML, acronyme de Yet Another Markup Language dans sa version 1.01, il devient l'acronyme récursif de YAML Ain't Markup Language (« YAML n'est pas un langage de balisage ») dans sa version 1.12, est un format de représentation de données par sérialisation Unicode. Il reprend des concepts d'autres langages comme XML, ou encore du format de message électronique

Création du VNFDDNS :

OnBoard VNF ✕

Nom *

Description :
Onboards a VNF.

Description

TOSCA Template Source
 ▼

TOSCA Template File ⓘ

Figure IV.15. Création du VNFDDNS**IV.6.2.3. Création de VNF Manager****❖ Déploiement de VNF :**

La prochaine partie consiste à déployer le VNF sur la base du modèle VNF intégré présenté ci-dessus. Lors du déploiement d'un VNF, nous devons fournir Virtual Infrastructure Manager (VIM), le Virtual Network Function Descriptor ainsi que le fichier de configuration (TOSCA) spécifique de la VNF à déployer

La figure ci-dessous décrit l'étape de déploiement de VNF.

Déploiement du VNFFirewall :

Deploy VNF ✕

VNF Name *

VNFFirewall

Description

OpenWRT Firewall

Description :

Deploys a VNF.
 If the VNFD template is parameterized, upload a yaml file with values for those parameters.
 If the VNFD template is not parameterized, any yaml file uploaded will be ignored.
 If a configuration yaml file is uploaded, it will be applied to the VNF post its successful creation.

VNF Catalog Name

VNFDFirewall

VNFD template Source

Fichier

TOSCA Template File ⓘ

Parcourir...

VIM Name

VIM1

Region Name

RegionOne

Parameter Value Source

Fichier

Parameter Value File ⓘ

C:\Users\User\Desktop\template\tosca-co

Parcourir...

Configuration Value Source

Fichier

Configuration Value File ⓘ

Parcourir...

Figure IV.16. Déploiement du VNF Firewall

Déploiement du VNFDhcp :

Deploy VNF ✕

VNF Name *

Description

VNF Catalog Name

VNFD template Source

TOSCA Template File ⓘ

VIM Name

Region Name

Parameter Value Source

Parameter Value File ⓘ

Configuration Value Source

Configuration Value File ⓘ

Description :

Deploys a VNF.
If the VNFD template is parameterized, upload a yaml file with values for those parameters.
If the VNFD template is not parameterized, any yaml file uploaded will be ignored.
If a configuration yaml file is uploaded, it will be applied to the VNF post its successful creation.

Figure IV.17. Déploiement du VNF DHCP

Déploiement du VNF DNS :

Deploy VNF✕

VNF Name *

Description

OpenWRT DNS

VNF Catalog Name

VNFD template Source

TOSCA Template File ⓘ

Parcourir...

VIM Name

Region Name

Parameter Value Source

Parameter Value File ⓘ

C:\Users\User\Desktop\template\tosca-co Parcourir...

Configuration Value Source

Configuration Value File ⓘ

Parcourir...

Description :

Deploys a VNF.

If the VNFD template is parameterized, upload a yaml file with values for those parameters.

If the VNFD template is not parameterized, any yaml file uploaded will be ignored.

If a configuration yaml file is uploaded, it will be applied to the VNF post its successful creation.

Figure IV.18. Déploiement du VNF DNS

❖ Graphe de la topologie du réseau VNF

La figure ci-dessous décrit la topologie du réseau VNF montrant les instances que nous avons créé précédemment.

Nous avons donc réalisé avec succès notre expérience

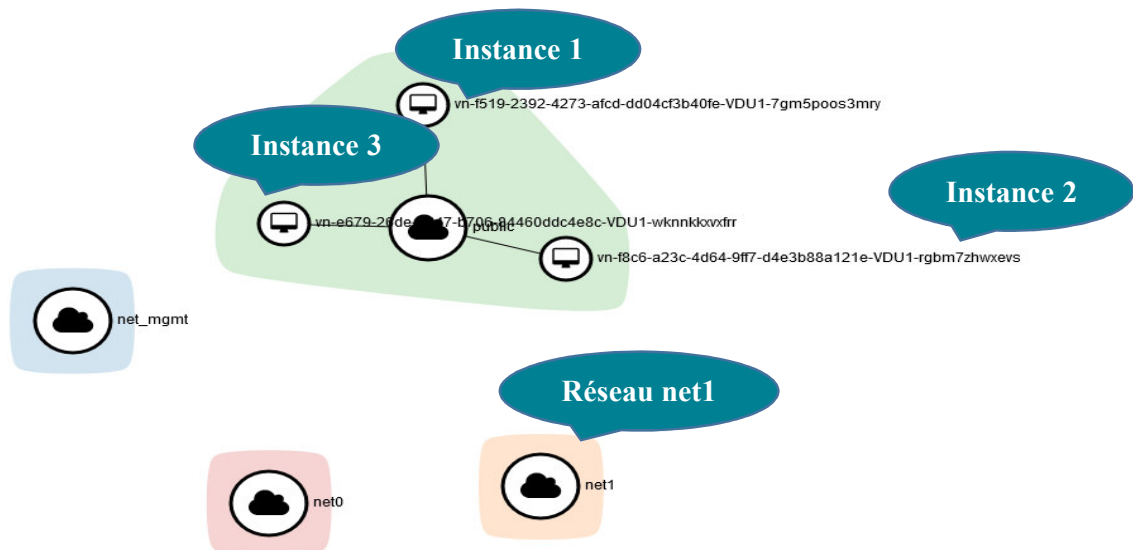


Figure IV.21. Graphe de la topologie du réseau VNF

IV.7. Conclusion :

Dans ce chapitre, nous avons présenté la maquette expérimentale, la phase de réalisation de notre projet en exposant les solutions mises en place, la démarche de travail, le principe de fonctionnement de chaque solution, ainsi que les tests effectués.

Conclusion générale

Conclusion générale

Avec l'accroissement constant des infrastructures réseaux actuelles, un nouveau modèle de fonctionnement et de déploiement devient nécessaire. SDN (Software Defined Networking) et la virtualisation des réseaux sont des technologies conçues pour répondre à ces nouvelles problématiques : elles permettent de rationaliser l'utilisation des ressources disponibles, tout en offrant une flexibilité que des équipements physiques dédiés ne peuvent fournir.

Le travail effectué dans le cadre de ce master aborde principalement la mise en place d'une infrastructure NFV basée sur Openstack.

Dans le premier chapitre, nous avons donné une idée générale sur la virtualisation et le Cloud Computing. Nous avons fait par la suite une étude sur les différentes Technologies révolutionnaires qui sont les leviers de mise à jour des réseaux pour prendre en charge la croissance de ses services.

Pour implémenter notre projet, nous avons utilisé la solution open source OpenStack qui permet, principalement, de déployer des infrastructures de *Cloud Computing* (infrastructure en tant que service).

Dans le troisième chapitre, nous avons introduit NFV-MANO qui a pour but de gérer et orchestrer les fonctions réseau virtualisées, ainsi que le projet Tacker d'Openstack afin de traiter des cas d'utilisation de l'orchestration NFV et du gestionnaire VNF

Enfin, nous avons présenté une étude de cas pratique sur toutes les notions théorique qu'on a vu dans les chapitres précédents.

- [1] <https://123virtualization.files.wordpress.com/2013/04/1-0-objectif-virtualisation-point-sur-la-virtualisation.pdf>
- [2] <https://dumas.ccsd.cnrs.fr/dumas-00587661/document>
- [3] http://www.alcantis.fr/index_fichiers/virtualisation_systeme_information.pdf
- [4] mémoire allocation de ressources dans le cloud computing. Ammour liza, chouggar chanez. Mast 2017 262
- [5] Etude et Mise en Place d'une Solution Cloud Computing Privé au sein de Tunisie Télécom, Hannachi Slim 2014/2015
- [6] <https://itandsi.files.wordpress.com/2014/09/les-fondamentaux-du-cloud-computing.pdf>
- [7] mémoire Implémentation d'un service Cloud Privé pour ATM Mobilis, Lynda CHERMAK, Katia NAILI le 14/07/2016
- [8] <https://azure.microsoft.com/fr-fr/overview/what-is-cloud-computing/>
- [9] Mise en place d'une solution Cloud de type « *Security as a Service* » et « *Storage as a Service* » sur *OpenStack* au sein d'ATM Mobilis, TAHAR DJEBBAR Abdellah Seddik, ZENOUNE Safa
- [10] <https://www.levelcloud.net/why-levelcloud/cloud-education-center/advantages-and-disadvantages-of-cloud-computing/>
- [11] MISE EN OEUVRE DES ASPECTS DE GESTION DES RÉSEAUX DÉFINIS PAR LOGICIELS (RÉSEAUX SDN), SEIFEDDINE BEN CHAHED
- [12] SEIFEDDINE BEN CHAHED, Mise en œuvre des aspects de gestion des réseaux définis par logiciel (Réseaux SDN)
- [13] Fouad BENAMRANE, Etude des Performances des Architectures du Plan de Contrôle des Réseaux (Software-Defined Networks), le 18 Janvier 2017
- [14] <http://blog.webwerks.in/data-centers-blog/southbound-vs-northbound-sdn-what-are-the-differences>
- [15] <https://searchsdn.techtarget.com/answer/The-role-of-northbound-APIs-in-an-SDN-environment>
- [16] Mohamed Said Seddiki, Allocation dynamique des ressources et gestion de la qualité de service dans la virtualisation des réseaux, le 14 Avril 2015
- [17] https://www.ciena.fr/insights/articles/What-is-NFV-prx_fr_FR.html

- [18] <https://wiki.sdn.ieee.org/pages/viewpage.action?pageId=65780>
- [19] <http://blogs.univ-poitiers.fr/f-launay/tag/nfvi/>
- [20] <https://www.slideshare.net/anandbajaj/nfv-foundationnfv-for-dummies>
- [21] <https://www.ixiacom.com/resources/network-function-virtualization-nfv-5-major-risks>
- [22] <http://www.ingrammicroadvisor.com/data-center/5-differences-between-sdn-and-network-functions-virtualization>
- [23] <https://www.openstack.org/software/>
- [24] https://access.redhat.com/documentation/en-US/Red_Hat_Storage/2.1/html-single/Configuring_Red_Hat_OpenStack_with_Red_Hat_Storage/index.html
- [25] Etude_et_mise_en_place_dune_solution_cloudcomputing_privée_pour_une_entreprise.pdf, Laribi Imen, 26-mai-2014
- [26] https://www.memoireonline.com/07/15/9194/m_Etude-sur-la-securite-du-cloud-computing25.html
- [27] <https://www.supinfo.com/articles/single/5046-decouverte-openstack>
- [28] <https://docs.openstack.org/mistral/latest/>
- [29] <https://wiki.openstack.org/wiki/Barbican>
- [30] <https://nguyentrihai.com/?p=130>
- [31] https://www.linuxfoundation.org/wp-content/uploads/2017/06/LF_CaseStudy_OPNFV_20170613.pdf
- [32] <https://www.telocloudbridge.com/blog/a-beginners-guide-to-nfvmanagement-orchestration-mano/>
- [33] https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf
- [34] <http://www.informit.com/articles/article.aspx?p=2755705&seqNum=2>
- [35] https://dspace.library.uvic.ca/bitstream/handle/1828/846/Arora_Simar_MSc_2017.pdf?sequence=3&isAllowed=y
- [36] <https://wiki.openstack.org/wiki/Tacker>

Nom	Date de sortie	Nom de composant inclus
<i>Austin</i>	21 Octobre 2010	Nova, Swift
<i>Bexar</i>	3 Février 2011	Nova, Swift, glance
<i>Cactus</i>	15 Avril 2011	Nova, Swift, glance
<i>Diablo</i>	22 Septembre 2011	Nova, Swift, glance
<i>Essex</i>	5 Avril 2012	Nova, Swift, glance, horizon, Keystone
<i>Folsom</i>	27 Septembre 2012	Nova, Swift, glance, horizon, Keystone, Neutron, cinder
<i>Grizzly</i>	4 Avril 2013	Nova, Swift, glance, horizon, Keystone, neutron, cinder
<i>Havana</i>	17 octobre 2013	Nova, Swift, glance, horizon, Keystone, neutron, cinder, heat, ceilometer
<i>Icehouse</i>	17 Avril 2014	Nova, Swift, glance, horizon, Keystone, neutron, cinder, heat, ceilometer, trove
<i>Juno</i>	16 octobre 2014	Nova, Swift, glance, horizon, Keystone, neutron, cinder, heat, ceilometer, trove, sahara
<i>Kilo</i>	30 Avril 2015	Nova, Swift, glance, horizon, Keystone, neutron, cinder, heat, ceilometer, trove, sahara, ironic, murano
<i>Liberty</i>	15 octobre 2015	Nova, Swift, glance, horizon, Keystone, neutron, cinder, heat, ceilometer, trove, sahara, ironic, murano, zaqar, searchlight, mistral, manila, designate, congress, barbican, aodh
<i>Mitaka</i>	07 Avril 2016	Aodh, barbican, ceilometer, cinder, cloudkitty, congress, designate, freezer, glance, heat, horizon, ironic, keystone, magnum, manila, mistral, monasca-api, monasca-log-api, murano, neutron, nova, sahara, searchlight, senlin, solum, swift, tacker, trove, zaqar
<i>Newton</i>	06 Octobre 2016	Aodh, barbican, ceilometer, cinder, cloudkitty, congress, designate, freezer, glance, heat, horizon, ironic, keystone, magnum, manila, mistral, monasca-api, monasca-log-api, murano, neutron, nova, panko, sahara, searchlight, senlin, solum, swift, tacker, trove, vitrage, watcher, zaqar
<i>Ocata</i>	22 février 2017	Aodh, barbican, ceilometer, cinder, cloudkitty, congress, designate, freezer, glance, heat, horizon, ironic, keystone, magnum, manila, mistral, monasca-api, monasca-log-api, murano, neutron, nova, panko, sahara, searchlight, senlin, solum, swift, tacker, tricircle, trove, vitrage, watcher, zaqar

Tableau A : Version openstack

Point de référence	Limites	Utilisation définie dans la structure
Os-Ma-nfvo	OSS / BSS <-> NFVO	<ul style="list-style-type: none"> Description du service et gestion des packages VNF. Gestion du cycle de vie du service réseau (instanciation, interrogation, mise à jour, mise à l'échelle et terminaison). Gestion du cycle de vie VNF. Gestion des stratégies (accès, autorisation, etc.) pour les instances de service réseau. Interrogation du service réseau et des Instances VNF à partir d'OSS / BSS. Transmission des événements, de l'utilisation et des performances des instances de service réseau vers OSS / BSS.
Ve-Vnfm-vnf	VNFM <-> vnf	<ul style="list-style-type: none"> Instanciation, requête d'instance, mise à jour, mise à l'échelle ou arrêt des machines virtuelles. Configuration et événements concernant VNF, de VNFM à VNF. Configuration et événements de VNF à VNFM.
Ve-Vnfm-em	VNFM <-> EM	<ul style="list-style-type: none"> Instanciation, requête d'instance, mise à jour, mise à l'échelle ou arrêt des machines virtuelles. Configuration et événements concernant VNF de VNFM à EM. Configuration et événements de EM à VNFM.
Nf-Vi	NFVI <-> VIM	<ul style="list-style-type: none"> Allouer, mettre à jour migrer, terminer les machines virtuelles. Créer, configurez, supprimez les connexions inter-VM. Événements de défaillance, enregistrements d'utilisation, informations de configuration pour le VIM pour les ressources NFVI (physiques, logicielles, virtuelles).
Or-Vnfm	NFVO <-> VNFM	<ul style="list-style-type: none"> Instanciation, requête d'état, mise à jour, mise à l'échelle, terminaison et requête de package du VNF. Transmission des événements VNF et des informations d'état.
Or-Vi	NFVO <-> VIM	<ul style="list-style-type: none"> Réservation, publication et mise à jour des ressources NFVI. Allocation, désallocation et mise à jour des images du logiciel VNF. Configuration, utilisation, événements et résultats de NFVI à NFVO.
Vi-Vnfm	VIM <-> VNFM	<ul style="list-style-type: none"> Informations sur la réservation, l'allocation et la diffusion des ressources NFVI. Événements, utilisation, résultats de mesure, etc. pour une ressource NFVI utilisée par une VNF.
Vn-Nf	NFVI <-> VNF	<ul style="list-style-type: none"> Exigences en matière de cycle de vie, de performances et de portabilité de VNF.

Tableau B : Points de référence de la structure ETSI NFV

```

tosca_definitions_version: tosca_simple_profile_for_nfv_1_0_0

description: OpenWRT with services

metadata:
  template_name: OpenWRT2

topology_template:
  node_templates:
    VDU1:
      type: tosca.nodes.nfv.VDU.Tacker
      capabilities:
        nfv_compute:
          properties:
            num_cpus: 1
            mem_size: 1 GB
            disk_size: 5 GB
      properties:
        image: OpenWRT2
        config:
          dhcp: |
            package dhcp
            # default openwrt dhcp & dns config
            config dnsmasq
              option domainneeded '1'
              option boguspriv '1'
              option filterwin2k '0'
              option localise_queries '1'
              option rebind_protection '1'
              option rebind_localhost '1'
              option local '/lan/'
              option domain 'lan'
              option expandhosts '1'
              option nongcache '0'
              option authoritative '1'
              option readethers '1'
              option leasefile '/tmp/dhcp.leases'
              option resolvfile '/tmp/resolv.conf.auto'
              option localservice '1'
            config dhcp 'lan'
              option interface 'lan'
              option start '100'

```

```

              option limit '150'
              option leasetime '12h'
              option dhcpv6 'server'
              option ra 'server'
              option ra_management '1'
              #the request to www.facebook.com will end to 1.2.3.4
            config 'domain'
              option 'name' 'www.facebook.com'
              option 'ip' '1.2.3.4'

            config 'domain'
              option 'name' 'dangtrinh.com'
              option 'ip' '192.168.1.140'
      #wgt_driver: openwrt
      monitoring_policy:
        name: ping
        parameters:
          count: 3
          interval: 10
        actions:
          failure: respawn

    CP1:
      type: tosca.nodes.nfv.CP.Tacker
      properties:
        management: true
        anti_spoofing_protection: false
      requirements:
        - virtualLink:
            node: VL1
        - virtualBinding:
            node: VDU1

    VL1:
      type: tosca.nodes.nfv.VL
      properties:
        network_name: public
        vendor: Tacker

```

TOSCA-VNFD-DNS-DHCP

```
tosca_definitions_version: tosca_simple_profile_for_nfv_1_0_0

description: OpenWRT with services

metadata:
  template_name: OpenWRT2

topology_template:
  node_templates:

    VDU1:
      type: tosca.nodes.nfv.VDU.Tacker
      capabilities:
        nfv_compute:
          properties:
            num_cpus: 1
            mem_size: 2 GB
            disk_size: 5 GB
      properties:
        image: OpenWRT2
        config: |
          param0: key1
          param1: key2
        mgmt_driver: openwrt
        monitoring_policy:
          name: ping
          parameters:
            count: 3
            interval: 10
        actions:
          failure: respawn
```

```
CP1:
  type: tosca.nodes.nfv.CP.Tacker
  properties:
    management: true
    order: 0
    anti_spoofing_protection: false
  requirements:
    - virtualLink:
        node: VL1
    - virtualBinding:
        node: VDU1

VL1:
  type: tosca.nodes.nfv.VL
  properties:
    network_name: public
    vendor: Tacker
```

TOSCA-VNFD-Firewall

```

vdus:
VDUI:
config:
  firewall: |
    package firewall

    config defaults
      option syn_flood '1'
      option input 'ACCEPT'
      option output 'ACCEPT'
      option forward 'REJECT'

    config zone
      option name 'lan'
      list network 'lan'
      option input 'ACCEPT'
      option output 'ACCEPT'
      option forward 'ACCEPT'

    config zone
      option name 'wan'
      list network 'wan'
      list network 'wan6'
      option input 'REJECT'
      option output 'ACCEPT'
      option forward 'REJECT'
      option masq '1'
      option mtu_fix '1'

    config forwarding
      option src 'lan'
      option dest 'wan'

    config rule
      option name 'Allow-DHCP-Renew'
      option src 'wan'
      option proto 'udp'
      option dest_port '68'
      option target 'ACCEPT'
      option family 'ipv4'

```

```

config rule
  option name 'Allow-Ping'
  option src 'wan'
  option proto 'icmp'
  option icmp_type 'echo-request'
  option family 'ipv4'
  option target 'ACCEPT'

config rule
  option name 'Allow-IGMP'
  option src 'wan'
  option proto 'igmp'
  option family 'ipv4'
  option target 'ACCEPT'

config rule
  option name 'Allow-DHCPv6'
  option src 'wan'
  option proto 'udp'
  option src_ip 'fe80::/10'
  option src_port '547'
  option dest_ip 'fe80::/10'
  option dest_port '546'
  option family 'ipv6'
  option target 'ACCEPT'

config rule
  option name 'Allow-MLD'
  option src 'wan'
  option proto 'icmp'
  option src_ip 'fe80::/10'
  list icmp_type '130/0'
  list icmp_type '131/0'
  list icmp_type '132/0'
  list icmp_type '143/0'
  option family 'ipv6'
  option target 'ACCEPT'

```

```

config rule
  option name 'Allow-ICMPv6-Input'
  option src 'wan'
  option proto 'icmp'
  list icmp_type 'echo-request'
  list icmp_type 'echo-reply'
  list icmp_type 'destination-unreachable'
  list icmp_type 'packet-too-big'
  list icmp_type 'time-exceeded'
  list icmp_type 'bad-header'
  list icmp_type 'unknown-header-type'
  list icmp_type 'router-solicitation'
  list icmp_type 'neighbour-solicitation'
  list icmp_type 'router-advertisement'
  list icmp_type 'neighbour-advertisement'
  option limit '190/sec'
  option family 'ipv6'
  option target 'REJECT'

```

tosca-config-openwrt-firewall

```
vdus:
VDU1:
config:
  dhcp: |
    package dhcp

    config dnsmasq
      option domainneeded '1'
      option boguspriv '1'
      option filterwin2k '0'
      option localise_queries '1'
      option rebind_protection '1'
      option rebind_localhost '1'
      option local '/lan/'
      option domain 'lan'
      option expandhosts '1'
      option nonegcache '0'
      option authoritative '1'
      option readethers '1'
      option leasefile '/tmp/dhcp.leases'
      option resolvfile '/tmp/resolv.conf.auto'
      option localservice '1'

    config dhcp 'lan'
      option interface 'lan'
      option start '100'
      option limit '150'
      option leasetime '12h'
      option dhcpv6 'server'
      option ra 'server'
      option ra_management '1'

    config 'domain'
      option 'name' 'www.facebook.com'
      option 'ip' '1.2.3.4'

    config 'domain'
      option 'name' 'www.google.com'
      option 'ip' '192.168.1.140'
```

tosca-config-openwrt-dhcp-dns