RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE POPULAIRE
MINISTÉRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ M'HAMED BOUGARA BOUMERDES

FACULTE DES SCIENCES DE L'INGÉNIEUR

DEPARTEMENT DE GÉNIE ÉLECTRIQUE

MÉMOIRE DE FIN D'ÉTUDE POUR L'OBTENTION D'UN MASTER

SPÉCIALITÉ RÉSEAUX ET TÉLÉCOMMUNICATION

THÉME

Migration de machine virtuelle en temps réel

Encadrer par: Dr.Yassine Meraihi.

Présenté par: Kiared Akila.

Jury : Dr. Hossine faiza.
Dr. Mehdi Issmahane.

2018/2019

Remerciements

Je remercie Dieu

Je remercie mes parents

Je remercie l'encadreur et les membres du jury d'avoir accepté d'évaluer ce travail

Je remercie également tous ceux qui ont aidé pendant de la réalisation de ce travail.

Résumé:

Le progrès des technologies constant à engendrée un accroissement du trafic de données, qui nécessite un réseau mobile performant qui est capables de répondre au mieux aux exigences de ces utilisateurs, ce dernier nécessitera une configuration/ reconfiguration rapide et simple des ressources réseau, par l'utilisation de la technologie SDN, qui découpe le plan de contrôle de façon centraliser du plan de données des périphérique réseau, afin de fournir aux utilisateurs une bonne Qos, le passage à l'échelle, et la haute disponibilité, au moyen des techniques ; parmi ces derniers est la migration des VMs hébergent des services, lu par un client entre deux infrastructures Cloud différentes, qui a été présentée dans ce travail, qui repose exclusivement sur la migration à l'aide de l'unité de contrôle ONOS.

Mots-clés : Cloud computing, machine virtuelle, SDN, ONOS.

ABSTRACT:

The progress of constant technologies has led to an increase in data traffic, which requires an efficient mobile network that is able to best meet the needs of these users, which will require a quick and simple configuration/reconfiguration of network resources, using SDN technology, which cuts the control plan centrally from the data plan of the network devices, in order to provide users with a good Qos, scale-up, and high-speed Availability, using techniques; among these is the migration of VMs hosting services, read by a client between two different cloud infrastructures, which was presented in this work, which is based exclusively on migration using the ONOS control unit.

Keywords: Cloud computing, virtual machine, SDN, ONOS.

ملخص

وقد أدى تقدم التكنولوجيات الثابتة إلى زيادة في حركة نقل البيانات، مما يتطلب وجود شبكة متنقلة تتسم بالكفاءة وقادرة على تلبية إحتياجات هؤلاء المستخدمين على أفضل وجه، وهو ما سيتطلب تهيئة/إعادة تشكيل سريعة وبسيطة لموارد الشبكة باستخدام تكنولوجيا SDN، التي تخفض خطة التحكم مركزيا من خطة البيانات الخاصة بأجهزة الشبكة، وذلك من أجل تزويد المستخدمين بأنظمة تشغيل جيدة وموسعة وتوفرا عالى السرعة باستخدام التقنيات؛ ومن بين هذه الخدمات ترحيل خدمات إستضافة الأجهزة الافتراضية، التي يقرأها أحد العملاء بين بنيتين مختلفتين من الهياكل الأساسية للشبكات، والتي تم تقديمها في هذا العمل، والتي تعتمد حصرا على الترحيل باستخدام وحدة التحكم في نظام التشغيل ONOS.

الكلمات المفتاحية: حوسبة الشبكات، جهاز ظاهري، SDN، ONOS.

Sommaire

Introduction générale …… 07
CHAPITRE 01 : Etat de l'art
1.1-Introduction…… 09
1.2.CloudComputing
1.2.1.Définitions……..... 09
1.2.2-Les Caractéristiques essentielle du Cloud Computing……..…. 10
1.2.3- Modèles de services ……..……... 10
1.2.3.1- Infrastructure as a Service ……..……..… 11
1.2.3.2-Platform as a Service……..… 12
1.2.3.3-Software as Service ……... 12
1.2.4- Modèles de déploiement
1.2.4.1- Cloud Public …… 12
1.2.4.2- Le Cloud privé …… 13
1.2.4.3-Le Cloud hybride……..…… 13
1.3- La virtualisation ……... 14
1.3.1.Enjeux de la virtualisation ……..…. 14
1.3.2-Définition ……... 14
1.3.3-Mécanisme……...……..... 15
1.3.3.1-Virtual Machine…… 15
1.3.3.2-Virtual Machine Manager / Hyperviseur ……..... 15
1.3.3.3-Virtual Infrastructure Manager ……..... 15
1.3.3.4-Réseau virtuel…….. 16
1.4- La Migration de VM……. .. 17
1.4.1-Notion de migration …… …17
1.4.2-Stratégies de migration de VM ……. 18
1.4.2.1-Migration à chaud…….. 18
1.4.2.2-Migration par internet …….. 19
1.4.3- Migration de la mémoire ……. 12
1.4.3.1-Migration de mémoire par pré-copie……..… 20
1.4.3.1.1-Phase d'échauffement ……. 20
1.4.3.1.2-Phase d'arrêt et de copie ……. 20
1.4.3.2- Migration de mémoire par post-copie ……..20
1.4.3.3-Migration de mémoire par post-copie hybride …….. 20

1.4.4-Les nécessiter de la migration…….…… 21
Conclusion…… 22
Chapitre 02 : Software Defined Networks - SDN
2.1-Introduction…… . 23
2.2-Network Operating System (NOS)…… …….. 23
2.3-Architecture des NOS…… 23
2.3.1-Plan de contrôle (Contrôle plane) …….. 23
2.3.2-Plan de données(Data plane/ forwarding plane)……..…….. 23
2.4-Software Defined Networks …….. ……..……. 24
2.5-Le protocole OpenFlow …… ……..……... 24
2.5.1-Structure d'un commutateur et fonctionnement d'OpenFlow……. 25
2.6- Controleurs SDN ……... 28
2.6.1-Architecture d'ONOS ……..……... 28
Conclusion…… ……... 31
Chapitre 03: Conception du système et implémentation
3.1-Introduction……..……. 32
3.2- Conception de l'architecture physique du système……..……. 32
3.3- La Description du scénario…… 34
3.4- Comparaison et choix du contrôleur…… …… 34
3.5- Implémentation de l'architecture et migration du service ……. 36
Conclusion…… …. 46
Conclusion générale…… ..48

Acronymes

CLICommand Line Interface(Interface de ligne de commande).

CC Cloud Computing (Informatique dans les nuages).

GUI graphical User Interface (Interface graphique utilisateur).

IT Information Technology (Technologies de l'information).

IOTInternet of Things (l'internet des objets).

IaaS Infrastructure as a Service(Infrastructure en tant que service).

LTE Long Term Evolution (Réseaux mobile).

NOS Network Operating System(Système d'exploitation réseau).

ODL Open Daylight.

ONOS **O**pen **N**etwork **O**perating **S**ystem **(Système d'exploitation réseau ouvert**).

OS **Operating System (Système d'exploitation).**

OVS Open vSwitch (Commutateur virtuel distribué).

OVSDB Open vSwitch Database Management Protocole (Le protocole de gestion de base de données vSwitch)

PaaS Platform as a Service (Plateforme en tant que service).

Qos Quality of Service (Qualité de service).

SaaS Software as a Service (Application en tant que service).

SDN Software Defined Networks (Réseau défini par l'application).

VM Virtual Machine (Machine virtuelle).

VMM Virtual Machine Manager (Moniteur de machine virtuelle ou Hyperviseur).

Liste des tableaux

1- Le Cloud privé. ……. 13

2- Processus de migration de VM à chaud par pré-copie …… 19

3- Comparaissent entre les trois stratégies de migration de la VM…….. 21

4- L'architecture SDN et la place d'Openflow…… 25

-

Table des figures

1- Modèle de service Cloud…….. 11
2- Modèle de déploiement d'un cloud hybride……. 14
3- Mécanisme de Virtualisation d'une machine physique…… 16
4- Concept du réseau virtuel…… 17
5- Migration de VM à chaud par pré-copie…….. 18
6- Vue simplifiée de l'architecture SDN. …… 24
7- Traitement d'un paquet dans un commutateur OpenFlow……. 26
8- Architecture Open vSwitch…… 28
9- Architecture d'ONOS……. 29
10- Intent Framework……..…. 30
11- Architecture physique du système……. . 33
12- Architecture physique du système- migration du flux……..... 36
13- ONOS CLI…….. 40
14- Mise en réseau avec Mininet…… 41
15- la topologie du réseaux- interface GUI……..… 43

Introduction générale:

Les Technologies de l'information et de la communication (TIC) qui désigne le domaine de la télématique a connu ces dernières années une explosion d'innovation dans le monde connecté, plus précisément des mobiles, débutant par des dispositifs mobiles monochromes caractérisés par des puissance de calcul très limitées, jusqu'à arriver à ceux qu'on connait actuellement, des écrans offrants des résolution et des puissance de calcul meilleurs. Ce progrès constant a engendré un accroissement du trafic de données, qui nécessite un réseau mobile performant, d'où interviens les chercheurs pour améliorer des technologies comme la technologie réseaux mobile (LTE).

En trouve d'autres technologies en immersion ces dernières années comme l'internet des objets (IOT) [16].[17] , elle a pour objectif de rendre le monde réel plus intelligent grâce à la connexion des objets, ces derniers obtiennent des informations qu'elles transmettent par réseau. Donc tout objet connecté à Internet et qui peut être contrôlé à distance fait partie de l'univers IoT. [18]Certaines statistique prévoient jusqu'à 6,4 milliards d'objets sont connectés dans le monde en 2016. Une augmentation de 30% par rapport à 2015. Les experts prédisent que les objets connectés seront plus de 20,8 milliards en 2020.

Par conséquent, les fournisseurs de services n'ont d'autres choix que d'augmenter leurs investissements afin de répondre efficacement aux besoins grandissants de leurs clientèles, [19]. [20] Comme une bonne qualité de service (Qos)[1], le passage à l'échelle par l'ajout à la demande des utilisateurs, de nouveaux dispositifs IOT, etc .Enfin, améliorer la productivité grâce à la gestion centralisée de l'infrastructure virtuelle. Parmi ces objectifs est de minimiser les coûts des délais d'arrêt des services dus à la migration simultanée de plusieurs machines virtuelles (VMs)[2] [23], que nous allons présenter dans les chapitres suivants.

Le reste de ce mémoire est organisé comme suit:
Un premier chapitre intitulé «état de l'art» présente les notions fondamentales Cloud Computing, les catégories des services, les modèles de déploiement, aussi la virtualisation, le mécanisme et ses avantages, ensuite la migration des VMs, ses étapes ses raisons d'effectuation, et les différentes solutions disponibles pour la création d'un environnement Cloud.

1-Répondre au mieux aux besoins des utilisateurs en termes de haute disponibilité, de bande passante suffisante, de latence, de sécurité des données, etc.
2- un système d'exploitation (OS) installé sur un logiciel qui imite un matériel dédié. L'utilisateur final a la même expérience sur une machine virtuelle que dans le matériel dédié.

Un deuxième chapitre intitulé « Software Defined Networks (SDN)», présente quelques limitations des réseaux traditionnels ainsi que la grande difficulté quant à leur administration et gestion. Ensuite détaillé comment l'introduction de la nouvelle technologie
SDN a remédié à ces limitations.

Dans la même partie nous allons aussi exposé l'architecture de SDN et la structure d'un commutateur et fonctionnement d'OpenFlow.En suite, nous allons également présenter quelques contrôleurs open-source les plus utilisés.

Dans le troisième chapitre, nous allons aborder l'implémentation d'une architecture physique de réseau, et la présentation du type de contrôleur SDN utilisé pour la configu- ration des dispositifs réseaux, et le processus de migration du flux entre VM.

Chapitre 01:Etat de l’art

1.1.Introduction:

La Direction Informatique a le rôle d’étudier, mettre en place, et maintenir des systèmes informatiques fiables (matériel et logiciel) permettant d’autoriser les procédures administratives et techniques, développer des applications de gestion, etc, conformément aux besoins exprimés par l’utilisateur et par les clients. Parmi les technologies utilisée par les fournisseurs de services afin de subvenir aux ces besoins est la technologie du Cloud Computing (CC).

L’apparition du CC vient d'une évolution de certaines technologies telles que la virtualisation du matériel informatique. En effet, cette notion permet d'optimiser les ressources matérielles en les partageant entre plusieurs environnements dans le but de pouvoir exécuter plusieurs systèmes virtuels sur une seule ressource physique et fournir une couche supplémentaire d’abstraction du matériel.

Au niveau de ce chapitre, nous allons présenter les notions fondamentales de CC, virtualisation et la migration de VM. Commencerons par expliquer son principe, les différents services qu’il fournit le Cloud, ainsi que ses modèles de déploiement.

1.2. Cloud Computing:

1.2.1-Définitions:

CC [1], ou informatique dans les nuages, tel qu’il défini par l’institut national des normes et de la technologie(NIST), est un modèle qui permet un accès pratique, partagé, et à la demande des utilisateurs, à un ensemble de ressources informatiques configurables (par exemple : des réseaux, des serveurs, espace du stockage, des applications et des autres services).

Donc, le Cloud Computing consiste à la mise à disposition des ressources informatique sous forme de services à la demande. De cette façon, les applications et les données ne se trouvent plus sur le poste de l'utilisateur, mais dans un nuage composé d'un certain nombre de serveurs distants interconnectés par l'intermédiaire d'un réseau, généralement Internet, au moyen d'une excellente bande passante indispensable à la fluidité du système, pour stocker des données ou les exploiter.

Les utilisateurs peuvent déployer des VMs dans ce nuage, ce qui leur permet d'utiliser un certain nombre de ressources (espace disque, mémoire vive, ou encore du CPU processeur) et bénéficier de multiple services accessibles à partir d'un ordinateur, d'un téléphone, d'une tablette ou tout autre appareil ou moyens pouvant s'en servir.

1.2.2-Les Caractéristiques essentielles du Cloud :

Le Cloud Computing s'appuie sur une architecture client serveur, parmi les autres caractéristiques essentielles qu'il possède:

-Les serveurs qui forment le Cloud sont hébergés dans des data centers. La plupart de ces centres comportent des dizaines de milliers de serveurs et des moyens de stockage pour permettre des montées en charge rapides;

-Le Cloud est un service mesurable, La facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Une unité de traitement stoppée n'est pas facturée. le client paye la quantité des ressources qu'il a consommées.

Remplacer un parc de serveurs physiques par une architecture virtualisée constitue une véritable révolution dans la conception des systèmes d'information, tout en assurant la performance et la disponibilité des services. Cette notion a évolué encore plus, et a fait apparaître un nouveau terme de Cloud Computing, qui se base sur la virtualisation.

Ces spécificités font de la technologie Cloud Computing une nouvelle option qui offre à ses utilisateurs la possibilité d'accès à des logiciels et à des ressources informatiques avec la flexibilité et la modularité souhaitées et à moindre coûts.

1.2.3- Modèles de services [29]:

Les types de Cloud, indiquent la responsabilité du fournisseur de Cloud et du client. Il existe trois modèles de service:

1.2.3.1- Infrastructure as a Service (IaaS);
1.2.3.2- Platform as a Service (PaaS);
1.2.3.3- Software as a Service (SaaS);

Ces modèle de service peut être déployé de manière différente :
Cloud privé;
Cloud publique;
Cloud hybride;

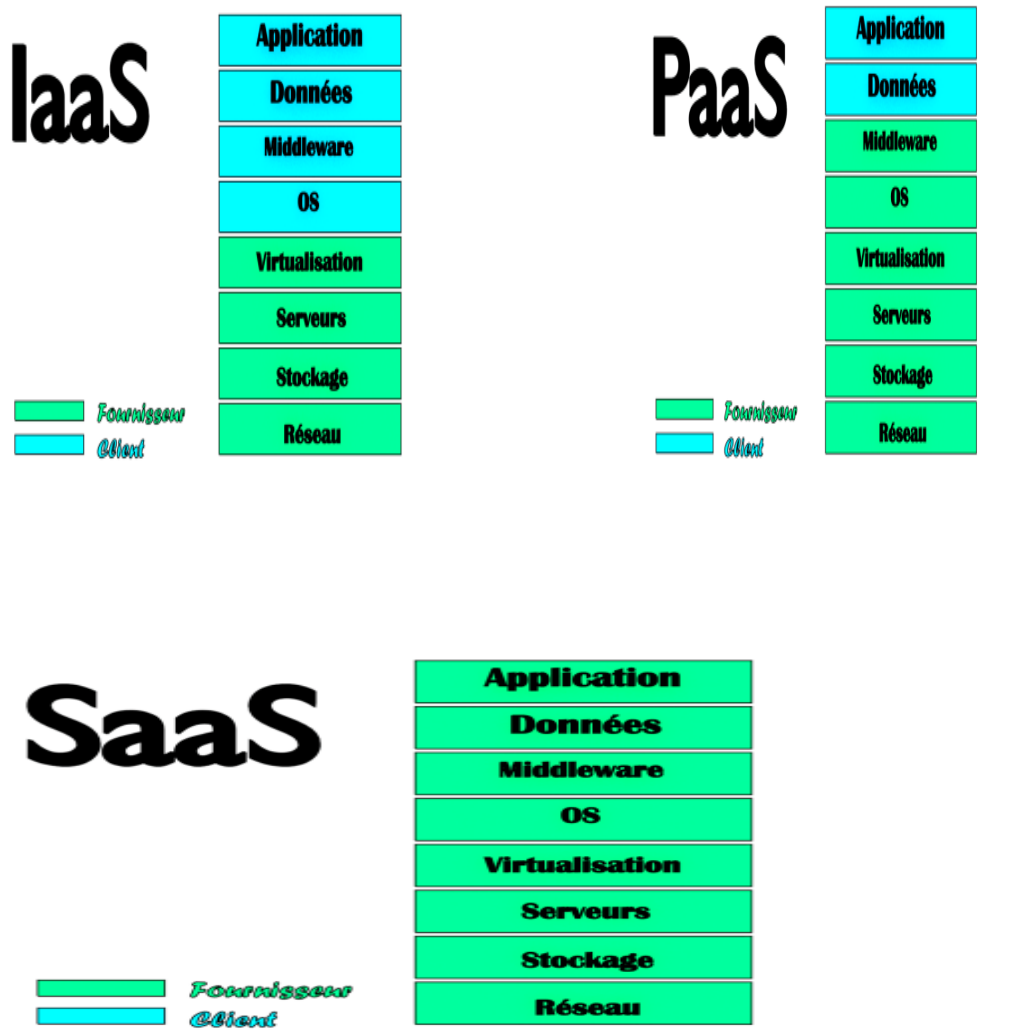


Fig.1.1- Modèles de service Cloud.

1.2.3.1- Infrastructure en tant que service:

Le fournisseur propose la location de composants informatiques virtuelle (Expliqué ci-dessous) comme des espaces de stockages, une bande passante, des systèmes d'exploitation et d'autres ressources informatiques fondamentales que l'utilisateur pourra exploiter ou contrôler dans des limites. Les utilisateurs d'une IaaS peuvent donc utiliser ces composants à la demande des serveurs virtuels sans avoir à gérer les machines physiques (coûts de gestion, remplacement de matériel, etc), et tous contrôles de l'infrastructure Cloud sous jacente.

L’IaaS offre une grande flexibilité, avec une administration à distance, et permet d’installer tout type de logiciel. En revanche, cette solution nécessite la présence d’un administrateur système au sein de l’entreprise.

1.2.3.2- Plateforme en tant que service:

Dans ce modèle de service Cloud, une plateforme d’exécution sur laquelle l’entreprise utilisatrice va pouvoir installer, configurer, contrôler et utiliser les applications voulues, en utilisant les outils nécessaires du fournisseur de services qui déploie pour le développement de logiciels (langage de programmation, bibliothèques, outils), telle que l’infrastructure qui offre le service (système d'exploitation, réseau, serveurs, stockage) et les outilsd'infrastructure sont sous la responsabilité de fournisseur.

1.2.3.3- Application en tant que service:

Ce modèle de service est caractérisé par l'utilisation d'une application partagée qui fonctionne sur une infrastructure Cloud d’un fournisseur de services. L'utilisateur accède à l'application par le réseau au travers de divers types de terminaux (par exemple *via* un navigateur web, a travers un service de messagerie électronique comme Gmail pour l’accès au courrier électronique). L’abonné ne gère et ne contrôle pas l’infrastructure qui offre le service (système d'exploitation, réseau, serveurs, stockage, etc).

De cette façon, l'utilisateur final n'a plus besoin d'installer tous les logiciels existants sur sa machine de travail ce qui réduit par la suite la maintenance en supprimant le besoin de mettre à jour les applications.
Les solutions SaaS constituent la forme la plus répandue de CC.

1.2.4- Modèles de déploiement:

1.2.4.1- Cloud Public:

Dans ce modèle de déploiement les ressources informatiques sont accessibles depuis un réseau public, son utilisation est fournie et disponible, elle est gérée, exploitée et partagée entre plusieurs clients par des organisations.

Cette infrastructure est possédée par une organisation de vendre des services de Cloud computing.

1.2.4.2- Le Cloud privé:

Ce modèle à une typologie de Cloud la plus répandue; 73% des Clouds déployés dans les entreprises selon des statistiques fait en 2013 sont des Clouds privés [2].

L'infrastructure Cloud privé est utilisée par une seule organisation, et destinés à satisfaire les besoins propres d'une seule entreprise. Elle peut être gérée par ce client ou par un prestataire de service et peut être située dans les locaux de l'entreprise cliente ou bien chez le prestataire.

On distingue deux types de Cloud privé : interne et externe.

Les Cloud privés internes	*Les Cloud privés externes*
les serveurs hébergeant les services sont localisés dans l'entreprise. Ces serveurs sont accessibles à travers un réseau sécurisé, interne et fermé.	Les ressources physiques sont hébergées chez un prestataire. - l'infrastructure est accessible via des réseaux sécurisés de type VPN.
Ce type de Cloud est créé, administré et géré en interne par une entreprise.	l'infrastructure est exploitée par l'entreprise.

Tableau -1- Le Cloud privé.

L'utilisation d'un nuage privé permet de garantir, par exemple, que les ressources matéri- elles allouées ne seront jamais partagées par deux clients différents.

1.2.4.4-Le Cloud hybride:

Cette infrastructure Cloud est composé de deux ou plusieurs types de nuages distinctes énumérés ci dessus (privées, publique) qui restent des entités uniques, communiquent via une connexion cryptée, et qui sont connectés via la technologie standardisée qui permet la portabilité des données et des applications entre ses différentes infrastructures. L'utilisation d'un cloud hybride peut grandement faciliter la connectivité sur le lieu de travail. Outre la gestion des fichiers, les entreprises doivent y intégrer différents processus métiers, tels que la messagerie interne, la planification des tâches, l'informatique décisionnelle, le traitement analytique et d'autres systèmes. Les offres de cloud public seules ne s'intègrent pas facilement (voire pas du tout) au matériel sur site. Les appareils tels que les imprimantes, scanners, fax, ainsi que le matériel de sécurité physique comme les caméras de sécurité, les détecteurs d'incendie, peuvent être des handicaps à l'adoption du cloud public. Plutôt que d'isoler ces appareils cruciaux du reste du réseau de l'entreprise, l'utilisation d'un composant de cloud privé serait beaucoup plus efficace.

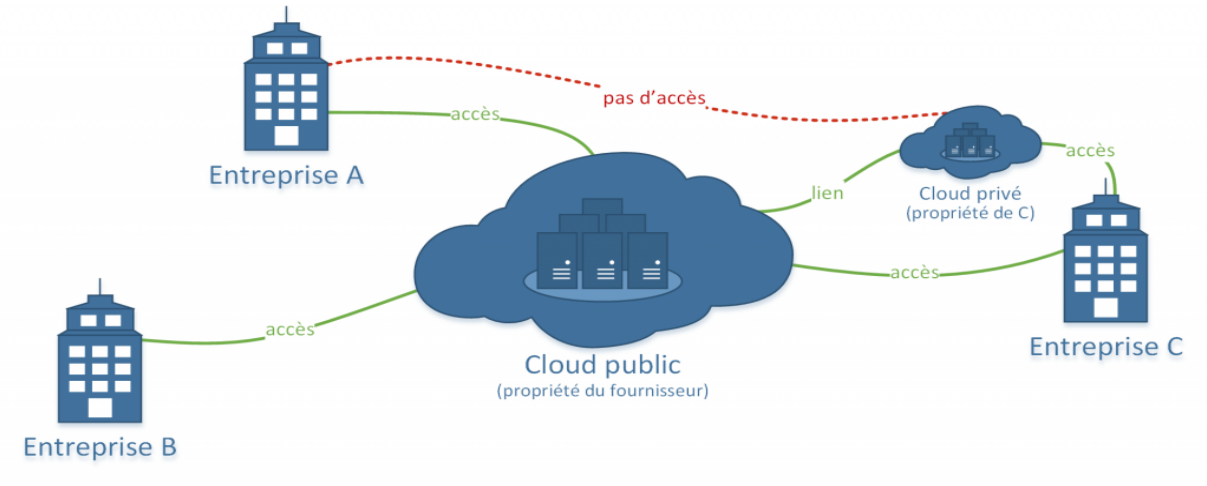


Fig.1.2 - Modèle de déploiement d'un Cloud hybride [3].

1.3- La virtualisation:

1.3.1-Enjeux de la virtualisation:

Actuellement, les entreprises rencontrent des besoins.

- Pour la maintenance, il serait utile d'améliorer des services tels que la disponibilité des applications,

-la flexibilité et le temps de réponse,
-Réduction des coûts énergétiques, d'infrastructure et des installations.
-Accès aux applications et aux données des bureaux en tout lieu.

Parmi les solutions pour répondre à ces besoins est :

La mise en œuvre de la virtualisation de matériel informatique, qui a un intérêt parmi les autres de la migrationdes machines virtuelles.

1.3.2-Définition de La virtualisation:

La virtualisation [8].[9].[28] recouvre l'ensemble des techniques matérielles et/ou logiciels qui permettent de faire fonctionner sur une seule machine plusieurs environnements d'exécution virtuelle (machines virtuelles) qui sont créés sur cette et seule machine physique.

Plusieurs instances différentes et cloisonnées d'un même système ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes. Ces partitions permettaient un traitement multitâche, à savoir l'exécution simultanée de plusieurs de ces applications et processus.

1.3.3-Mécanisme de la virtualisation:

1.3.3.1-Virtual Machine:

Une VM est un environnement d’exécution virtuel créé à partir d’une image pour instancier cette machine. Parmi les phases qui comportent le cycle de vie d’une machine virtuelle: create, suspend, save, migrate et destroy.

1.3.3.2-Virtual Machine Manager / Hyperviseur:

La mise en œuvre d’une VM nécessite l’ajout d’une couche logicielle à la machine physique. Cette couche se place entre le matériel et le système d’exploitation et s’appelle hyperviseur ou moniteur de machine virtuelle (VMM). Les machines virtuelles tournant sur un même nœud physique sont gérées et contrôlées par le VMM.

1.3.3.3-Virtual Infrastructure Manager:

Le Virtual Infrastructure Manager (VIM) est responsable de la gestion centralisée de l’infrastructure virtuelle (machine virtuelle, réseaux virtuels et de stockage virtuel). Il fournit les fonctionnalités basiques pour le déploiement, le contrôle et la supervision des VMs tournants sur différents nœuds physiques.

Un système d'exploitation principal appelé « système hôte» [8] est installé sur un serveur physique unique. Ce système sert d'accueil à d'autres systèmes d’exploitation.

Le VMM est installé sur le système d'exploitation principal, il attribut à chaque VM son besoin de ressources du serveur physique (mémoire, espace disque, etc), il redirige les requêtes d'entrées sorties vers les ressources physiques, il permet la création d'environnements indépendants sur lesquels seront installés d'autres systèmes d'exploitation «systèmes invités ». Ces environnements sont des VMs.

Un système invité est installé dans une machine virtuelle qui fonctionne indépendamment des autres systèmes invités dans d'autres VMs.

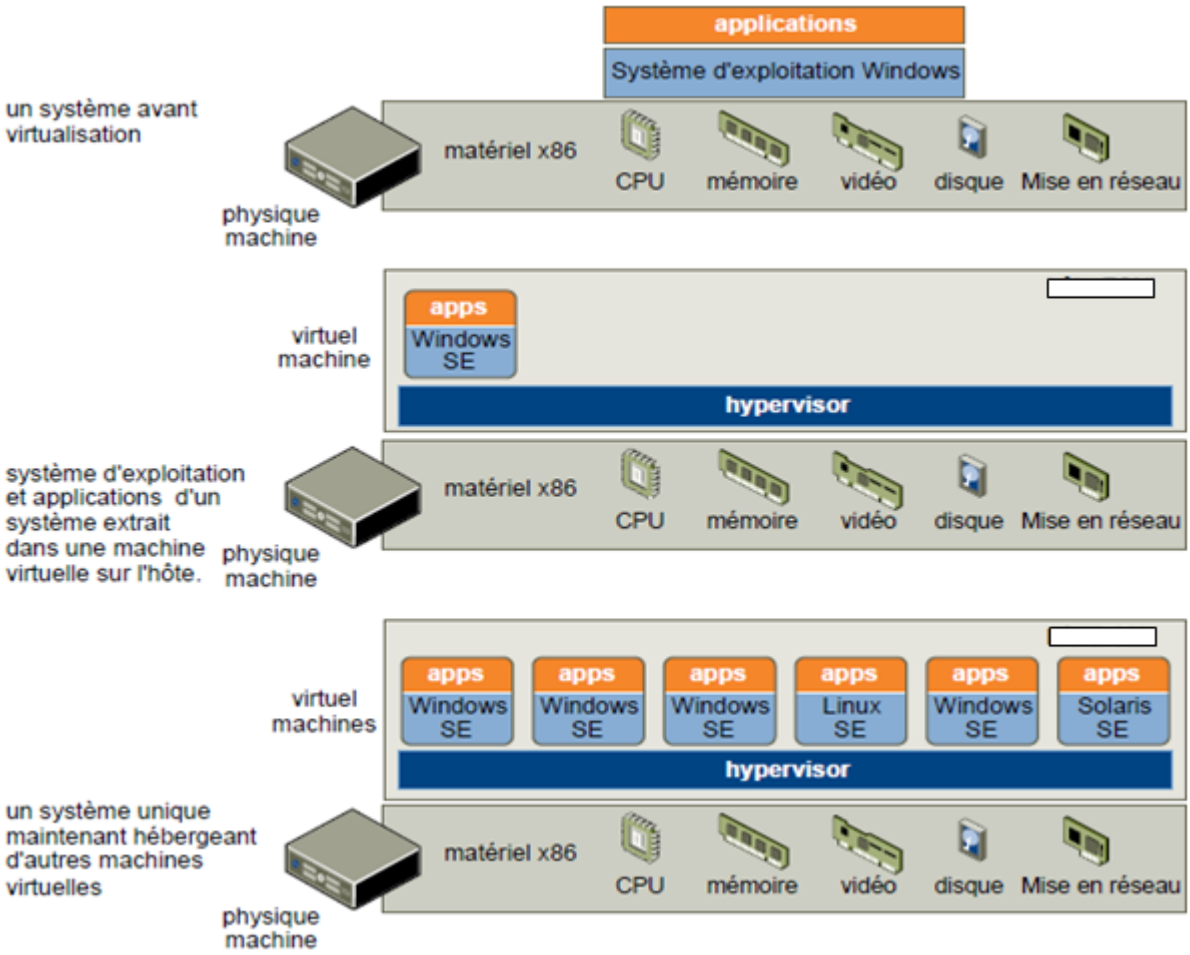


Fig.1.3- Mécanisme de Virtualisation d'une machine physique [10].

1.3.3.4-Réseau virtuel:

Dans les hyperviseurs de base, une machine virtuelle possède une ou plusieurs cartes d'interfaces réseau virtuelles.

Un commutateur virtuel est un logiciel intégré à l'hyperviseur qui se comporte comme un commutateur matériel.

Dans un environnement virtualisé, des VMs peuvent être reliées les unes aux autres sur un réseau virtuel implémenté au dessus d'une infrastructure réseau physique.

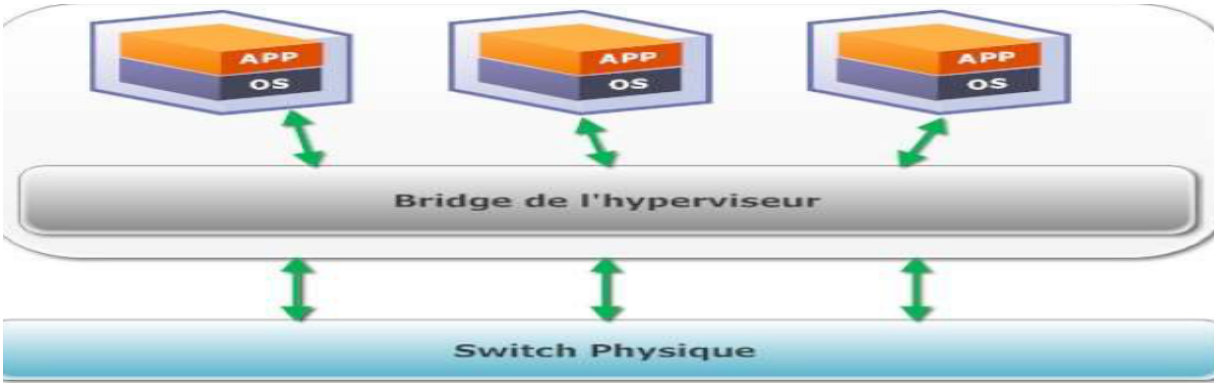


Fig.1.4- Concept du réseau virtuel.

1.4- La Migration de VM :

Il y a quelques années, la migration des machines virtuelles ne se faisait pas en temps réel, mais elle se réalisait seulement après un arrêt complet des VMs. Cela pouvait s'expliquer par le manque d'outils automatiques pour une migration en temps réel des VMs [27]. Cependant, en réponse à la croissance marquée du nombre des VMs et des réseaux virtuels, des outils qui réalisent cette dernière tache sont devenus indispensables, de bonnes techniques sont aussi nécessaires. Ces techniques ont l'objectif de minimiser les coûts des délais d'arrêt des services dus à la migration de VMs.

Cependant,toute migration en temps réel des VMs doit permettre la continuité des fonctionnements des services de ces VMs lors de leurs migrations, en diminuant le temps total de migration et des temps d'arrêt des services hébergés sur des machines virtuelles, afin de respecter les contrats de niveaux de service et la qualité globale des services virtuels [26].

1.4.1-Notion de migration [11]:

La migration de machines virtuelles consiste à déplacer l'état d'une machine virtuelle, ses fichiers, etc, d'un hôte vers une autre infrastructure physique. Cette méthode peut être abordée de plusieurs façons [31].

1.4.2-Stratégies de migration de VM:

1.4.2.1-Migration à chaud:

La migration à chaud est une stratégie utilisée qui permet à un hyperviseur de transférer, une VM d'un hôte à un autre sans être obligé de l'arrêter.

En effet, l'hyperviseur de l'infrastructure Cloud ou se situe initialement la VM doit être en mesure de transmettre les informations nécessaires à l'hyperviseur de l'infrastructure Cloud ou se situe la VM après sa migration.

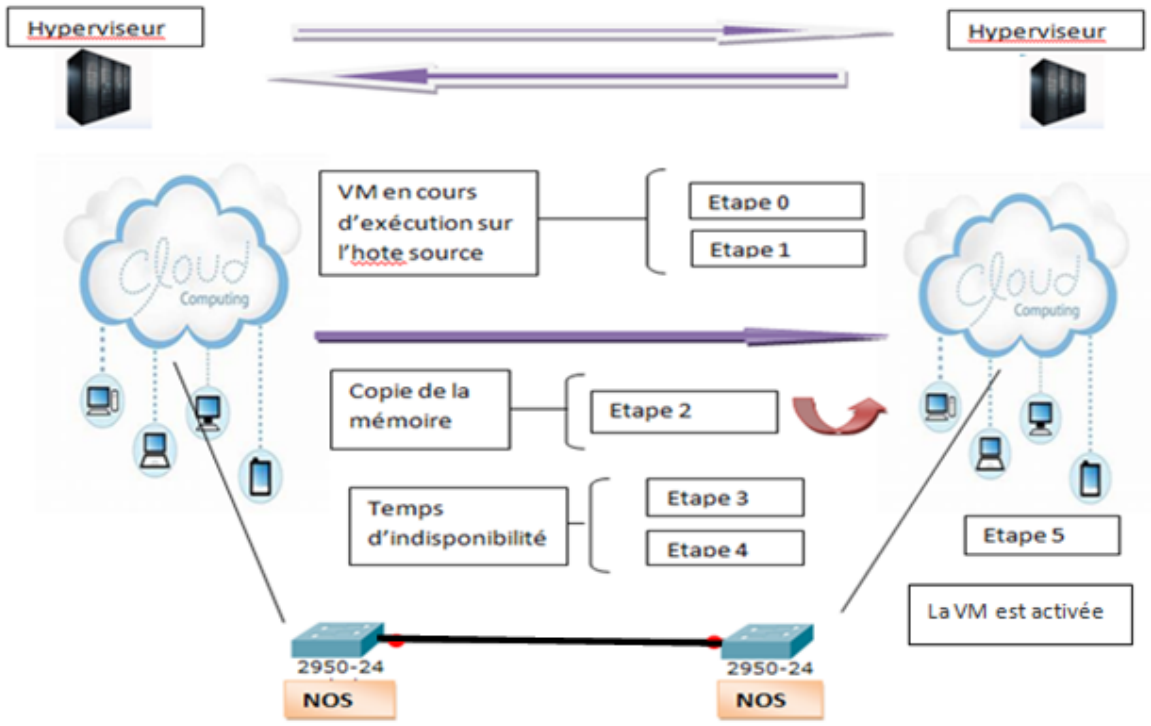


Fig.1.5- Migration de VM à chaud par pré-copie.

Etape 0: Pre-migration	la VM source est en marche, durant cette étape l'hyperviseur est à la recherche d'un hôte de destination qui héberge la machine à migrer.
Etape1: Réservation	L'hyperviseur se charge de réserver un conteneur sur l'hôte distant en lui envoyant une requête de réservation en y indiquant les ressources nécessaires à la VM (contenues de la mémoire, du disque, et les interface réseaux). Si la requête est acceptée par l'hôte distant, la phase de pré-copie peut démarrer.
Etape 2: Pré-copie itérative	Durant cette phase, l'hyperviseur copie les pages de mémoire sur l'hôte distant. Et d’autres pages seront envoyées par la suite à travers le réseau si sont modifiée sur l'hôte source Lors de transfert des pages précédentes (Cette étape est répétée).
Etape 3: Arrêt et copie	la VM est arrêtée sur l'hôte source et les pages modifiée précédemment restantes sont copiées sur l'hôte cible. Elle ne sera active sur aucun hôte. C’est la période pendant laquelle le service n’est pas disponible.
Etape4: Validation	l'hôte de destination indique à l'hôte source qu'il a bien réceptionné toute l'image de la VM. L'hôte source envoie un accusé de réception à l'hôte de destination.
Etape5: Activation	La VM est activée sur l'hôte cible et la table de routage du switch est mise à jour.

Tableau 2- Processus de migration de VM à chaud par pré-copie.

Cette implémentation de la migration, est réalisée par la plupart des hyperviseurs (Xen, V-Mware, KVM, etc). Cependant, lorsqu’une VM est en marche, il est important de minimiser le temps d’arrêt (down time), et le temps de migration total (total time of migration), qui représente la durée entre le début de la migration et lorsque la VM d’origine peut finalement être abandonnée.

1.4.2.2-Migration par internet:

Chaque ordinateur connecté à internet devient donc un hôte potentiel pour héberger une VM. En virtualisation, la mémoire de la VM étant généralement stockée sur un système

partagé (*Network File System*), il est nécessaire que la machine cible ait également accès à l'espace de stockage de la machine source.

Il existe d'autres stratégies de migration de la mémoire lors de migration à chaud :

1.4.3- Migration de la mémoire:

1.4.3.1-Migration de mémoire par pré-copie:

1.4.3.1.1-Phase d'échauffement:
Voir les étapes 0, 1, 2, décrites précédemment.

1.4.3.1.2-Phase d'arrêt et de copie:
Voir les étapes 3, 4, 5 , décrites précédemment.

1.4.3.2- Migration de mémoire par post-copie:

-l'arrêt du système sur l'hôte source;

-L'état du processeur est copié sur l'hôte cible;

- La machine est **ensuite redémarrée** sur l'hôte cible;

Ensuite, les pages mémoire sont copiées à l'aide d'une combinaison de 4 techniques:

-*demand-paging*: permet à l'hôte cible de demander une page mémoire manquante à l'hôte source;

-Active-push: permet à l'hôte source d'envoyer à la cible les pages mémoire fautives qui n'ont pas encore été demandées par l'hôte cible;

- *Pre-paging:* prédire la prochaine page fautive sur l'hôte cible en fonction des défauts de p-age récents;

- Dynamic self balooning: permet quant à lui de réduire le nombre de transferts des pages non allouées;

1.4.3.3-Migration de mémoire par post-copie hybride:

-Une seule phase de pré-copie est effectuée (étapes 0,1 ,2);

-La VM est stoppée Une fois la copie effectuée;

-Les pages modifiées sont copiées à leur tour sur la machine cible;

-La VM est ensuite redémarrée sur l'hôte cible;

-la phase de post-copie commence;

	le temps d'arrêt de service	le temps total de migration
Migration par pré-copie	-Est proportionnelle au nombre des pages qui ont été modifié lors de la copie de mémoire	-Proportionnels à la quantité de mémoire physique allouée à la VM et le downtime.
Migration par post-copie	-Est le temps de redémarrage de la machine cible et la copie de l'etat de cpu. -En post-copie le downtime est moins long qu'en pré copie. -La post-copie est plus optimisée pour les applications nécessitant beaucoup d'écritures en mémoire.	-Proportionnels à la quantité de mémoire physique allouée à la VM, et le downtime, et le temps des 4 techniques pour copier les pages mémoires manquantes ou modifier.
Migration par post-copie-hybride	-Est proportionnelle au nombre des pages qui ont été modifié lors de la copie de mémoire, et le temps de redémarrage.	-proportionnels à la quantité de mémoire physique allouée à la VM, et le le temps d'arrêt, et le temps des 4 techniques de poste-copie.

Tableau 3- Comparaissent entre les trois stratégies de migration de la VM.

1.4.4- Les nécessités de la migration:

En outre, à ce qui a été mentionné plus tôt la migration des VMs est nécessaire dans les cas suivants:

Tolérance au pannes:

S'il y a une défaillance d'un hôte ou s'il est besoin d'être arrêté pour des raisons de maintenance, afin de ne pas interrompre les services s'y trouvaient, nous devons migrer ces VMs vert un autre site [12].

Exigences d'expansion dynamique des ressources:

La répartition des ressources et la distribution de la VM de l'utilisateur dans le centre de données du Cloud. Par Ex: Les VMs d'un abonné devraient être situées aussi près que possibles de lui, afin d'éviter la saturation de réseau (bande passante) [13].

Changement de la topologie du réseau:

L'état du réseau peut changer pendant l'exécution d'un service donné en raison de défaillances des dispositifs réseau, rupture de liens ou d'autre raisons peut causer un changement de l'état de réseau, ce qui peut entrainer une diminution en bande passante de ce que les VMs ont besoin. Ce qui oblige une migration des VMs pour répondre à leur besoins [13].

Considération de la robustesse:

Pour des raisons de sécurité, les services situer sur des serveurs différents est isolé. Cependant, si plusieurs VMs est attribuées sur une même machine physique, et si certaines ressources sur d'autres machines physiques sont diffusées, les VMs sur la même machine physique devraient être migré vert autre machine physiques [13].

L'économie d'énergie:

Dans le cas ou le nombre de machines virtuelles en cours sur une machine physique est très petit, ces VMs peuvent être déplacées vers d'autres machines physiques via le processus de migration, et cette machine physique peut être mise en veille ou éteinte pour économiser de l'énergie. Cela peut réduire considérablement les dépenses d'exploitation des Datacenters des fournisseurs de services Cloud [13].

Conclusion:

Dans ce chapitre nous avons exposé le concept du Cloud Computing et de la virtualisation, ainsi que le concept de la migration de VM et en mettant en relief l'utilisation, les avantages, et les raisons d'effectuation. Cela nous a conduits à déterminer notre besoin pour mettre en place une plateforme de virtualisation et du Cloud. Dans le chapitre suivant nous allons présenter la technologie Software Defined Networks.

Chapitre 02: Software Defined Networks

2.1-Introduction:

La communication à travers le réseau est devenue indispensable dans tout type d'organisation ou infrastructure. Actuellement, avec la croissance accélérée des périphériques et tous dispositifs numériques connectés à Internet, et la croissance des technologies de communication, gérer l'infrastructure réseau devient plus complexe.

Dans la section suivante nous allons entamer la manière de la technologie Software Defined Networks (SDN).

2.2-Network Operating System:

Beaucoup d'équipements sont utilisés dans les réseaux tels que les commutateurs, routeurs, etc. Il faut savoir que ces derniers ont une partie logicielle appelée Network Operating System (NOS), qui leur permet de remplir correctement les taches qu'ils devraient réaliser.

C'est une plate-forme logicielle fonctionnant en client/serveur, afin de faciliter la programmation des dispositifs.

2.3-Architecture des NOS:

2.3.1-Plan de contrôle (Contrôle plane):

Qui définit comment un équipement avance le trafic, prennent la responsabilité de configuration, la gestion, et l'échange des informations des tables de routage.

2.3.2-Plan de données(Data plane/ forwarding plane):

La partie des commutateurs et routeurs qui assurent effectivement le mouvement des données, au prochain saut sur le chemin vers la destination finale, selon ce qui a été spécifié par le plan de contrôle.

2.4-Software Defined Networks [5]:

Avec la technologie SDN, on établit une séparation claire entre le plan de contrôle et le plan de données, le plan de contrôle est placé dans un contrôleur centralisé à pour but pratique de rendre programmables les réseaux par le biais de ce dernier, qui a une visibilité sur l'ensemble du réseau, y compris les hôtes qui s'y connectent et a une vision complète de la topologie du réseau. Par conséquence, la technologie SDN permet aux administrateurs de réseau, de mettre en œuvre des politiques de trafic uniques et flexibles dont les seules limitations sont liées à la capacité du logiciel faisant fonctionner le contrôleur, au moyen d'application SDN.

Les périphériques réseau prennent leurs propres décisions en interne sur la meilleure façon d'aiguiller le trafic. Ces décisions s'appuient sur les informations distribuées collectées par d-es protocoles. L'approche SDN simplifie l'administration de ces périphériques. En effet ces derniers n'auront plus besoin de traiter des différentes normes de protocoles, mais d'accepter les instructions de programmation des contrôleurs SDN qui sont sous son autorité, en mettant à disposition des administrateurs des interfaces de programmation d'applications (API), en forment une communication entre contrôleurs-périphérique.

2.5- L'architecture SDN et la place d'Openflow:

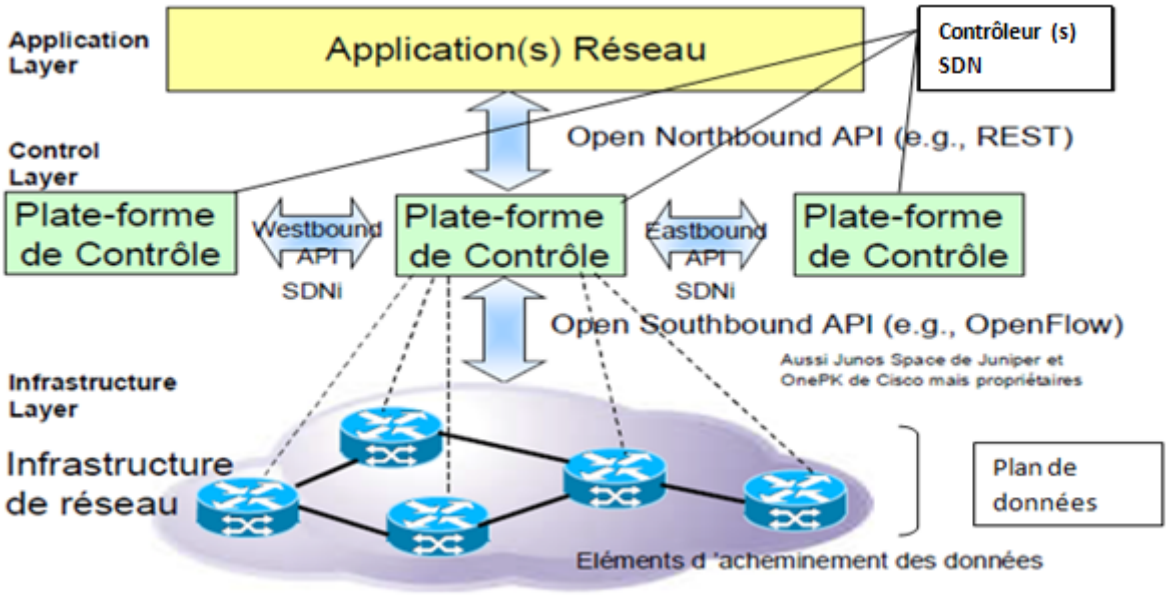


Fig.2.6 - Vue simplifiée de l'architecture SDN [6].

L'architecture SDN et la place d'Openflow se présente comme suit:

La couche infrastructure	La couche de contrôle	La couche applicative
Contient les éléments de réseau responsable de l'acheminement du trafic et qui supportent des protocoles qu'ils partagent avec le contrôleur.	Basée sur le contrôleur SDN qui offre une visibilité globale du réseau et des équipements d'infrastructure.	Apporte l'automatisation des applications au travers du réseau à l'aide d'interfaces programmables ouvertes.

Tableau 4- L'architecture SDN et la place d'Openflow.

La communication entre les équipements et le contrôleur SDN central se fait via un protocole, cette propriété n'est possible que si les périphérique réseaux sont compatibles avec ce protocole, qui permettre:
-D'établir une négociation par des messages d'information de ces commutateurs afin que le contrôleur puisse disposer d'une vue globale logique du réseau physique.
-Et mettre en œuvre des instructions de gestion et de configuration du réseau qu'ils reçoivent de ce contrôleurs, et qui permettent de programmer leur plan de données (des tables de routage des switchs par exemple). Pour ce faire, l'Open Networking Foundation (ONF) a publié OpenFlow.

Tant que, la première interface de communication standart définie entre les couches de contrôle et de transfert de données au dessue de TCP d'une architecture SDN est c'elle d'OPENFLOW, donc son rôle est fondamental dans la conception de l'architecture SDN.

2.5.1-Structure d'un commutateur et fonctionnement d'OpenFlow:

Les équipement réseaux OPENFLOW implémentent une manière de traitement des paquets [7].

Un flux représente un ensemble de paquets avec des valeurs servant de critère et un ensemble d'actions à exécuter. Tous les paquets d'un flux reçoivent les mêmes politiques du dispositif de transmission.

Les principales fonctions des commutateurs sont les suivantes :

- **Fonction de support du contrôle**: Interagit avec la couche contrôle SDN afin de supporter la programmabilité via des interfaces.

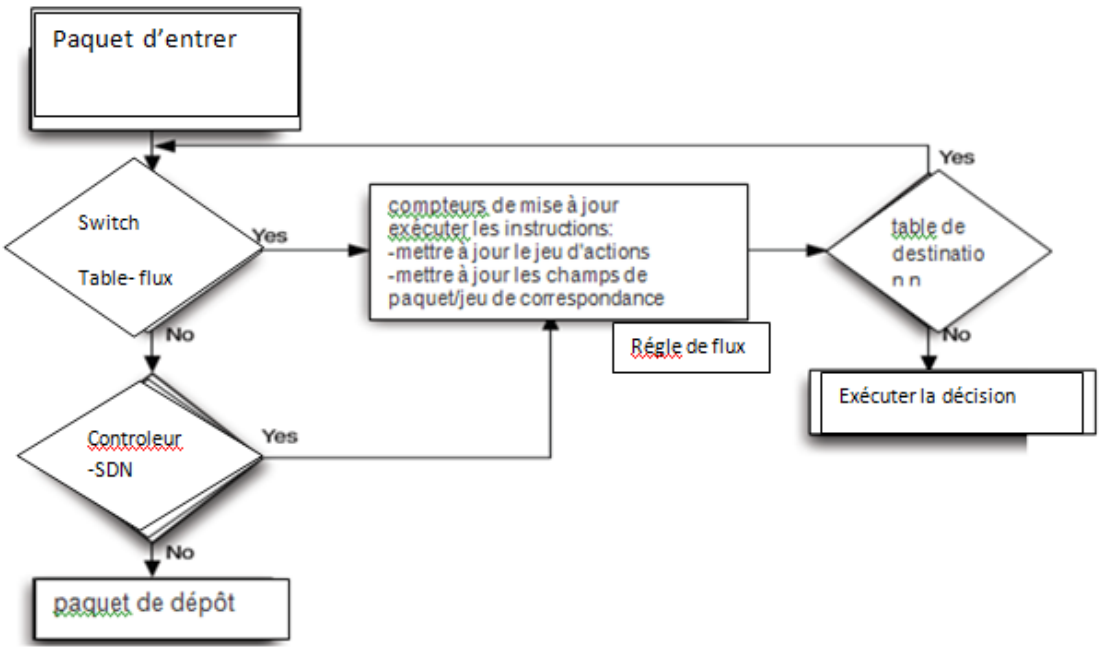


Fig.2.7- Traitement d'un paquet dans un commutateur OpenFlow [32].

Le commutateur communique avec le contrôleur et le contrôleur gère le commutateur avec le protocole OpenFlow.

▪**Fonction d'acheminement des données**: Accepte les flux de données entrants provenant d'autres équipements de réseau et des systèmes de terminaison et les relaie sur un chemin de commutation qui a été calculé et établi à partir des règles définies par les applications SDN.

Sachons que les commutateurs OPENFLOW et les routeurs les plus modernes contiennent dans la couche de donnée (data path) des tables de flux «flow table» avec des règles qui sont utilisées pour effectuer des fonctions de transfert, indiquées dans les entêtes de paquets.

- Lorsque le switch reçoit un paquet il respecte la règle qui lui est associé dans la flow table.

-Si le paquet est nouveau pour le switch et qu'aucune règle ne correspond, le switch envoi le paquet (l'en-tête) au contrôleur qui prendra une décision. Il peut «droper» le paquet ou ajouter une entré dans la flow table pour que le switch puisse traiter les futurs paquets similaires.

▪ **Open vSwitch (OVS) :**

Open vSwitch est une implémentation logicielle d'un switch ethernet, est prévu pour fonctionner comme un commutateur dans les environnements de machines virtuelles. En plus de proposer une visibilité et un contrôle standard des interfaces à la couche réseau virtuelle, il a été conçu pour supporter une installation distribuées sur plusieurs serveurs physiques. Open vSwitch supporte plusieurs technologies de virtualisation basées sur linux.

▪ **Open vSwitch Database Management Protocole (OVSDB):**

Est un protocole de configuration d'OpenFlow établissent des relations entre contrôleurs et commutateurs. Une mise en œuvre Open vSwitch se compose d'un serveur de base de donnée-es du commutateur, des gestionnaires et des contrôleurs qui exploitent le protocole OVSDB p-our fournir des informations de configuration au ce serveur.

▪ **Virtualisation de réseau:**

Un administrateur peut définir ses propres topologies virtuelles, suivant ses propres objectifs, puis d'appliquer des politiques sur ces topologies abstraites. Et cela sera la responsabilité de l'hyperviseur d'installer une configuration physique qui est sémantiquement équivalente à la politique virtuelle.

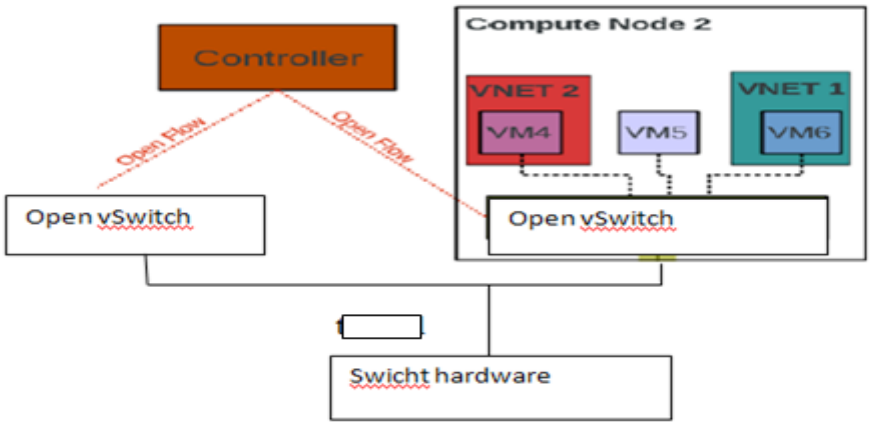


Fig.2.8- Architecture Open vSwitch.

2.6- Controleurs SDN: Parmi l`ensemble des contrôleurs SDN existants : NOX, POX, OpenDaylight(ODL), et **O**pen **N**etwork **O**perating **S**ystem (ONOS).

NOX :

Le premier contrôleur SDN, développé initialement par Nicira Neworks (basé sur les messages OPENFLOW). Il fournit aux administrateurs réseau une interface programmable avec le réseau, grâce à laquelle ils seront en mesure de développer des applications de contrôle et de gestion du réseau.

Il y a plusieurs contrôleurs Open Source sur le marché mais seuls deux sont actifs aujourd'hui : ODL et ONOS.

ODL:

Est le contrôleur open source de la Fondation Linux. Il supporte OpenFlow et d'autres API southbound; ce qui pourrait rendre plus facile l'intégration avec des réseaux multifournisseurs moderne et hétérogène.
Il fournit à ses clients un haut niveau d'abstraction dans le développement de configuration, et comprend des fonctionnalités critiques telles qu' une haute disponibilité, la modularité, le passage à l'échelle.

ONOS [14] [15]:

ONOS est un contrôleur SDN open source, tend à se concentrer autour des besoins des fournisseurs de services, il est conçu d'ajouter de nouvelle fonctionnalités (modularité), et de démarrer ou arrêter diverses autres (configurabilité), afin de fournir à ses utilisateurs le passage à l'échelle et la haute disponibilité.
Parmi l'ensemble des protocoles qui prend en charge, OpenFlow et OVSDB.

2.6.1-Architecture d'ONOS [21]:

- **Northbound Abstractions.**

-Vue globale de réseau.

-intent Framework.

- **Distributed Core.**
- **Southbound Abstractions.**

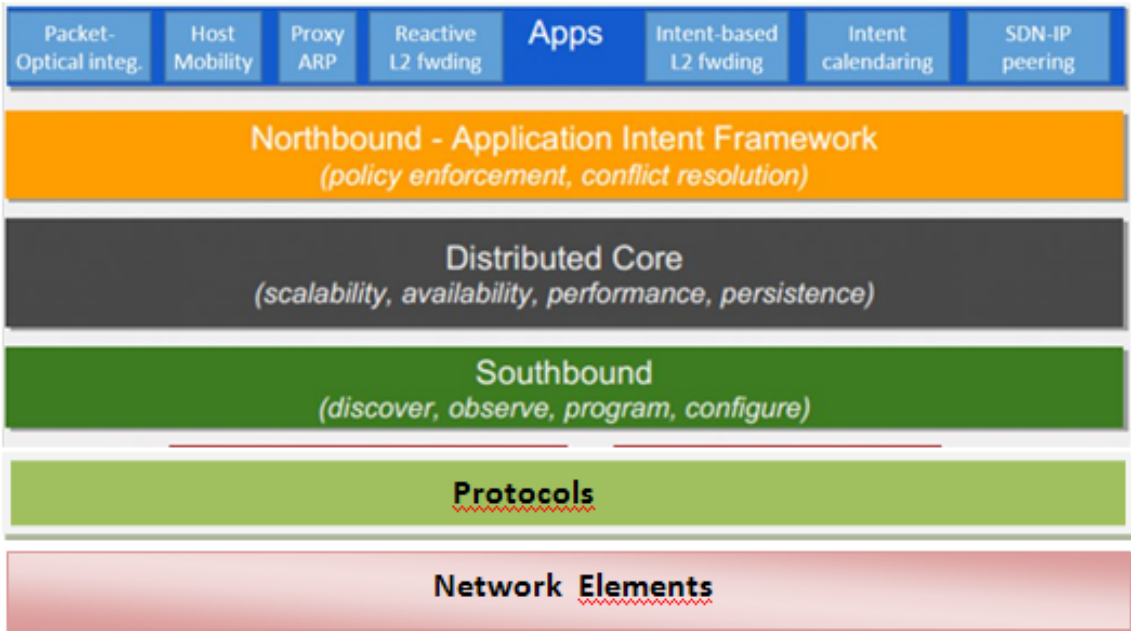


Fig.2.9- Architecture d'ONOS.

▪ **Northbound Abstractions:**

-Vue globale de réseau:

Une application peut interagir avec cette vue globale du réseau via les APIs (par exemple: c

réer une application simple pour calculer les plus courts chemins entre deux périphériques du réseau).

-Intent Framework:

Permet aux administrateurs de programmer le réseau avec un haut niveau d'abstraction et selon leurs exigences au moyen d'intentions qu'il soumettent au contrôleur. Ce dernier s'occupe de leur traduction grâce au module appelé « intent compilation » en « instalable intents ». Ces actions sont installées au niveau des commutateurs OpenFlow grâce au processus appelé «intent instalation» ce qui se traduit par l'installation de règles de flux sur un OpenFlowswitch particulier.

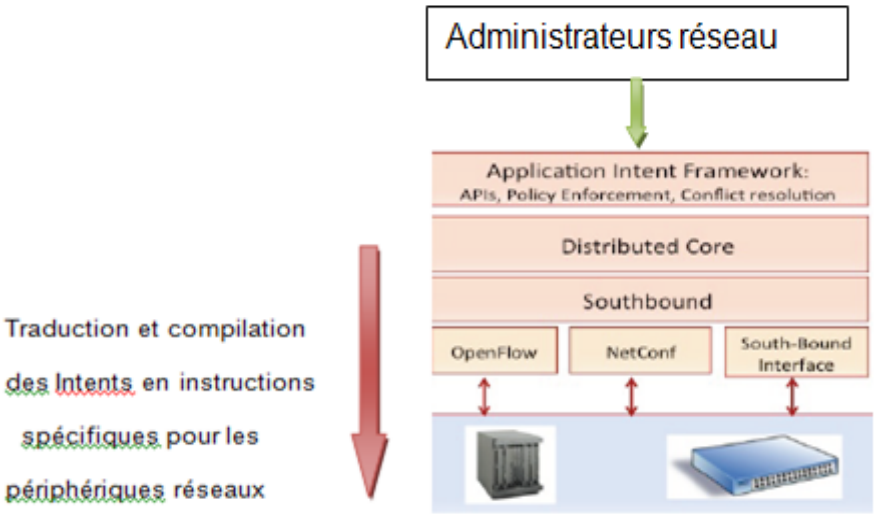


Fig.2.10- Intent Framework [21].

▪ **Distributed Core:**

▪Le déploiement d'ONOS se fait sous forme d'instance logicielles identique, leur rôle est gérer un certain nombre de périphériques du réseau telle que:

- Si un openFlow switch est géré par plus d'une instance une seule est considéré comme instance maitre.

▪Le but de communication des instances ONOS par l'échange de messages entre elle, est le suivant:

-Fournir une vue globale du réseau aux applications.

-Éviter les conflits de gestion dans le cas ou un OpenFlow switch a plus d'une instance pouvant potentiellement le manager.

Tous ses caractéristiques permettent:

-De déployer et sans interruption de service, de nouvelles instances dans le but d'augmenter les capacités de notre plan de contrôle et ce, pour répondre à une extension du réseau.

-Et si une instance maitresse tombe en panne, les OpenFlowswitchs qu'elle manageait se verront être gérés par une autre, sans avoir à recrées et à resynchroniser leurs tables de flux, telle que cette panne ou autre défaillance au niveau du plan de contrôle, sera transparent pour les applications.

-Fournir la possibilité de mettre à jour des instances sans interruption de service.

▪ **Southbound Abstractions:** fournies par ONOS consistent en un format générique pour la représentation des éléments constituant le réseau.

Conclusion:

Dans ce chapitre, nous avons abordé quelque limitation des réseaux traditionnels ainsi que la grande difficulté quant à leur administration et gestion. Nous avons ensuite détaillé comment l'introduction de la nouvelle technologie SDN a remédié à ces limitations. Ceci à été possible grâce à la séparation du plan de contrôle et de données. Dans la même partie nous avons aussi exposé en détail l'architecture de SDN et le protocole OpenFlow.

En suite, nous avons également présenté quelques contrôleurs open source les plus utilisés, ainsi que l'architecture d'ONOS. Dans le chapitre suivant nous allons aborder une architecture physique par simulation, et le processus de migration du flux entre VM.

Chapitre 03:

Conception de système et implémentation

3.1- Introduction:

Afin de mettre en œuvre dans la sous section suivante une migration de VM exécutant un service, entre deux infrastructures Cloud différentes, nous allons mis en place l'architecture c-omposée d'acteurs. Cette architecture consiste à la migration d'un flux lu par un abonné. Nous allons présenter le scénario et les étapes d'implémentation de la migration [23].

3.2- Conception de l'architecture physique du système:

La mise en place de la qualité de service nécessite en premier lieu la reconnaissance des di-fférents services. Celle-ci peut-se faire sur la base de nombreux critères [22].[30]:

· La source et la destination du paquet.

· Les ports source et de destination.

· La congestion des réseaux.

· La validité du routage (gestion des pannes dans un routage en cas de routes multiples par ex-emple).

· La bande passante consommée.

· Les temps de latence.

En fonction de ces critères et autres, différentes stratégies peuvent ensuite être appliquées po-ur assurer une bonne qualité de service.

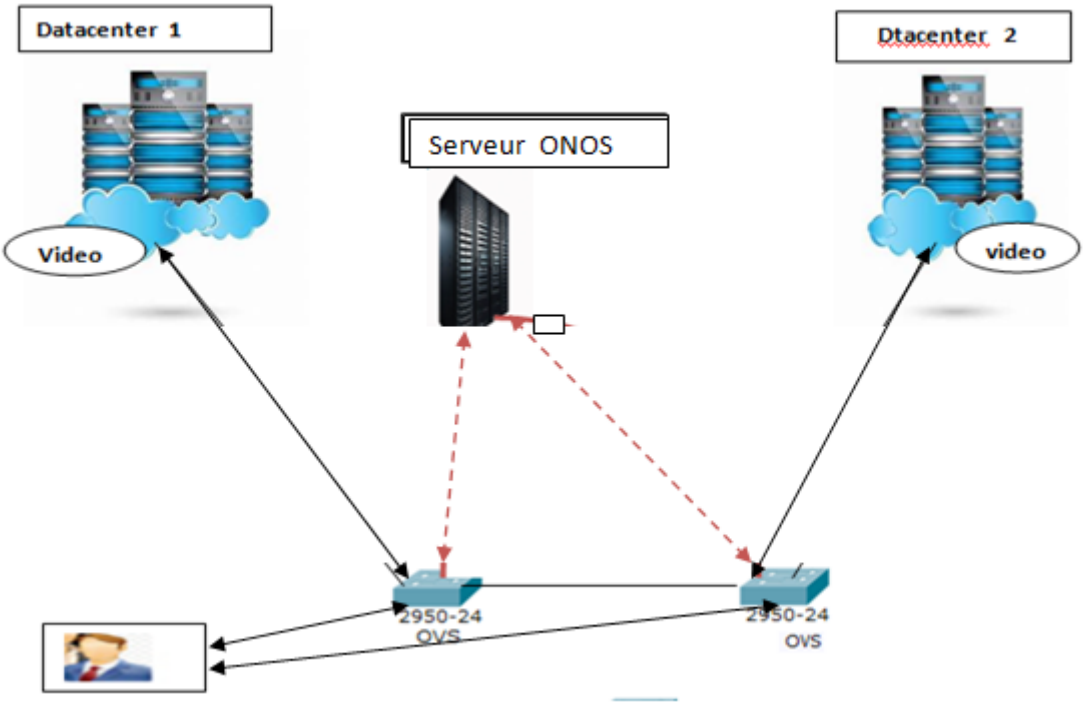


Fig.3.11- Architecture physique du système.

-L'infrastructure Datacenter 1 peut être utilisée pour traiter et stocker de grandes quanti- tés de données d'un niveau de sécurité très élevé pour des services tel que le Cloud, qui centralise l'ensemble des logiciels et des données dans des centrales numériques multi-clients en les rendant accessibles en ligne. Le Datacenter 1 il devra allouer les ressources nécessaires à la VM (CPU, RAM, espace de stockage, etc), qui devront exécuter ce qui a été demandé.

- Nous aurons besoin d'une infrastructure Cloud (Datacenter 2) ayant les ressources néces- saires à la restauration de la VM avec le service qu'elle exécutait avant la migration.

- Les sw open flow:

Avant la migration, le contrôleur SDN devra s'assurer de la connectivité entre l'utilisateur et la VM qui est en train d'exécuter le service qu'il a demandé.

Après la migration et avant que la VM ne démarre au niveau de l'infrastructure Cloud des- tination, la connectivité entre l'utilisateur et la VM devra maintenir, et assurer que l'échange d'information entre ces derniers se fera avec la même qualité de service définie initialement.

Le consommateur devra être en mesure de lire ses données avant et après la migration, avec un minimum downtime.

3.3- La Description de scénario:

-Lorsque le client demande un service auprès du fournisseur de services Cloud, ce dernier lui déploie dans son data center une VM avec les ressources virtuelles nécessaires. Cette dernière exécutera le service demandé avec la Qos nécessaire.

- Cette VM sera gérée par l'hyperviseur de Datacenter 1.

- Le contrôleur SDN établi la connectivité client/service, de la source à la destination, en précisant l'ensemble des commutateurs de chemin.

- Pour différentes raisons de mobilité, manque de ressources matérielles, etc, la migration de ces VMs ers une autre infrastructure Cloud doit être en mesure.

-La migration n'est possible que si le fournisseur de services Cloud destination a les ressources suffisantes pour relancer ce même service après la migration.

-Une reconfiguration du réseau est nécessaire de sorte à rétablir la connectivité client et le même service héberger dans le Datacenter 2.

Ceci étant dans le but de fournir la haute disponibilité à ces utilisateurs, ainsi qu'une bonne qualité de service.

3.4- Comparaison et choix du contrôleur:

Parmi l'ensemble des contrôleurs cités précédemment (NOX, Ryu, ODL et ONOS), dans le but de choisir le contrôleur le plus adapté à la mise en œuvre de notre travail, beaucoup d'entre eux ont été comparés dans plusieurs articles et revues. Nous nous sommes basés sur la comparaison réalisée par les auteurs qui affirme qu'ONOS est plus performant en matière de débit et de latence dans l'environnement utilisé. Après avoir exécuté l'ensemble de cas de test, ONOS a surpassé ODL dans presque tous les tests exécutés, révélant que pour un réseau à faible performance ONOS a montré de meilleurs résultats [26].

Dans ce qui suit, ONOS est utilisé pour informer le protocole OPENFlow de la configuration de dispositifs réseau. Le protocole OVSDB transmettra cette configuration aux périphérique réseau concernés, et ce dans le but de définir la bande passante du bon port d'après OVS.

- **Mininet:**

Mininet [33] est un émulateur réseau qui crée des hôtes, des liens virtuels, des pilotes et des commutateurs capable de fonctionner avec le protocole OpenFlow, et d'expérimenter la technologie SDN.

Les réseaux Mininet exécutent des applications de code et de réseau Unix/Linux et utilisent le noyau Linux pour émuler les éléments réseau.

Mininet utilise la virtualisation basée sur les processus pour exécuter de nombreux périphériques réseau dans un noyau. Cette fonctionnalité fournit des processus individuels avec des interfaces réseau séparées. Mininet vous permet de créer un commutateur OpenFlow, des pilotes pour gérer ces commutateurs et ces hôtes pour la communication sur le réseau simulé. Les connexions entre le commutateur et les hôtes sont effectuées en utilisant l'Ethernet virtuel "veth".

Un administrateur réseau a accès à ces périphériques à l'aide de l'interface CLI, entre autres, pour configurer la couche de données.

Les services de base d'ONOS, ainsi que ses applications, sont écrits en Java chargé dans le conteneur OSGi Karaf, qui est une structure de données utilisé pour stocker des objets sous une forme organisée qui suit des règles d'accès spécifiques. C'est un processus ou un ensemble de processus isolés du reste du système [27], les application et services s'exécutent rapidement en parallèles.

▪ L'administrateur est en mesure de gérer et de configurer le réseau d'une manière plus automatisée en utilisant le REST API, dans le sens ou il spécifie par exemple qu'après une certaine durée de temps, il faut soumettre tel ou tel ensemble de règles de gestion, comme la migration de flux lue par un client.

▪ Notez que SDN est toujours présent au stade du développement, même si certains fabricants fournissent déjà la migration en plus de la multitude d'alternatives libres qui existent actuellement.

▪ La possibilité de travaux futurs de projet ONOS au stade du développement:

ONOS est une plateforme de gestion SDN relativement récente et son utilisation pratique est long. il y a de nombreuses caractéristiques à étudier :

• l'utilisation d'un analyseur de protocole pour observer le trafic entre le commutateur et pilote.

• Créer des applications personnalisées en les saisissant dans le dossier "apps" de ONOS pour une gestion réseau plus personnalisée.

• Utiliser ONOS sans Mininet pour créer des appareils virtuels.

3.5- Implémentation de l'architecture et migration du service :

-La mise en œuvre de l'architecture présentée dans la figure 12 par simulation:

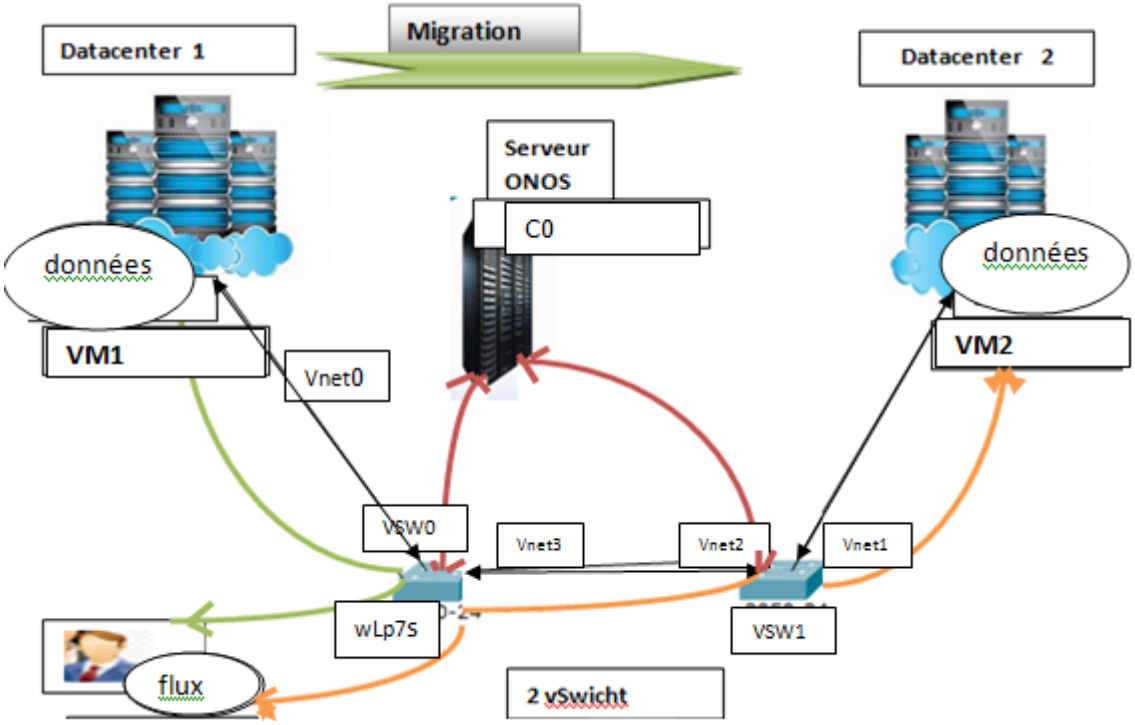


Fig.3.12- Architecture physique du système- migration du flux.

- Implémentation logicielle des switchs Ethernet :

-installation d'Open vSwitch : $ sudo apt-get install openvswitch-switch
-creation des switchs: $ ovs-vsctl add-br vsw0
$ ovs-vsctl add-br vsw1
-On vérifie que notre bridge existe :

```
# ip link show vsw0
```

On ajoute dans le fichier /etc/network/interfaces la configuration ci-dessous pour rendre notre bridge persistant.

```
Auto vsw0
allow-ovs vsw0
iface vmbr0 inet manual
ovs_type OVSBridge
ovs_ports none
```

-Les mêmes étapes de configuration pour le switch 2 (vsw1);

-Patch entre deux bridges

- Pour chacun des deux bridges à liée, il faut exécuter l'ensemble des commandes suivantes:

```
$ ovs-vsctl add-port
vsw1 vnet2
$ ovs-vsctl add-port vsw0 vnet3
$ ovs-vsctl set interface vnet2 type=patch
$ ovs-vsctl set interface vnet2 options:peer=vnet3
```

- Pour rendre cette configuration persistante, il faut modifier le fichier /etc/network/interfaces:

```
auto vsw0
allow-ovs vsw0
iface vsw0 inet manual
ovs_type OVSBridge
ovs_ports patch-br0
auto vsw1
allow-ovs vsw1
```

- iface vsw1 inet manual
- ovs_type OVSBridge
- ovs_ports patch-br1
-
- allow-vsw0 patch-br1
- iface patch-br1 inet manual
- ovs_bridge vsw0
- ovs_type OVSPatchPort
- ovs_patch_peer patch-br0
-
-
- allow-vsw1 patch-br0
- iface patch-br0 inet manual
- ovs_bridge vsw1
- ovs_type OVSPatchPort
- ovs_patch_peer patch-br1

-Affectation de l'interface «wLp7s0» à vsw0: $ ovs-vsctl add-port vsw0 wLp7s0

- Pour voir la création: $ ovs-vsctl show
-Configuration VM1: $ ovs-vsctl add-port vsw0 vnet0
$ virsh edit vm1
-Interface réseau pour la VM1: $ ifconfig vnet0
- Pour voir la création: $ ovs-vsctl show
-Afficher la «ForwardingTable » dans OVS: $ovs-appctl fdb/show vsw0

-Configuration VM1: $ ovs-vsctl add-port vsw1 vnet1

-Configuration VM2: $ virsh edit vm2

-Interface réseau pour la VM2: $ ifconfig vnet1
- Pour voir la création: $ ovs-vsctl show

-Afficher la «ForwardingTable » dans OVS: $ ovs-appctl fdb/show vsw1

-Installation d' ONOS:
Disponible sur le site officiel;
La version: onos-1.15.0.tar.gz;

Les commandes d'installation d'ONOS:

$ Sudo passwd root
$ Su
#apt-get update
#java -version
unzip -v
#apt-get install maven git openjdk-8-jre openjdk-8-jdk unzip
nano /etc/netplan/50-netcfg.yaml

wget http://repo1.maven.org/maven2/org/onosproject/onos-releases/1.15.0/onos
tar xzf onos-1.15.0.tar.gz
ls
cd onos-1.15.0
cd bind
ls
export JAVA-HOME = /usr/lib/jvm/java-1.8.0-openjdk-amd64

-ONOS Ecosystem:
POUR L'EXÉCUTION DES COMMANDES QU'OFFRE ONOS, NOUS DEVONS UTILISÉE LES COMMANDES SUI-VANTES, POUR LANCÉES L'INTERFACE DE LIGNE DE COMMANDE (CLI) D'ONOS:

cd onos-1.15.0
cd bin
./onos-sevice

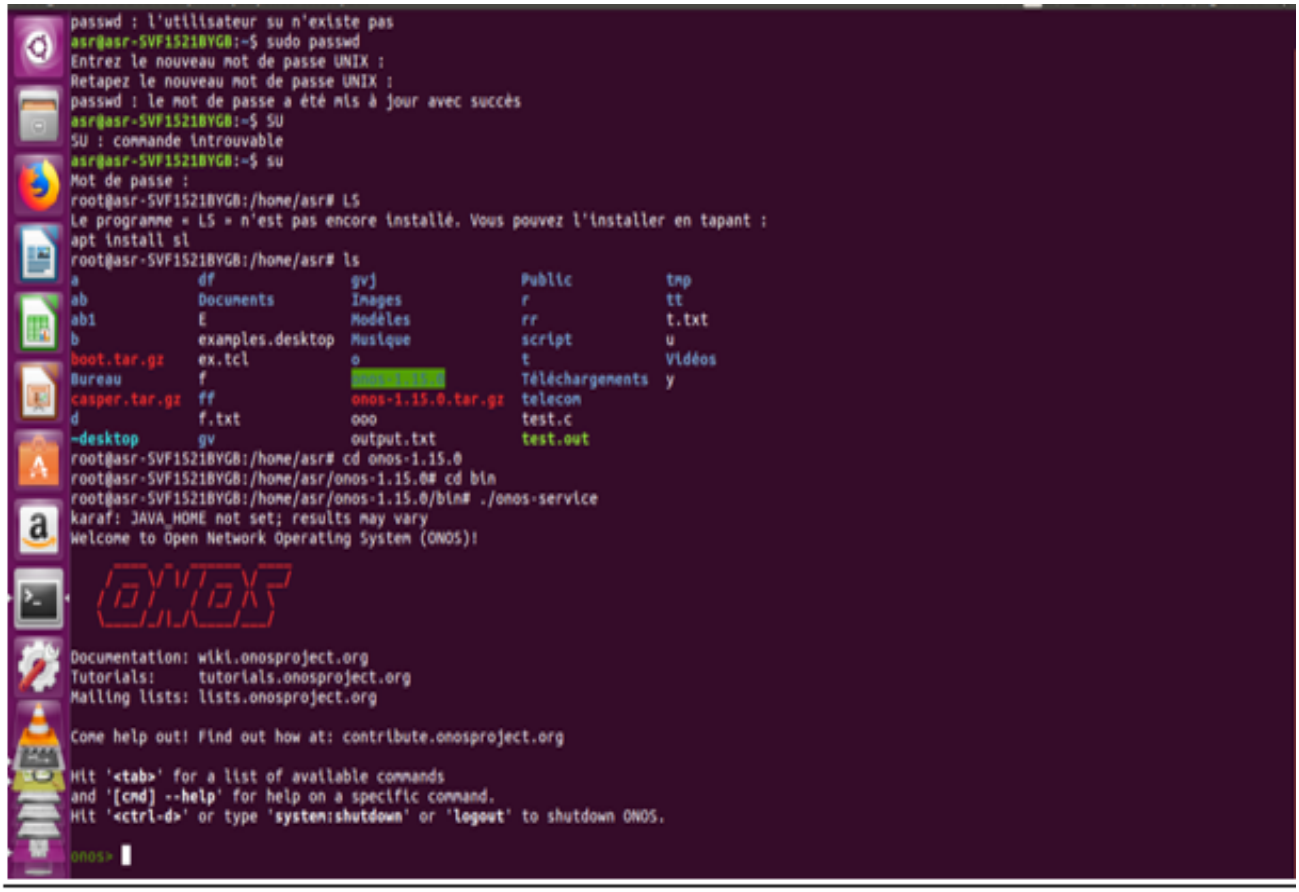


Fig.3.13- ONOS CLI.

Pour accéder à interface graphique utilisateur (GUI), interface Web qu'offre ce contrôleur:

URL: http://192.168.1.3:8181/onos/ui/index.html
User: onos
Password: rocks

Cette dernière fournit aux administrateurs un résumé de la topologie de réseau, et des détails sur chaque périphérique, et sur des applications**.**

-Désactive openvswitch-controller si installé ou en cours d'exécution:

sudo service openvswitch-controller stop
sudo update-rc.d openvswitch-controller disable

-Mise en réseau avec Mininet:

Il existe deux façons de configurer les topologies sous l'émulateur Mninet, soit à travers :

- les lignes de commande CLI
- un code python

```
sudo mn --topo linear,2,2 –-switch ovs  controller=remote,ip=192.168.1.3
```

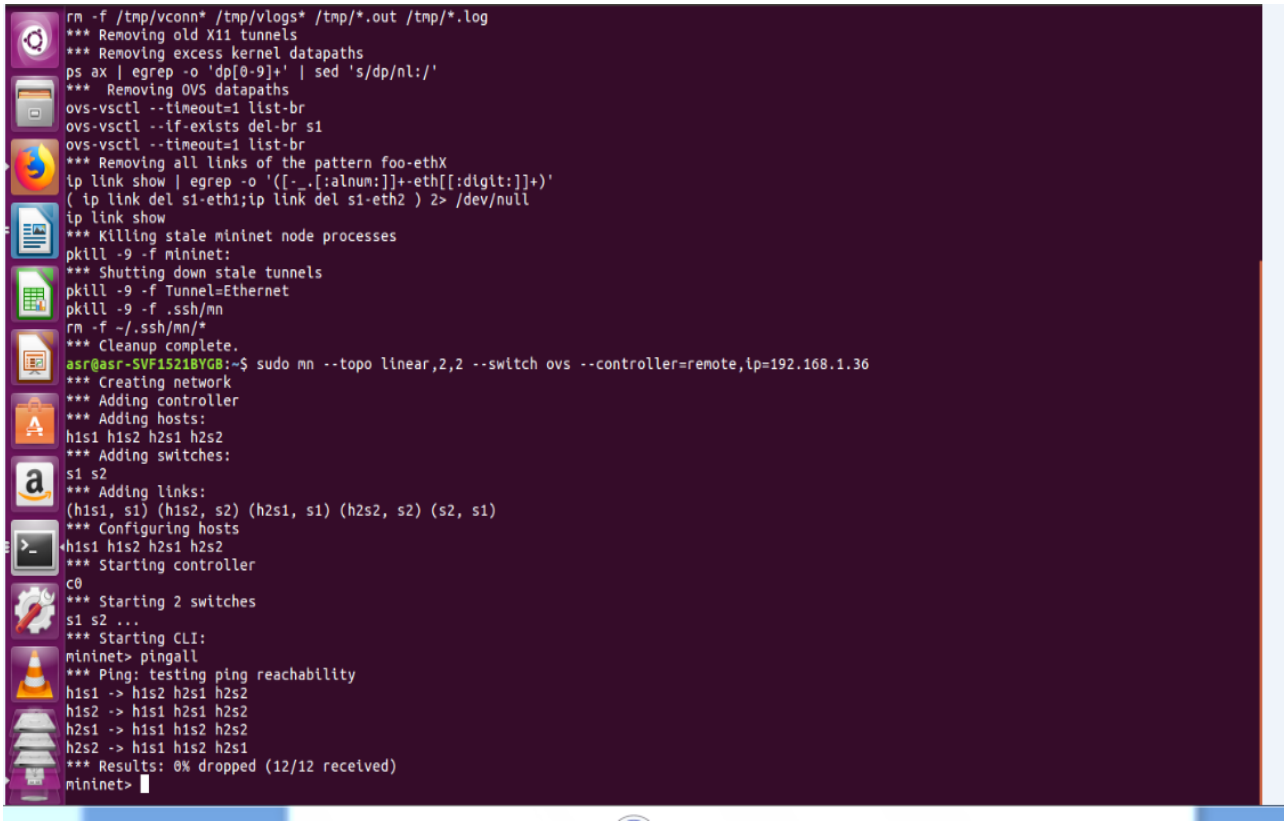


Fig.3.14- Mise en réseau avec Mininet.

-Au sein de la CLI, nous pouvons utiliser l'API Mininet faite avec Python pour exécuter plusieurs actions:

```
#!/usr/bin/python
from mininet.net import Mininet
from mininet.node import Controller, OVSSwitch, RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel
def myTopo():
net = Mininet( controller= RemoteController, switch=OVSSwitch, build=true)
c1 = net.addController( 'c0', ip='192.168.1.36' )
s1 = net.addSwitch( 's1' )
s2 = net.addSwitch( 's2' )
h1=net.addHost ('h1')
h2=net.addHost ('h2')
h1=net.addHost ('h3')
h2=net.addHost ('h2')
net.addLink(s1,s2)
net.addLink(s1,h1)
net.addLink(s1,h2)
net.addLink(s2,h1)
net.addLink(s2,h2)
net.build()
c0.start()
s1.start( [ c0 ] )
s2.start( [ c0 ] )
CLI( net )
net.stop()
if  name == 'main':
setLogLevel('info' )
myTopo ()
```

Les topologies personnalisées sont stockées dans le répertoire personnalisé, qui se trouve dans le dossier Mininet.
Pour exécuter ces topologies, vous devez exécuter les instructions suivantes:

```
root@controller: sudo mn --custom \
~/mininet/custom/ topology.py --topo mytopo
```

Ce programme est un script de python qui commence par la déclaration des bibliothèques utilisées dans le programme, puis déclare une fonction 'main' appelée myTopo qui sera exé-

cutée à la fin du script. Avec ce code, nous avons créé une topologie composée de deux switches, qui sont attachés à des hôtes. Ce réseau est géré par un contrôleur ‘c0’.

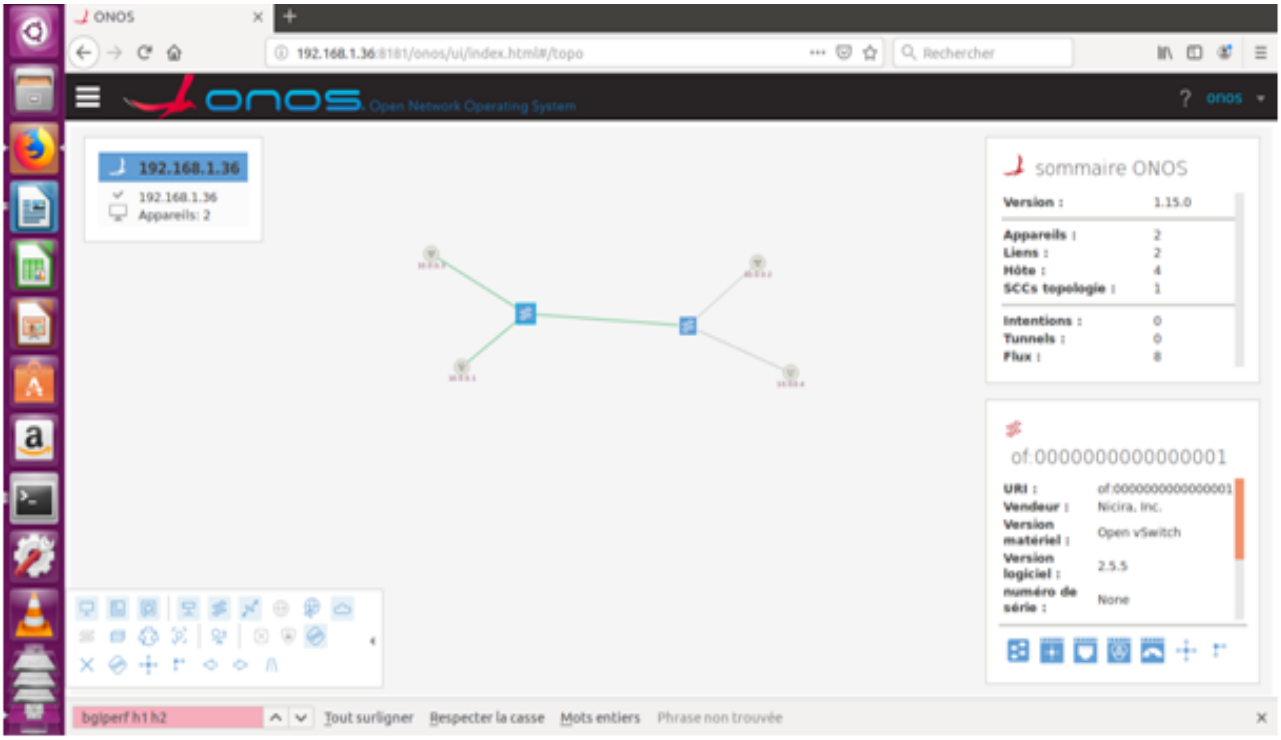


Fig.3.15- la topologie du réseaux-interface GUI.

Enfin, lorsque le démarrage d’une topologie Mininet, chaque commutateur peut être connecté à un contrôleur distant à l'intérieur ou à l'extérieur de l'ordinateur. Dans ce cas les deux vswitch openflow sont connectés au contrôleur interne distant d’ip : 192.168.1.3 .
Un pilote intégré par défaut qui agit comme un commutateur avec OpenFlow intégré. Ce commutateur crée une table de flux.

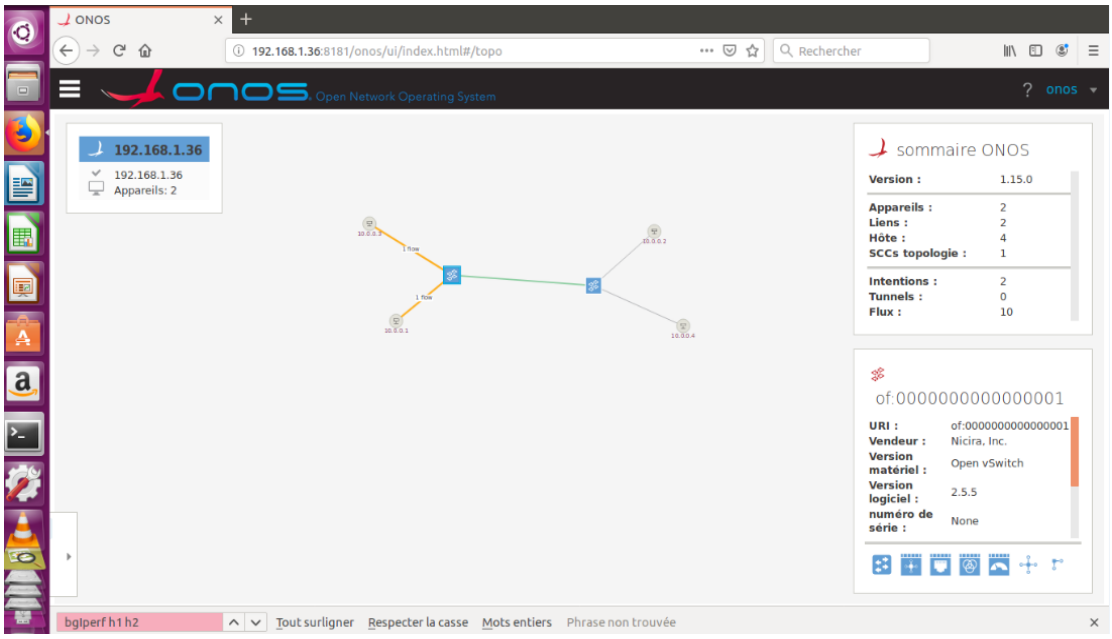
- Le protocole OpenFlow doit être activé sur le pilote ONOS:

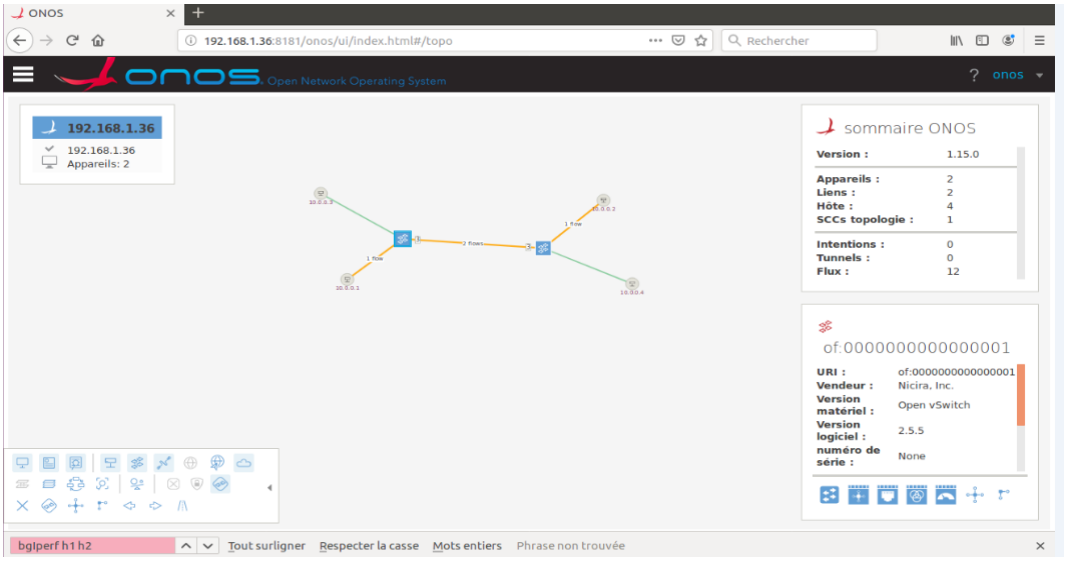
```
onos> app activate org.onosproject.openflow
```

Lorsqu'un hôte souhaite communiquer avec un autre hôte, le commutateur envoie ce paquet au contrôleur et le contrôleur entre une entrée de flux dans le commutateur qui indique comment router les paquets.
L'étape suivante consiste à effectuer un test de connexion entre les deux hôtes. Dans ce scénario le protocole ICMP est utilisé.

```
mininet> s1h1 ping s1h2
```

La raison pour laquelle le ping a échoué est qu'aucune règle de flux n'est installée sur le calque les données de commutation, qui achemine le trafic de manière appropriée. ONOS fournit l'application de nom "Reactive Forwarding" qui installe des flux de routage, mais cette application n'est pas activée par défaut.onos> apps activate org.onosproject.fwd





La coloration des liens utilisés avec une couleur orange, indique le chemin de trafic a suivre, et depuis quel Datacenter le flux et lue pour le client.

▪ Etablir la connectivité client-service, en spécifiant à l'ensemble des paquets de données le chemin à suivre, par la mise en œuvre des règles de flux (intents):

-Chemin du client au service (Datacenter1): La commande suivante permet de spécifier au controleur ONOS que pour le vswitch vsw0 (s1), tous trafic entrant par le port wLp7s0(eth1), l'envoyer en sortie par le port vnet0(eth2);

```
onos> add-point-intent s1/1 s1/2
Point to point intent submitted:
PointToPointIntent{id=0x0, key=0x0, appId=DefaultApplicationId{id=2, name=org.onosproject.cli}, priority=100, resources=[], selector=DefaultTra
fficSelector{criteria=[]}, treatment=DefaultTrafficTreatment{immediate=[NOACTION], deferred=[], transition=None, meter=[], cleared=false, StatT
rigger=null, metadata=null}, ingress=FilteredConnectPoint{connectPoint=s1/1, trafficSelector=DefaultTrafficSelector{criteria=[]}}, egress=Filte
redConnectPoint{connectPoint=s1/2, trafficSelector=DefaultTrafficSelector{criteria=[]}}, constraints=[], links=null, resourceGroup=null}
```

```
onos> add-point-intent  s1/1  s1/2
```

Les reste des commandes suivent la même logique;

-Chemin du Datacenter1 au client:

```
onos>add-point-intent  s1/2   s1/1
```

S'il ya un événement qui nécessite la migration de service, l'administrateur réseau décide d'effectuer une migration du flux, telle que le client étant déjà en train de lire le flux à l'instant correspondant, et après il reçoive le contenu lu par le Datacenter 2, et ce par la définition d'une nouvelle connectivité client/service, et en évitant de définir la résiliation de la connectivité client et datacenter1 qui cause une interruption de service, afin de réduire le temps d'indisponibilité.

```
onos>add-point-intent  s1/1  s1/3

onos>add-point-intent  s2/3  s2/1
```

```
onos>add-point-intent  s2/1  s2/3

onos>add-point-intent  s1/3  s1/1
```

La migration du flux est équivalente à la redirection du trafic entre le client et le service de l'ancienne vers la nouvelle infrastructure.

▪ Le temps d'interruption de service, qui est la durée de temps nécessaire au contrôleur SDN pour effectuer ces opérations, il représente le temps nécessaire obtenue lors de l'util- isation du CLI d'ONOS pour soumettre de nos intentions afin que la requête soit postée au web server d'ONOS, et l'interaction avec ce dernier pour l'effectuation des deux opérations précédente (résiliation et rétablissement de la connectivité client/Datacenter 2), telle que dans ce cas le downtime est proportionnelle seulement au temps de rétablissement de la nouvelle connectivité. Ce temps est mesurable par des fonctionnalités fournies par ONOS dans la section suivante.

Conclusion:

Dans ce chapitre, nous avons abordé quelque étapes de conception d'une architecture rése- au, ainsi que le scénario d'interaction des acteurs composent le système déployée pour la réa- lisation de travail qui porte sur la migration de flux d'un fournisseur de services Cloud à un autre. Par la suite nous avons présentais l'implémentation de l'architecture, avec l'explicati- on de fonctionnement de script qui se charge de l'établissement d'une connectivité client serv- ice.

Conclusion générale:

Avec l’avènement des innovations technologique récentes telles que la virtualisation des machines et des systèmes, le cloud computing ou encore l’Internet des objets, et l'arrivée de nouveaux protocoles, d'équipements, et des technologies utilisées, manifeste par deslimites de capacité de manipulation, la bande passante, et des limites actuelles des architectures réseaux deviennent de plus en plus problématiques pour les opérateurs et les administrateurs réseaux.

Le réseaux mobiles 5G est autres technologies sont capable de répondre au mieux aux exigences de ces utilisateurs en terme de performance. Une nouvelle façon d’administrer les réseaux est la technologie SDN, qui a pris un élan considérable dans l’ensemble des technologies comme le déploiement de la 5G, par la simplification de la configuration et la gestion des réseaux actuels et les rendre programmables ou, en d’autres termes, plus simple à tester, à corriger et à faire évoluer.

Donc la technologie SDN permettre l’innovation grâce à leur programmation, afin de fournir aux utilisateurs une bonne Qos, le passage à l’échelle, et la haute disponibilité. Cette dernière permet de garantir la continuité de service fournie aux utilisateurs en réduisant au mieux, ou en suppriment les interruptions de service résultante de différentes raisons, au moyen de techniques très efficaces, parmi ces technique utilisées est la migration de VMs d’une infrastructure à une autre a l’aide de contrôleur SDN ONOS.

Bibliographie:

[1] PMTIC, LE CLOUD Recherche et stockage – Organiser et stocker. PMTIC - LabSET – ULg 2014. http://le-cloud.net/. Consulté: 27/07-30/07/2019.

[2] Licence Creative Com. http://creativecommons.org/licences/by-sa/3.0/deed.fr. consulté: 31/07/2019.

[3] *Source :http://blog.blaisethirard.com.* Consulté: *27/07-31/07/2019.*

[4] https://www.figer.com/email.html. Consulté: 27/07-1/08/2019

[5] https:// Whatis.com/fr. SD-WAN ou la modernisation du réseau d'entreprise. https://www.lemagit.fr/essentialguide/SD-WAN-ou-la-modenisation-du-reseau-dentreprise. Consulté: 27/07-05/08/2019.

[6] Software Defined Network-Principes et Architecture. EFORT. Consulté: 27/07-05/08/2019.

[7] Le Protocole OpenFlow dans l'Architecture SDN (Software Defined Network). EFFORT. http://www.efort.com. Consulté: 27/07-08/08/2019.

[8] http://reseau-informatique.prestataires.com/conseils/virtualisation. Consulté: 27/7 1/08/2019.

[9] Thierry LONGEAU. La Virtualisation des systèmes d'information. Consulté: 27/07-1/08/2019.

[10] Composants de base VMware vSphere. ESXi 5.0. vCenter Serveur 5.0 . http://www.vmware.com/fr/support/pubs. Consulté: 27/07-1/08/2019.

[11] Vincent Kherbache. Ordonnancement des migrations à chaud de machines virtuelles. Consulté: 27/07-12/08/2019.

[12] Huandong wang, Yong Li, Ying Zhang, and Depeng Jin. Virtual machine migration planning in software-defined networks. IEEE Transactions on Cloud Computing. Consulté: 27/07- 10/08.

[13] Bo Hu, Shanzhi Chen, Jianye Chen, and Zhangfeng Hu. A Mobility-Oriented Scheme for Virtual Machine Migration in Cloud Data Center Network. IEEE Access, PP(99):1-1. Consulté: 27/07-2/08/2019.

[14] Introducing ONOS - a SDN network operating system for Service Providers. http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf. Consulté: 15/08-19/08/2019.

[15] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi.Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, Guru Parulkar. Open Networking Laboratory, USA NEC Corporation of America, USA Create-Net, Italy. ONOS: Towards an Open, Distributed SDN OS. Consulté: 15/08-20/08

[16] Felix Wortmann, Kristina Fluchter, et al. The Internet of Things. Business et Information Systems Engineering, Consulté: 20/08-25/08.

[17] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. IEEE Transaction on industrial informatics. Consulté: 20/08/-28/08.

[18] 5G use cases and requirements. https://www.ramonmillan.com/documentos/bibliografia/5GUseCases_Nokia.pdf.Consulté: 20/08/-28/08.

[19] A deliverable by the NGMN alliance. NGMN 5G WHITE PAPER. Next generation mobile networks. https://www.ngmn.org/5g-white-paper/5g-white-paper.html. Consulté: 20/08/-28/08.
[20] Vmware infrastructure automating high availability (ha) service with vmware ha. https://www.vmware.com/pdf/vmware_ha_wp.pdf. Consulté: 20/08/-28/08.
[21] Introducing ONOS - a SDN network operating system for Service Providers. Consulté: 20/08/-28/08.
[22] Brad Peterson Citrix Systems. Inc. Démonstration | Virtualisation : De l'idée à la pratique.**https://www.youtube.com/watch?time_continue=11&v=4pkVfsmk5CE**.Consulté: 1/09/-13/09.
[23] Namame Mehdi.Inter Cloud Virtual machine migration. usthb. MS. Consulté: 05/08-10/08.
[24] Luis Miguel Sciacca Urruchurtu. MAIER, GUIDO ALBERTO. Performance analysis tool for SDN controllers : ONOS versus OpenDayLight comparison. 2015/2016. Consulté: 1/09/-13/09.
[25] Maria George, Deepa V Jose. Analyse comparative des performances des contrôleurs dans les logiciels réseaux définis à l'aide MININET. IJERTV8IS070214. 2019. Consulté: 1/09/-13/09.
[26] Dr.S-E BENBRAHIM. MIGRATIONS EN TEMPS RÉEL DES MACHINES VIRTUELLES INTERDÉPENDANTES. Université montréale.2016. Consulté: 13/09-20/09/2019.
[27] Un conteneur Linux, qu'est-ce que c'est ?. https://www.redhat.com/fr/topics/containers / whats-a-linux-container. 20/09-consilté: 25/09/2019.
[28] M.JENDOUBI. Mise en place d'une plateforme de virtualisation et déploiement d'une solution Cloud privé open source. Uvt.MS.2015. Consulté : 1/09/-13/09.
[29] Tharam Dillon, Chen Wu, Elizabeth Chang. Cloud Computing: Issues and Challenges. International Conference on Advanced Information Networking and Applications. 2010 24th IEEE. Consulté: 27/07-12/08/2019.
[30] David Bernstein, Erik Ludvigson, Krishna Sankar, Steve Diamond, Monique Morrow. Blueprint for the Inter cloud- Protocols and formats for Cloud Computing Interoperability. Consulté: 1/09/-13/09.
[31] Pradip D. Patel, Miren Karamta, M. D. Bhavsar, M. B. Potdar. Live Virtual Machine Migration Techniques in Cloud Computing: A Survey. International Journal of Computer Applications (0975 – 8887) Volume 86 – No 16, January 2014. Consulté: 27/07-12/08/2019.
[32] Luis Miguel Sciacca Urruchurtu and Manuel Alejandro Cadenas Reyes. Performance analysis tool for SDN controllers : ONOS versus OpenDayLight comparison. Consulté: 13/09- 20/09/2019.