UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE

DEPARTEMENT INFORMATIQUE

Mémoire de fin d'étude

En vue d'obtention de diplôme de MASTER PROFISSIONNEL
en ingénierie des systèmes d'information

Thème

Etude et comparaison des failles de sécurité d'OpenStack et OpenNebula

Encadré par :
Mme S.FELLAG

Jury :
Présidente : Mme M.BENTAIB
Examinatrice : Mme F.ACHEMOUKH

Réalisé par :

HAMDANI nadir
KERROUM salima
OUALLOUCHE nacera

Année 2018/2019

Sommaire

Table de figure

Introduction générale

Actuellement Internet connait une évolution sans précédent ce qui entraine également celle des technologies basées sur lui. Parmi ces technologies, on a l'informatique en nuage.

La simplicité de gestion des infrastructures informatiques, la continuité d'activité et la réduction des coûts de mise en place et de maintenance des systèmes d'information ont été souvent les plus grandes priorités des entreprises. Toutefois, les solutions présentes sur le marchésont généralement complexes et coûteuses. En parallèle, les évolutions quotidiennes au niveau des infrastructures réseau ont conduit à l'évolution de nouveaux périphériques informatiques qui rendent la tâche de déploiement et de gestion encore plus difficile à assurer. De toutes ces contraintes est né le besoin du Cloud Computing qui est apparu comme une solution révolutionnaire à un grand nombre de défis auxquels les entreprises doivent faire face.

Le Cloud Computing est un nouveau modèle informatique qui consiste àproposer et à fournir des ressources informatiques sous forme de services à la demande, accessiblesde n'importe où, n'importe quand et par n'importe qui. Ce nouveau conceptpermet à des entreprises d'externaliser le stockage de leurs données et de leur fournirune puissance de calcul supplémentaire pour le traitement de grosse quantité d'information.

Certes, cette technologie offre plusieurs avantages comme un déploiement rapide, un paiement à l'usage, une réduction des coûts, une délivrance de services plus rapide, un accès au réseau omniprésent ; en raison de toutes ces diverses caractéristiques elle est devenue une solution intéressante pour les entreprises.

Cependant, la sécurité des données en transit dans un Cloud reste un challenge et une inquiétude majeure pour les fournisseurs de Cloud. En effet, ces données sont la cible de plusieurs attaques réseau, qui ont pour but d'interrompre, d'intercepter, de modifier et de fabriquer des informations. Par conséquent, il est essentiel de faire face à ces attaques en vue d'améliorer l'utilisation et l'adoption du Cloud. En fait, la protection de la vie privée et la sécurité des données sont primordiales dans l'utilisation des services Cloud. Il n'est donc pas surprenant que la sécurité soit la première préoccupation des entreprises qui désirentpasser au Cloud. Face à ce défi plusieurs solutions sont proposées : des solutions propriétaires et des solutions open source afin de pouvoir déployer des infrastructures de Cloud en toute sécurité.

C’est dans ce cadre que s’inscrit notre projet de fin d’étude consistant à l’élaboration et configuration des infrastructures Cloud open source : « OpenStack » et « OpenNebula » dans le but de pouvoir comparer ces deux solutions par rapport au niveau de sécurité qu’elles offrent aux utilisateurs.

Le présent mémoire est articulé autour de quatrechapitres intitulés :

- Le premier chapitre « **Généralités sur le Cloud Computing »** : ce chapitre est consacré à définir les notions fondamentales du cloud computing ainsi que certaines notions pour la bonne compréhension de ce concept et nous avons étudié les éléments constitutifs du cloud.

- Le deuxième chapitre « **La sécurité dans le Cloud Computing** » : dans ce chapitre nous avons entamé la sécurité dans le Cloud Computing, nous avons présenté les services de sécurité exigés, les différentes attaques et menacespossibles, ainsi que les solutions de sécurité proposées.

- Le troisième chapitre « **Mise en place des solutions OpenStack et OpenNebula** » : ce chapitre consiste à présenter deux solutions open source (OpenStack et OpenNebula) mises sur le marché et étudier leur architecture et leur fonctionnement. Il est également consacré à la mise en place de ces deux solutions et examiner leur aspect sécuritaire.

- Le quatrième chapitre « **Les solutions proposées pour remédier aux failles de sécurité dans OpenStack et OpenNebula» :**dans ce chapitre nous allons proposer des solutions ou des contre-mesures des failles liées à OpenStack et OpenNebula.

Nous finalisons par une conclusion générale dans laquelle nous allons citer nos acquis durant la réalisation de notre projet et nos perspectives.

Chapitre I : Généralités sur le Cloud Computing

Introduction

Actuellement, le Cloud Computing représente une révolution dans le monde informatique.En effet le Cloud Computing est émergé dans les dernières années comme une solution universelleutilisée par différents types d'utilisateurs.

Le cloud computing est une nouvelle technologie qui permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosse quantité d'informations.
Face à cette tendance nous allons présenter dans ce chapitre quelques généralités sur le Cloud Computing, à savoir sa définition, ses caractéristiques, les services qui l'offre, ses éléments constitutifs, ses modèles de déploiement ainsi certaines notions pour la bonne compréhension de ce concept et enfin on termine avec ses avantages et inconvénients.

I. Historique

L'évolution vers le Cloud a connu plusieurs étapes :

- **1960** : John McCarthypionnier de l'intelligence artificielle suggérait lors d'une conférence au MIT (Massachusetts Institute of Technology)que la technologie informatique partagée (« time-sharing ») pouvait construire un bel avenir dans lequel la puissance de calcul et même les applications spécifiques pouvaient être vendues comme un service public. (KARTIT, 2016)
- **1970** : L'avènement des réseaux qui a rendu possible l'exécution déportée des tâches informatiques.
- **1980** : Client/serveur
- **1990** : Internet
- **1997** : Ramnath Chellappa professeur de système d'information présenta un paradigme informatique « le cloud ».
- **1999** : Salesforces fut le premier à transformer ce concept en business avec le logiciel de gestion de la relation client éponyme.
- **2000** : Service Web

- **2002** : Face à un problème rencontré par Amazon : « le surdimensionnement de son parc de serveurs hors période de fêtes », il décida de louer ses serveurs à d'autres entreprises à la demande pour rentabiliser ses machines.
- **2006** : L'apparition d'Amazon EC2 (Elastic Compute Cloud).

C'est en 2009 que la réelle explosion du Cloud survint avec l'arrivée sur le marché de sociétés comme Google (Google App Engine), Microsoft (Microsoft Azure), IBM (IBM Smart Business Service), Sun (Sun Cloud) et Canonical Ltd (Ubuntu Enterprise Cloud).(KARTIT, 2016)

II.Définition du cloud

Le Cloud signifie « nuage » et Computing « informatique », le Cloud Computing est donc l'informatique en nuage ou nuagique ou encore infonuagique (au Québec) pour une traduction littérale anglais français.

Bien que nous avons trouvé de nombreuses définitions disponibles, nous retiendrons cependant la définition la plus utile, la plus complète et la plus populaire provient de l'Institut national des normes et de la technologie (**NIST**)[1]qui définit le Cloud Computing comme étant un modèle informatique permet l'accès via un réseau de télécommunication, à la demande et en libre-service, à des ressources informatiques partagées configurables.

1 NIST : Le National Institute of Standards and Technology, ou NIST, est une agence du département du Commerce des États-Unis. Son but est de promouvoir l'économie en développant des technologies, la métrologie et des standards de concert avec l'industrie.

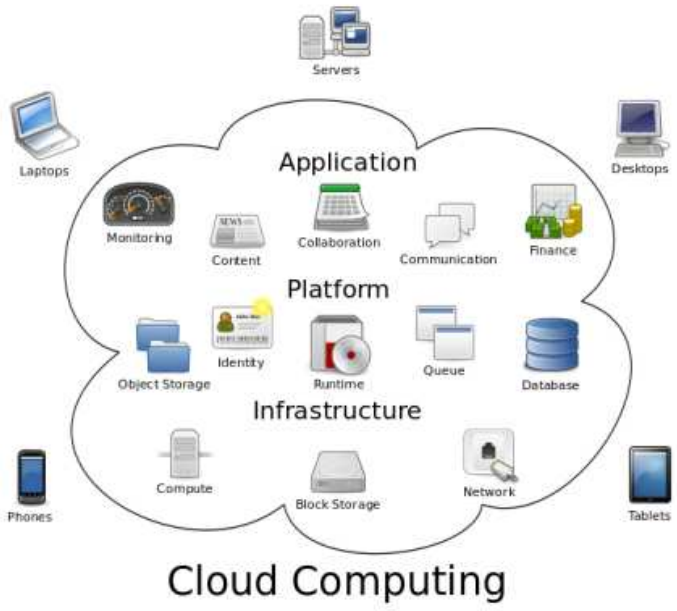

Figure 1 : Le cloud computing*(Foulon, 2016)*

III. Les caractéristiques du cloud computing

Pour mieux comprendre les concepts de base et des technologies dans le Cloud, nous extrayons du document de définition de NIST les cinq caractéristiques suivantes :

- **Service à la demande**

  Les ressources sont disponibles au moment où le client lessouhaite. Ce service est souvent effectué par le fournisseur de façon automatique sans nécessiter d'interaction humaine.

- **Large accès au réseau**

  Les capacités sont disponibles sur le réseau et sont accessibles via des mécanismes standards qui favorisent l'utilisation par des plates-formes clientes minces ou épaisses et hétérogènes (par exemple, les téléphones mobiles, les ordinateurs portables et les assistants personnels numériques ou PDA[2]).

- **Ressources partagées**

  Les ressources matérielles du fournisseur sont partagées entre les différents utilisateurs du service.

- **Elasticité rapide**

2 PDA : Personal Digital Assistant, un PDA est un appareil électronique destiné à un usage personnel

En fonction de la demande, les ressources et les capacités peuvent être rapidement et automatiquement déployées et mises à l'échelle à n'importe quelle quantité et à tout moment.

- **Service mesuré**

Toutes les ressources allouées peuvent être surveillées et contrôlées afin de mesurer leurs consommations avec un niveau d'abstraction approprié selon le type du service
(Par exemple stockage, traitement, bande passante et comptes d'utilisateurs actifs).

- **Paiement à l'usage (pay-as-you-use)**

Le coût est proportionnel à l'usage, donc l'utilisateur paye pour exactement ce qu'il utilise.

IV. Les éléments constitutifs du Cloud Computing

Les éléments pouvant constituer le système Cloud sont les suivants :

IV.1. La virtualisation

Dans cette section nous commencerons pardéfinir le concept de virtualisation. Ensuite nous allons présenter les différents objectifs, les différentes techniques et les différents types de la virtualisation.

IV.1.1. Définition

La virtualisation recouvre l'ensemble des techniques matérielles et/ou logiciels qui permettent d'exécuter plusieurs machines virtuelles dites invitées (guest) sur une seule machine dite hôte (host). La virtualisation constitue le socle du Cloud Computing, car elle offre la possibilité de mettre en commun des ressources informatiques à partir de clustersde serveurs, et ainsi d'affecter ou réaffecter dynamiquement des machines virtuelles aux applications à la demande.
Pour bénéficier de cette technologie, il suffit d'équiper une machine d'un logiciel de virtualisation permettant d'ajouter une couche de virtualisation, appelée hyperviseur.(GERVAISE, 2012)

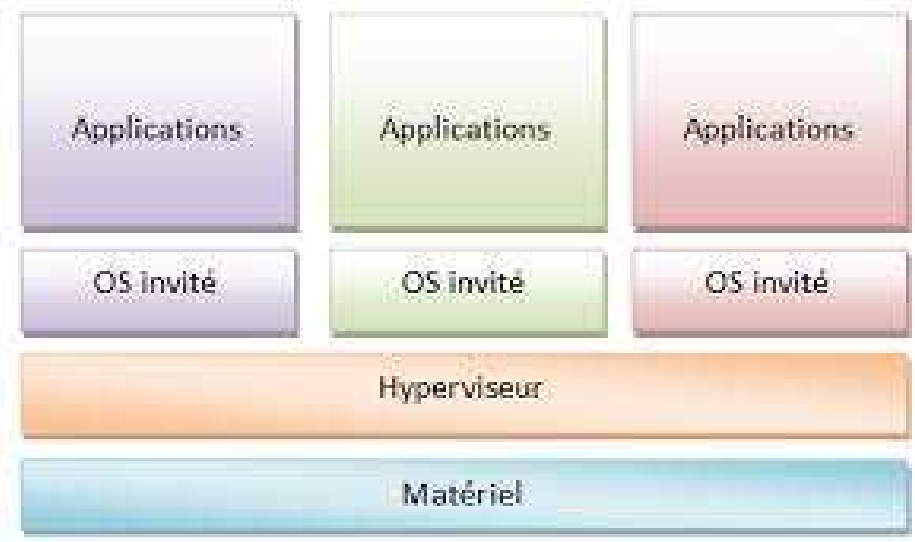

Figure 2: Les différentes couches d'un serveur virtualisé*(David GELIBERT, 2012)*

IV.1.2. Les objectifs de la virtualisation

- Rationaliser les évolutions en besoins matériels, en empilant plusieurs serveurs sur une machine physique, tout en augmentant la disponibilité de l'ensemble.
- Réduction des coûts de l'infrastructure physique.
- Augmentation de la flexibilité et de l'efficacité opérationnelle.
- Disponibilité accrue des applications et amélioration de la continuité d'activité.
- Amélioration de la gestion et de la sécurité des postes de travail

IV.1.3. Les hyperviseurs

Un hyperviseur, également appelé gestionnaire de machine virtuelle, est un programme qui permet à plusieurs systèmes d'exploitation (Windows, Linux, etc..) de partager un seul hôte matériel.

On compte deux types d'hyperviseurs : Le type 1 dit **natif** et le type 2 dit **logiciel**

- **Le type 1 (natif)**

Un hyperviseur de type 1 a comme particularité de s'installer directement sur la couche matérielle (à comprendre qu'il est relié directement au matériel de la machine hôte).

Il est alors considéré comme outil de contrôle du système d'exploitation, c'est à dire qu'il s'agit d'un noyau allégé et optimisé pour la virtualisation de machines.

Au démarrage de la machine physique, l'hyperviseur prend directement le contrôle du matériel, et alloue l'intégralité des ressources aux machines hébergées.
Le gros avantage de ce type d'hyperviseur, c'est qu'il permet d'allouer la quasi-totalité des ressources disponibles aux machines virtuelles.

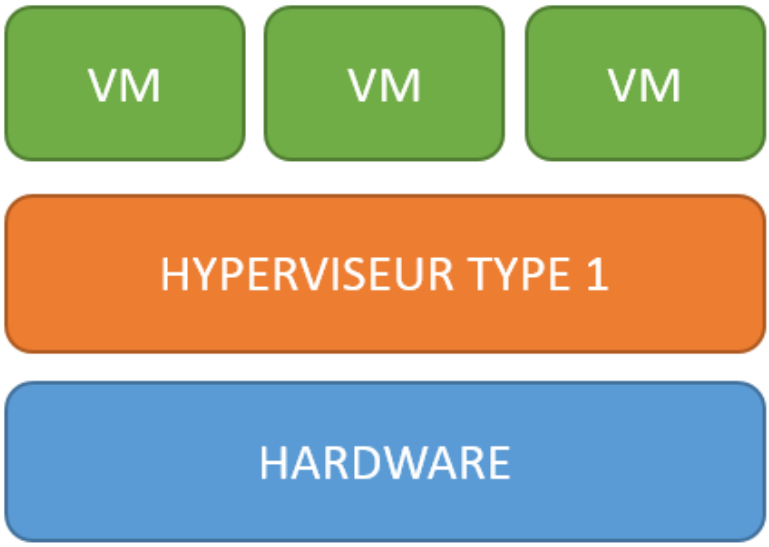

Figure 3 : Hyperviseur type 1*(PETER, 2016)*

Exemples des hyperviseurs de type 1 :

- ✓ ESXI Server
- ✓ Oracle VM
- ✓ Hyper-V
- ✓ Xen…

- **Le type 2 (logiciel)**

Un hyperviseur de type 2 est considéré comme un logiciel, s'installant et s'exécutant sur un système d'exploitation déjà présent sur la machine physique.

Le système d'exploitation virtualisé par un hyperviseur de type 2 s'exécutera dans un troisième niveau au-dessus du matériel, celui-ci étant émulé par l'hyperviseur.

L'avantage d'utiliser ce type d'hyperviseur est la possibilité d'installer et d'exécuter autant d'hyperviseurs que l'on désire sur notre système hôte, ce type n'étant pas relié directement au matériel.

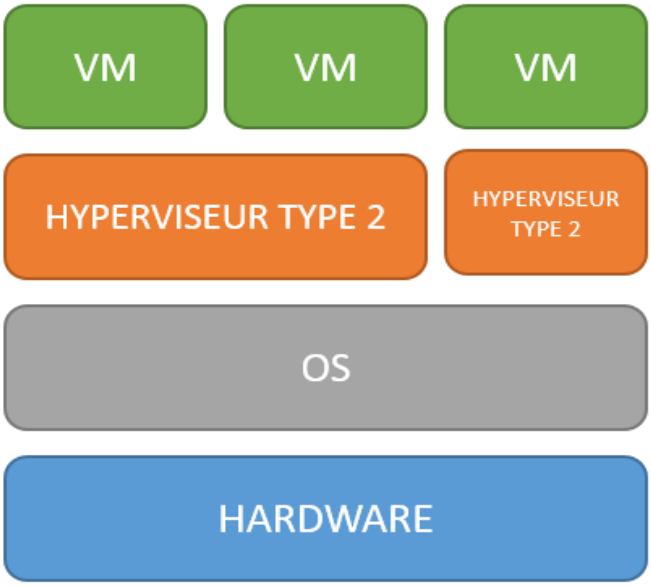

Figure 4 : Hyperviseur de type 2*(PETER, 2016)*

Exemples des hyperviseurs de type 2 :

VMware Server (anciennement connu sous le nom **gsx**)
VMware Workstation, VMware Fusion, l'hyperviseur open source
Microsoft Hyper-V, Virtual PC et Virtual Server
VirtualBox d'Oracle

IV.1.4. La machine virtuelle (VM)

La machine virtuelle est un logiciel ou un système d'exploitation qui affiche le comportement d'un ordinateur physique, mais est également capable d'effectuer des taches telles que l'exécution d'application et de programme.

Propriétés clés des machines virtuelles :

Les caractéristiques des VMs détaillées ci-dessous offrent plusieurs avantages.

Partitionnement :

- Exécutez plusieurs systèmes d'exploitation sur une machine physique.
- Répartissez les ressources système entre les machines virtuelles.

Isolation :

- Assurez l'isolation des pannes et la protection de la sécurité au niveau matériel.
- Maintenez les performances en déployant des contrôles avancés des ressources.

Encapsulation :

- Enregistrez dans des fichiers l'état complet des différentes machines virtuelles
- Déplacez et copiez des machines virtuelles aussi facilement que des fichiers

Indépendance vis-à-vis du matériel :

- Provisionnez ou migrez n'importe quelle machine virtuelle vers n'importe quel serveur physique

IV.1.5. Les techniques de virtualisation

Il existe plusieurs catégories de virtualisation, utilisant chacune des technologies différentes. Les technologies les plus répandues sont :

- Le cloisonnement
- La virtualisation complète
- La paravirtualisation
- La virtualisation assistée par le matériel

IV.1.5.1. Le cloisonnement / L'isolation :

L'isolation est une technique qui intervient au sein d'un même système d'exploitation.

Elle permet de séparer un système en plusieurs contextes ou environnements. Chacun d'entre eux est géré par l'OS hôte comme un processeur isolé dans un conteneur partageant le même noyau, mais les programmes de chaque contexte ne sont capables de communiquer qu'avec les processus et les ressources associées à leur propre environnement.

Avec l'isolation, l'espace noyau n'est pas différencié, il est unique, partagé entre les différents contextes. Mais on définit de multiples espaces utilisateurs cloisonnés.

Il est ainsi possible de partitionner un serveur en plusieurs dizaines de contextes.

Le partage du même noyau limite cette technique aux environnements de même type.

Le projet le plus connu basé sur l'isolation est **OpenVZ**.

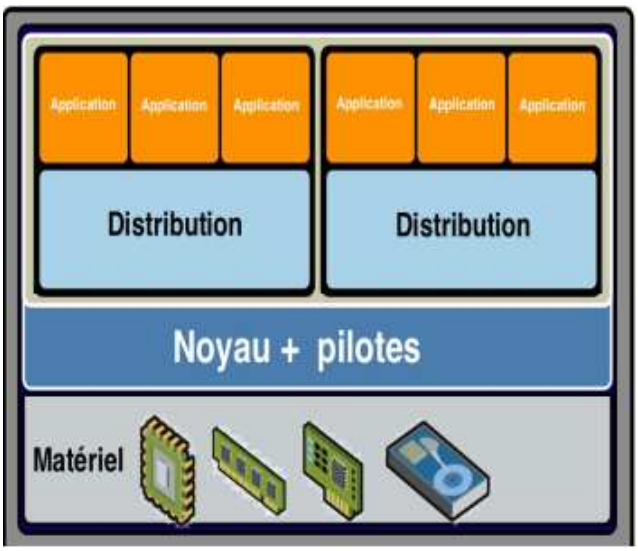

Figure 5 : Virtualisation d'isolation*(Seddiki, 2015)*

IV.1.5.2. La virtualisation complète :

La virtualisation complète permet de faire fonctionner n'importe quel système d'exploitation en tant qu'invité dans une machine virtuelle. Pour l'utilisateur final, ce type de virtualisation est plus simple à mettre en place et plus pratique (le système invité croit s'exécuter sur une véritable machine physique).

L'hyperviseur constitue la couche d'abstraction entre les OS invités et le matériel.

L'hyperviseur va permettre l'exécution de plusieurs machines virtuelles sur la machine physique. Il gère les accès mémoire, l'allocation du CPU et toutes les ressources nécessaires aux machines virtuelles.

L'hyperviseur gère l'ensemble des requêtes des machines virtuelles ce qui permet aux machines virtuelles de fonctionner sans aucune modification de leur noyau. Autrement dit, les machines virtuelles ne savent pas qu'elles s'exécutent de manière virtuelle.

Les solutions les plus connues de la virtualisation complète est : **VMware** et **VirtualBox**

La figure ci-dessous illustre la création de deux machines virtuelles sur une seule machine physique :

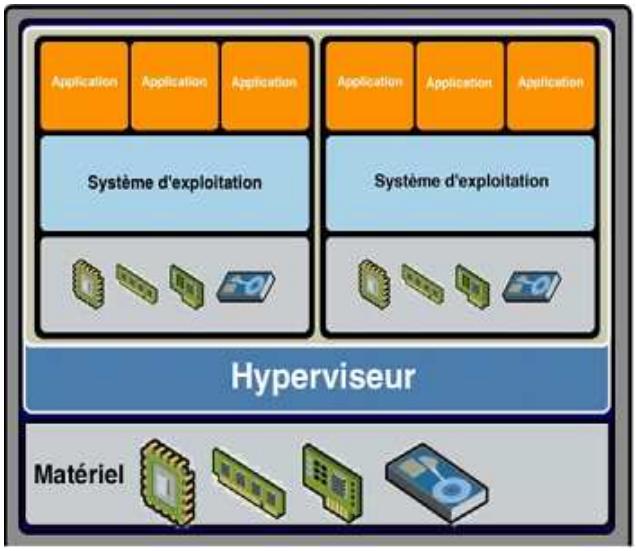

Figure 6 : La virtualisation complète*(Seddiki, 2015)*

IV.5.1.3. La paravirtualisation :

La paravirtualisation a des fonctionnalités différentes dans chaque technique de virtualisation, elle permet une coopération entre l'hyperviseur et le système d'exploitation invité. En effet, lors de l'exécution du système d'exploitation invité, l'hyperviseur capture les appels système de l'invité et les transmet au matériel.(Seddiki, 2015).

L'hyperviseur gère l'interface qui va permettre à plusieurs systèmes d'exploitation invités d'accéder de manière concurrente aux ressources.

Le système d'exploitation invité est « conscient » de l'exécution sur une VM.

Cette opération nécessite certaines modifications logicielles non seulement au niveau du système d'exploitation hôte mais également au niveau du système d'exploitation invité. Ce dernier doit être muni des pilotes permettant d'adresser des commandes au matériel.(Seddiki, 2015).

Exemples :

Xen : de Citrix

KVM : projet hyperviseur intégré dans le noyau linux

ESX/ESXi : hyperviseur leader deVMWare

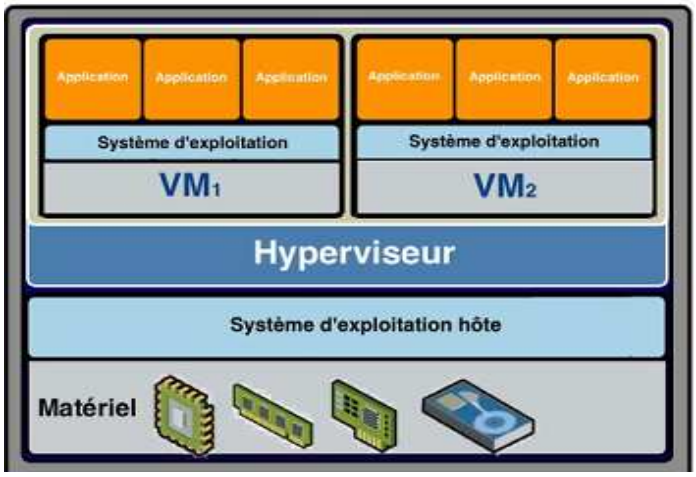

Figure 7 :La paravirtualisation*(Seddiki, 2015)*

IV.1.5.4. Virtualisation assistée par le matériel

La virtualisation assistée par le matériel est en réalité une extension du principe de virtualisation complète.

La principale modification qui est apportée est l'implémentation des instructions dans les processeurs : AMD-V chez AMD et Intel-VT chez Intel.

Ces instructions permettent aux VMs de gérer leurs propres interruptions et donc changement de contexte.

Elles permettent aux matériels de gérer directement les zones de mémoires vives disponibles au VM.

La virtualisation matérielle doit toutefois bien souvent être activée dans le BIOS/UEFI de l'ordinateur et paramétrée convenablement dans les paramètres de la machine virtuelle.

Exemples :Xen server, KVM, ESX/ESXi

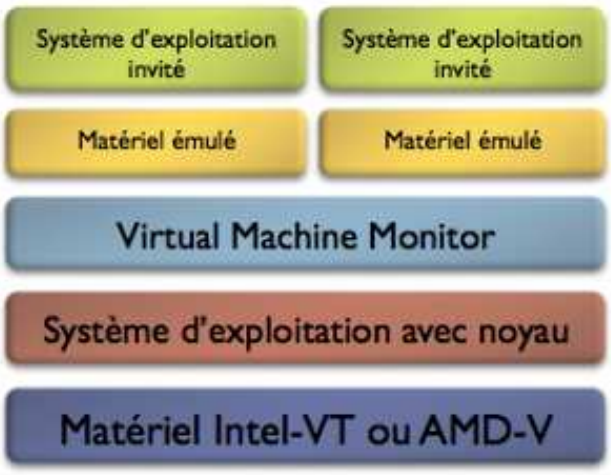

Figure 8 : Virtualisation assistée par le matériel*(Benkemoun, 2009)*

IV.1.6. Les types de virtualisation

En se basant sur les principes de virtualisation décrits plus-tôt, nous distinguons quatre (4) différents types de virtualisation :

IV.1.6.1. La virtualisation d'applications

La virtualisation d'application est une technologie logicielle qui va permettre d'améliorer la portabilité et la compatibilité des applications en les isolant du système d'exploitation sur lequel elles sont exécutées. Elle consiste à encapsuler l'application et son contexte d'exécution système dans un environnement cloisonné.

La virtualisation d'application va nécessiter l'ajout d'une couche logicielle supplémentaire entre un programme donné et le système d'exploitation, son but est d'intercepter toutes les opérations d'accès ou de modification de fichiers ou de la base de registre afin de les rediriger de manière totalement transparente vers une localisation virtuelle.(Ben Mahmoud, 2015)

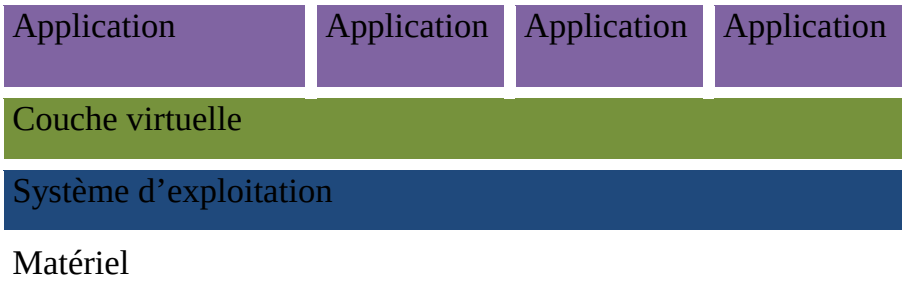

Figure 9 :Virtualisation d'applications*(Ben Mahmoud, 2015)*

Exemple : Wine

IV.1.6.2. La virtualisation du réseau

La virtualisation du réseau consiste à combiner des ressources réseau matérielles et logicielles dans une seule unité administrative. L'objectif de la virtualisation du réseau est de fournir aux systèmes et utilisateurs un partage efficace, contrôlé et sécurisé des ressources réseau.
Les composants de base de la virtualisation du réseau sont :

- Carte réseau virtuelle(VNIC)
- Switch(VSwitch)

IV.1.6.3. La virtualisation de stockage

La virtualisation des stockages permet d'exploiter au maximum les ressources, d'exploiter au mieux le stockage des disques durs.

En effet, il existe trois principales approches de stockage :

1. DAS (Direct Attached Storage)
2. NAS (Network Attached Storage)
3. SAN (Storage Area Network)

1. DAS (Direct Attached Storage) :

Le DAS est un système de stockage directement connecté à un ordinateur non accessible à d'autres ordinateurs qui utilise les technologies**SAS** (Serial Attached Storage),**SATA** (Serial Advanced Technology Attachment)et **SCSI** (Small Computer System Interface).

En matière de stockage, les services informatiques se sont toujours tournés traditionnellement vers des supports physiques:

- Disques durs autonomes
- Piles de disques RAID[3] (disques arrays) (voir (RELAZA, 2016))
- Bandes et supports optiques

2. NAS (Network Attached Storage) :

Un NAS (Serveur de stockage en réseau) désigne un périphérique de stockage (généralement un ou plusieurs disques durs) relié à un réseau par un protocole de communication tel que TCP/IP[4] par exemple.

Les serveurs NAS sont également capables de partager une instance de données entre plusieurs serveurs d'applications, offrant ainsi des capacités de collaboration entre plateformes.

Le stockage NAS ne peut envoyer que des fichiers, et non des blocs de données et cet envoi se fait à l'aide d'un protocole de transfert de fichiers :

- NFS[5] (Network File SYSTEM)

[3] **RAID** : est un ensemble de techniques de virtualisation du stockage permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit les performances, soit la sécurité ou la tolérance aux pannes de l'ensemble du ou des systèmes.

[4]**TCP/IP** : est l'ensemble des protocoles utilisés pour le transfert des données sur Internet.

- CIFS[6] (Common Internet File System)
- FTP[7] (File Transfer Protocol)
- …etc.

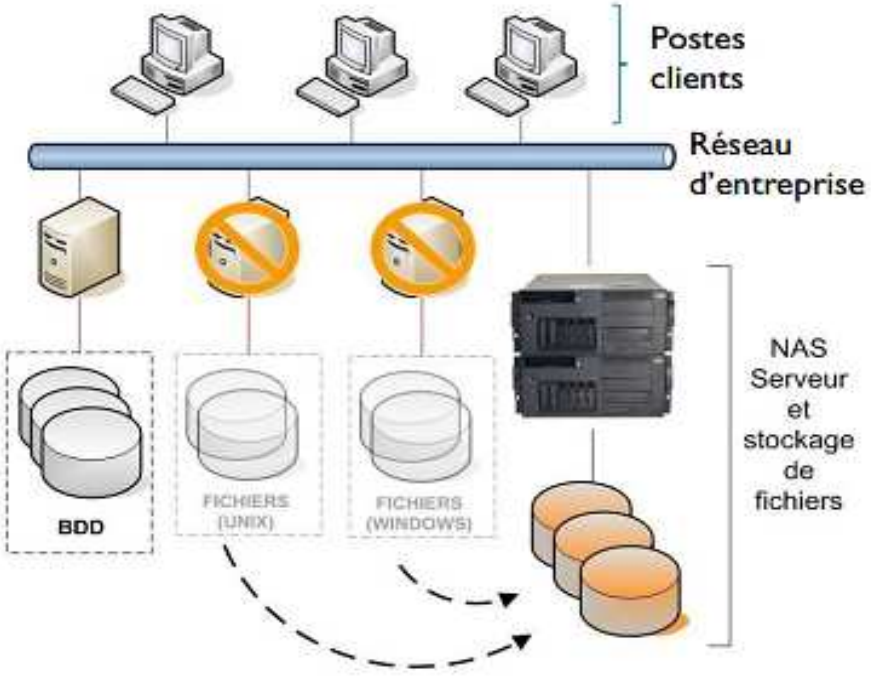

Figure 10 :Approche de stockage NAS*(Marlise, 2010)*

3. SAN (Storage Area Network):

SAN est un réseau physique en fibre optique, dont le but est de permettre la mise en relation de serveurs avec des baies de disques[8].Les données stockées sont routées et hiérarchisées via des commutateurs[9].

SAN est composé de :

- Serveurs
- Baies de disque(Storage Array)

[5]**NFS** : est un protocole de système de fichiers distribué ,permettant à un utilisateur sur un ordinateur client d'accéder aux fichiers sur un réseau informatique, un peu comme le stockage local

[6]**CIFS** : est un protocole permettant le partage de ressources (fichiers et imprimantes) sur des réseaux locaux avec des PC sous Windows.

[7] **FTP** :est un protocole de communication destiné au partage de fichiers sur un réseau TCP/IP

[8]**Baies de disque** : est un équipement composé d'un ensemble de disques regroupé (standard ou dense), un ou plusieurs contrôleurs composés de ports de liaisons avec les serveurs d'application, d'un bus (InfiniBand, Rapid IO, PCI ou PCI Express) ou d'une matrice de commutation (Switch ou Crossbar switché) interne d'échange, de mémoire cache, de CPU de traitement et d'une suite logicielle de gérance des composants et des fonctionnels embarqués

[9]**Commutateur** : Un commutateur réseau (en anglais switch), est un équipement qui relie plusieurs segments (câbles ou fibres) dans un réseau informatique et de télécommunication et qui permet de créer des circuits virtuels

- Eléments réseaux (switchs …)

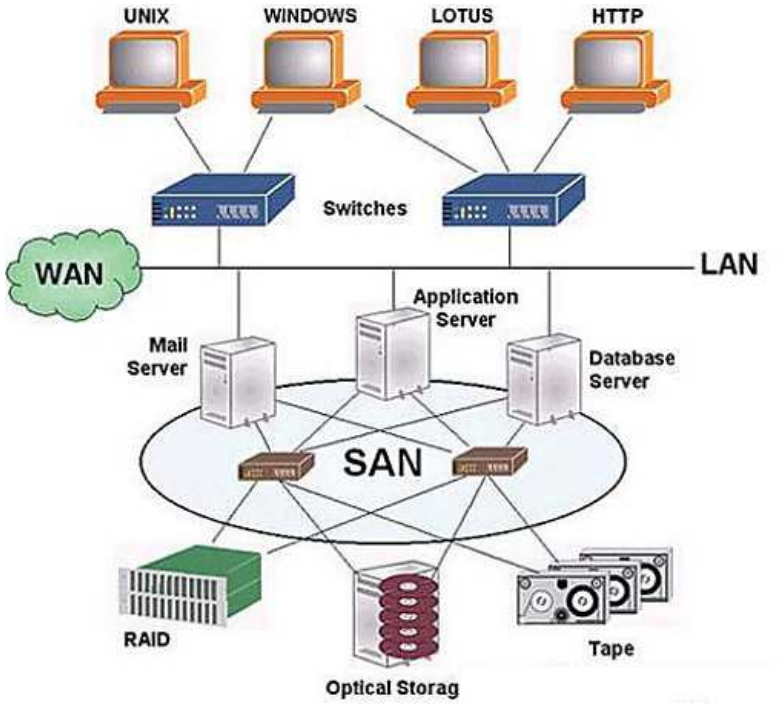

Figure 11:Approche de stockage SAN(Biulding, 2018)

IV.1.6.4. La virtualisation de serveurs

La virtualisation de serveurs consiste à faire fonctionner simultanément sur un seul serveur physique plusieurs serveurs virtuels.

La virtualisation de serveur permet de :

- Optimiser les performances d'un serveur.
- Réduire la surface au sol, la consommation électrique, le besoin de climatisation et le nombre d'administrateurs.
- Réaliser des économies (locaux, consommation électrique, personnel).

Figure 12 :Virtualisation des serveurs*(Ben Mahmoud, 2015)*

Ces quatre (4) types de virtualisations présentent chacun des caractéristiques différents de par leur tâches mais ont tous le même objectif : optimiser le fonctionnement du réseau et réduire l'utilisation de matériels physiques.

IV.2. Data center

Un data center ou centre de données, est une infrastructure composée d'un réseau d'ordinateurs et d'espaces de stockage. Cette infrastructure peut être utilisée par les entreprises pour organiser, traiter, stocker et entreposer de grandes quantités de données.(Pascal Faure, 2015)

Il est composé de :

- Salles sécurisées pour accueillir les équipements informatiques :

- Les baies, armoires de raccordement pour les serveurs, aux dimensions standardisées
- Les serveurs applicatifs, sur lesquels sont exécutés les logiciels
- Les serveurs de données, qui assurent le stockage des données
- Equipements réseau interconnectant les serveurs. Il s'agit notamment des routeurs, pare-feu, répartiteurs et commutateurs

- Infrastructures techniques assurant la continuité de l'alimentation électrique, du refroidissement des serveurs et de l'accès au réseau à très Haut Débit pour l'ensemble de ces ressources.

- Points d'accès aux réseaux électriques à haute tension et aux réseaux de télécommunication très Haut Débit.

- Un bâtiment spécialisé et sécurisé intégrant l'ensemble de ces composants.

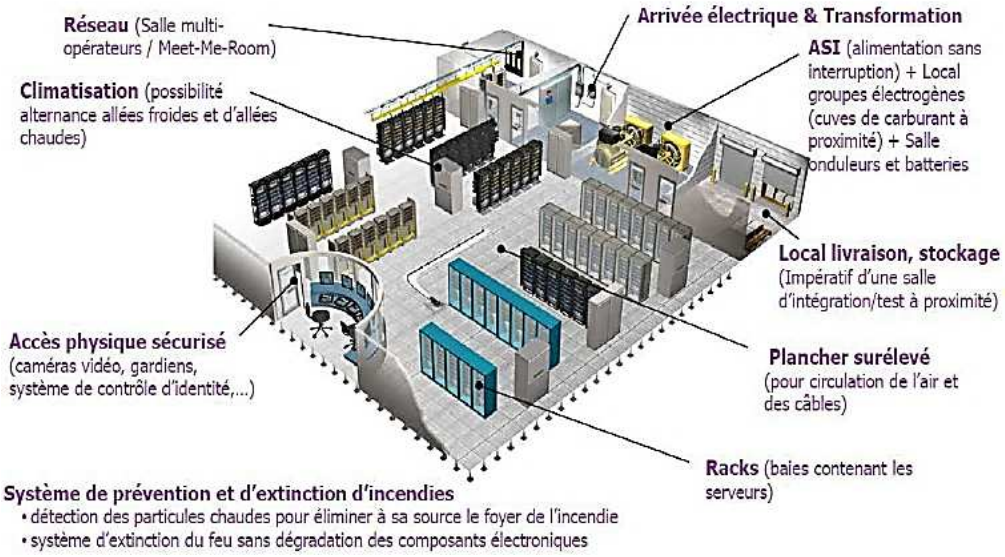

***Figure 13**:* Modélisation d'un data center(KARTIT, 2016)

V.Les services du cloud

Trois grands modèles d'usage du Cloud se dégagent actuellement, tous présentent des caractéristiques différentes, à savoir :

Infrastructure en tant que service (**IaaS**), plateforme en tant que service (**PaaS**) et application en tant que service(**SaaS**). La figure (15) représente l'organisation en couches de la pile de cloud de l'infrastructure physique aux applications. Ces niveaux d'abstraction peuvent également être considérés comme une architecture en couches ou les services d'une couche supérieure peuvent être composés des services de la couche sous-jacente.

Modèle de service	Outils d'accès et de gestion	Services offerts
SaaS 	Navigateur Web	Réseaux sociaux, suites bureautiques, CRM, traitement vidéo.
PaaS 	Environnement de développement	Langages de programmation, systèmes, gestionnaire de données structurés.
IaaS 	Gestionnaire d'infrastructure virtuelle	Serveurs de calcul, stockage de données, Pare-feu.

Figure 14 : Les couches du cloud computing*(Zohra, 2016)*

V.1. Software as a Services (SaaS):

SaaS est un modèlede distribution de logiciel au sein duquel un fournisseur tiers héberge les applications et les rend disponibles pour ses clients par l'intermédiaire d'internet.

Les logiciels et applications sont proposés aux utilisateurs par les fournisseurs selon un modèle par abonnement.

Les utilisateurs n'ont pas à gérer, installer ou mettre à niveau les logiciels, ce sont les fournisseurs qui s'en chargent.

Gmail est un exemple de Software as a Service. Le fournisseur de services met à disposition une application prête à l'emploi avec un espace de stockage qu'il pourra manipuler en gérant ses mails. Google, Twitter, Facebook et Flickr sont tous des exemples de SaaS, où les utilisateurs peuvent avoir accès aux services via n'importe quel appareil disposant d'une connexion à Internet.

Avantages et inconvénients du modèle SaaS

Avantages	Inconvénients
- Pas de coûts d'équipements supplémentaires **-Pas de frais d'installation** **-Paiement à l'utilisation** **-Utilisation évolutive** **-Mises à jour automatiques**	-Dépendances des prestataires -Sécurité (intrusion, piratage)

V.2. Platform as a Services (PaaS)

PaaS fournit aux utilisateurs un environnement dans lequel ils peuvent développer, gérer et mettre à disposition des applications.

Permet aux entreprises de se concentrer sur le développement sans se soucier de l'infrastructure sous-jacente.

Les fournisseurs gèrent la sécurité, les systèmes d'exploitation, les logiciels de serveur et les sauvegardes.

Un exemple est AWS Elastic Beanstalk qui est le service PaaS proposé par Amazon. Il permet d'automatiser le déploiement d'applications sur de multiples instances virtuelles.

Avantages et inconvénients du modèle PaaS

Avantages	Inconvénients
- Agilité :(garder le contrôle sur l'outillage et installer que les outils qui seront vraiment utiles). **-Paiement à l'utilisation** **-Les développeurs ne perdront plus jamais leur code, quand ils développeront sur le PaaS, avec la réplication et le back-up (sauvegarde de données) automatisés.**	-limitation des langages - Pas de personnalisation dans la configuration des machines virtuelles

V.3. Infrastructure as a services (IaaS)

Dans ce service un fournisseur offre aux utilisateurs l'accès à des ressources informatiques fondamentales comme des serveurs du stockage et de l'équipement de réseau.

L'utilisateur peut contrôler le système d'exploitation, le système de stockage, les applications logicielles et les composants réseaux mais pas l'infrastructure sous-jacente.

IaaS permet la virtualisation des tâches d'administration, ce qui libère du temps pour d'autres activités à plus forte valeur ajoutée.

Au lieu d'acheter le matériel, les utilisateurs paient pour l'IaaS à la demande.

Un exemple concret est celui de Netflix qui loue les infrastructures mises à disposition par Amazon Web Service et qui gère elle-même sa plate-forme ainsi que les transactions et les flux du stream sur sa plate-forme.

Avantages et inconvénients du modèle IaaS

Avantages	Inconvénients
- Réduire les coûts **- Évolutivité accrue (Flexibilité d'utilisation)** **- Déploiement simple** **-Capacité de stockage infini**	- Dépendance vis-à-vis du fournisseur - L'accès à Internet est indispensable - Problèmes éventuels liés à la politique de confidentialité(l'emplacement des serveurs du fournisseur)

Voici comment sont répartis les services sur un modèle Cloud en comparaison avec le modèle classique d'hébergement.

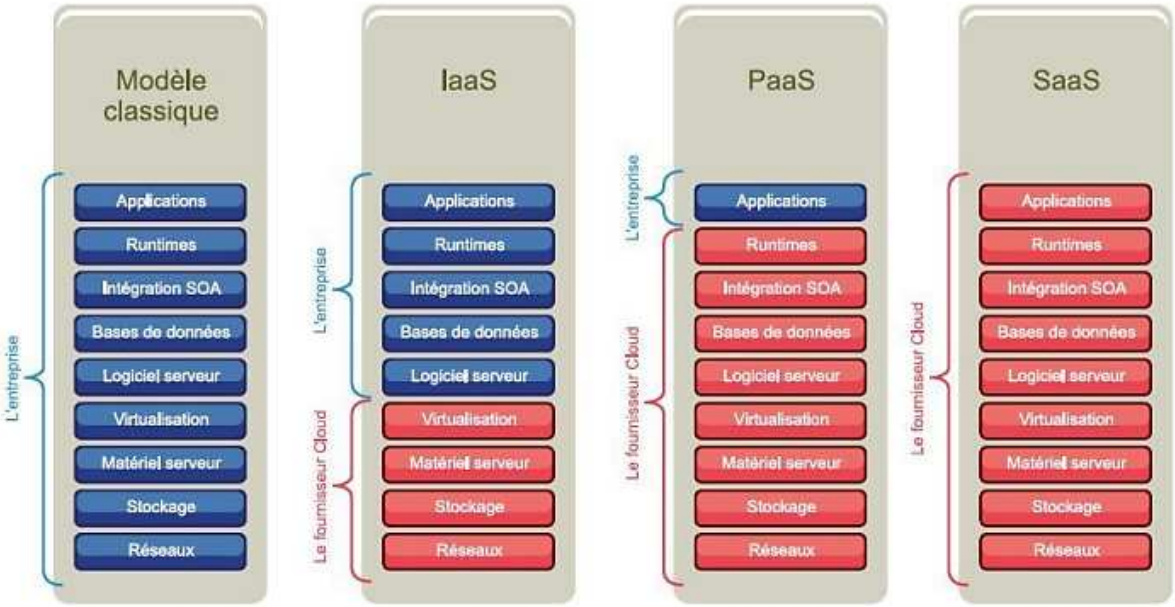

Figure 15:modèle classique et modèle cloud

VI. Les modèles de déploiement dans cloud

Selon la définition du Cloud Computing donnée par le NIST, il existe quatre modèles de déploiement des services de Cloud, à savoir : Cloud privé, Cloud communautaire, Cloud public et Cloud hybride. Ces modèles permettent de définir le degré d'accès de l'utilisateur final aux fournisseurs de services Cloud.

VI.1. Cloud public :

Un Cloud public repose sur le modèle standard de Cloud computing, dans lequel un fournisseur de services rend des ressources, telles que des applications et du stockage, accessibles au grand public via Internet. Les services de Cloud public disponibles pour quiconque souhaite les utiliser ou les acheter.

C'est une structure souple et ouverte, géré par un fournisseur tiers.

Exemple : Microsoft Azure, Amazon Web Services (AWS), …etc.

VI.2. Cloud privé :

Un Cloud privé est un ensemble des services et des ressources disponibles à un seul client par exemple une entreprise.

Il peut être géré par l'entreprise elle-même,oubien avec ses branches, dans ce cas il s'appelle « **Le Cloud privé Interne** ».

Il peut être géré par un prestataire externe loué par l'entreprise, dans ce cas on l'appelle « **Le Cloud privé Externe**»,il est accessible via des réseaux sécurisés de type VPN[10](Virtual Private Network).(BERKANI Nassima, 2015/2016)

L'avantage de ce type du Cloud par rapport au Cloud public réside dans l'aspect de la sécurité et la protection des données.
Ce type de cloud est :

[10]**VPN :** Virtual Private Network (réseau privé virtuel) désigne un réseau crypté dans le réseau Internet, qui permet à une société dont les locaux seraient géographiquement dispersés de communiquer et partager des documents de manière complètement sécurisée

- ✓ Cher pour le client.
- ✓ Dédie et sécurisé.
- ✓ Moins flexible comparé au cloud public.

Exemples : Eucalyptus, OpenNebula et OpenStack.(voir chapitre3)

VI.3. Cloud communautaire :

Il s'agit d'un genre de Cloud où les ressources sont partagées par plusieurs organisations.

Cette communauté d'organisation peut partager les tâches de gestion de ces ressources, comme la sécurisation des données, le déploiement d'applications, l'authentification …etc. Ces ressources, qui peuvent être installées dans ou hors-sites (les sites des organisations de la communauté), peuvent être gérées par une partie tierce comme par une des organisations faisant partie de la communauté.

Exemples :OpenCirrus formé par HP, Intel et Yahoo.

VI.4. Cloud hybride :

Avec le modèle hybride, les organisations peuvent associer le Cloud privé avec le publique, de cette façon, ces organisations peuvent déployer leurs principales et sensibles applications sur leur Cloud privé et déployé le reste sur le Cloud publique, ce qui peut réduire considérablement les coûts de construction et de management des ressources sur le Cloud.

Exemples: Google Apps, Amazon s3.

VII. Les plateformes du cloud computing

Dans cette partie nous allons citer quelques plateformes open source et propriétaires du cloud :

VII.1. Lesplateformes open source

Il existe de nombreuses plateformes open source qui sont émergées dans ce domaine du cloud, en voici quelques-unes :

1. Eucalyptus :

Issue d'un projet de recherche de l'université de Californie, cette plate-forme Cloud open source implémente des Cloud de type IaaS. L'objectif d'Eucalyptus est de permettre aux sites avec des clusters et une infrastructure de serveur existants d'héberger un cloud compatible avec l'AWS d'Amazon et (bientôt) l'API open source de Sun Cloud.

En outre, grâce à ses interfaces, Eucalyptus peut héberger des services de plate-forme cloud tels que AppScale (une application open source de Google AppEngine) et Hadoop, permettant de «mélanger et assortir» différents paradigmes de service et configurations dans le cloud. Enfin, Eucalyptus peut tirer parti d'une collection hétérogène de technologies de virtualisation au sein d'un même nuage, pour intégrer des ressources déjà virtualisées sans modifier leur configuration.

2. OpenNebula :(à voir en détail dans le chapitre 3)

Cette plateforme purement open source permet de déployer des Cloud privés, hybrides et publics. OpenNebula est une plateforme permettant de gérer un pool de ressources virtuelles. Vous pouvez créer des machines virtuelles et les configurer comme vous le feriez pour configurer une machine physique connectée à votre réseau.

OpenNebula est l'outil de gestion du cloud qui permet de synchroniser le stockage, le réseau et les techniques virtuelles, et aide les utilisateurs à déployer et gérer des machines virtuelles sur des ressources physiques selon les stratégies d'allocation des centres de données et des ressources cloud distantes. OpenNebula est principalement utilisé pour gérer le centre de données de cloud privé et l'infrastructure de cluster, et il prend également en charge le cloud hybride pour connecter l'infrastructure locale et publique. Ceci est très utile pour créer un environnement de cloud computing hautement évolutif.

3. **OpenStack :**(à voir en détail dans le chapitre 3)

Créé en juillet 2010 par la Nasa et l'hébergeur américain Rackspace, ce projet purement open source vise à fournir des solutions pour tous les types de nuages en étant simple à mettre en œuvre, hautement évolutive et riche en fonctionnalités. OpenStack est un système d'exploitation nuage qui contrôle de grandes surfaces de calcul, de stockage et les ressources réseau à travers un centre de données, le tout géré par un tableau de bord.

4. **Stratuslab**

Le projet StratusLab est née d'une collaboration académique informelle en 2008, cofinancé par la Commission Européenne afin d'élaborer une plateforme open source sur infrastructure as a services, il fournit des fonctionnalités pour la gestion dynamique des ressources typiques de calcul d'un nuage IaaS. Mais il fournit également des fonctionnalités supplémentaires pour simplifier la gestion de l'image et la fédération du nuage.

VII.2. Le cloud propriétaire

Quelques solutions propriétaires du cloud computing :

1. Windows Azure :

Azure est une plateforme de Microsoft pour les services PaaS du cloud computing. Il s'agit d'une plateforme de développement d'applications fournissant les services d'exécution et d'administration d'applications en offrant les outils nécessaires. Elle permet aux développeurs de programmer et de stocker directement leurs applications sur Internet en leur allouant dynamiquement des machines virtuelles de son centre de données (data center).

2. Google AppEngine :

AppEngine est une offre de Google pour les services de type PaaS. Le développement et le déploiement d'applications sur la plateforme de Google sont rendus possibles grâce à un SDK(22) conçu par Google et mis à la disposition des utilisateurs afin de leur permettre de développer en local pour ensuite déployer l'application vers l'Internet. L'idée est de permettre aux utilisateurs d'employer l'infrastructure de Google pour héberger leurs applications avec la

possibilité de définir le groupe d'utilisateurs de cette dernière. Ces applications bénéficient de la haute disponibilité des infrastructures de Google.

3. Amazon Web Service :

Amazon Web Services (AWS) est une division du groupe américain de commerce électronique Amazon.com, spécialisée dans les services de cloud computing à la demande pour les entreprises et particuliers.

Amazon Web Services fournit des services en ligne à d'autres sites internet ou applications clientes. La plupart d'entre eux ne sont pas directement exposés à l'utilisateur final, mais offrent des fonctionnalités que d'autres développeurs peuvent utiliser à travers des API.

En 2017, AWS propose plus de 90 services, comprenant le calcul, le stockage, le réseau, la base de données, l'analyse de données, des services applicatifs, du déploiement, de la gestion de système, de la gestion d'applications mobiles, des outils pour les développeurs et pour l'internet des objets. Les services les plus populaires sont Amazon Elastic Compute Cloud (EC2) et Amazon Simple Storage Service (S3).

VIII.Les avantages et les inconvénients du cloud Computing

Les avantages du cloud :

Le Cloud Computing offre de nombreux avantages que plusieurs travaux ont listés. Parmi ces avantages nous citons :

- Flexibilité : le Cloud utilise multi-locataire pour ces ressources au cours de l'exécution, une application peut utiliser plusieurs ressources. Cela offre une possibilité de demanderd'autres ressources nécessaires.
- Optimisation des coûts : réduction des effectifs informatiques et fixation du prix en fonction de la durée d'utilisation des ressources informatiques sans investissement initiale lourd.
- Portabilité : les organisations peuvent utiliser leurs puissances informatiques partout où les utilisateurs peuvent avoir un accès dans n'importe quelle localisation géographique.

- Simplicité d'utilisation : le Cloud offre des applications et des services installés et faciles à utiliser à travers des pages web.

Les inconvénients du cloud :

Comme nous avons mentionné, chaque nouvelle technologie arrivée porte des avantageset malheureusement suivi par quelques limites rencontrées par des utilisateurs du Cloud. Ces limites sont présentées comme suit :

- La gestion d'énergie : afin de définir un plan d'utilisation des ressources, le fournisseur doit définir une stratégie pour la gestion d'énergie (consommation d'électricité).

- Confidentialité et sécurité : dans n'importe quelle technologie, la sécurité pose toujours des problèmes. Le problème concerne les attaques lors des opérations du transfert de données. Dans le cloud, ce problème est posé dans les cas des cloud publics.

- Gestion de ressources : ce problème représente toujours les limites de chaque technologie. A cause de la nature multidimensionnelle des machines virtuelles, la gestion des ressources sera compliquée.

- Dépendance : en cas ou l'entreprise (ou client final) souhaite des fonctionnalités très spécifiques, il est peut-être difficile de convaincre le fournisseur de proposer ces fonctionnalités. Le client final doit choisir un fournisseur en qu'il a la confiance.

- Migration vers une autre offre difficile : il n'existe pas pour l'instant un standard entre les différents acteurs du domaine, donc, le risque d'incompatibilité du transfert de données.

Conclusion

Au cours de ce chapitre nous avons présenté les principaux concepts et points clés du Cloud Computing. La naissance du Cloud Computing a donné un nouveaumode de consommation de l'informatique et a changé la manière d'investissement desentreprises dans les infrastructures informatiques.

Malgré tous les avantages que le cloud nous offre, sa sécurité sera toujours une préoccupation majeure en raison de sa nature ouverte et publique.

Dans le chapitre suivant, nous concentrons sur la présentation de la sécurité dans le Cloud Computing, qui permet de garantir la confidentialité, l'intégrité, l'authenticité et ladisponibilité des informations.

Chapitre II : La sécurité dans le Cloud Computing

Introduction

Le Cloud computing est devenu incontournable dans la mise en place et la fourniture des services informatiques pour les entreprises. Aujourd'hui, plusieurs entreprises considèrent le cloud computing comme une force majeure à modifier de façon significative l'ensemble de la technologie d'information, la façon dont les centres de données sont construits, la façon dont les logiciels sont déployés, le traitement des mises à jour, etc.

Malgré tous ses avantages, la sécurité des données reste un sujet d'inquiétude pour lesentreprises et représente un frein majeur pourl'adoption de cette technologie. Pour atténuer ce problème de sécurité, nous présentons à travers ce chapitre la sécurité dans le cloudcomputing en étudiant les différentesattaques et menaces liées à la sécurité du cloud ainsi que les solutionsproposées dans ce domaine.

I. Sécurité informatique

La sécurité informatique, d'une manière générale, désigne l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour :

- Assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

- Protéger l'information contre l'accès, l'utilisation, la divulgation, la perturbation, la modification ou la destruction afin de garantir la confidentialité, l'intégrité et la disponibilité.

La sécurité informatique vise généralement cinq principaux objectifs :

1. L'intégrité : c'est-à-dire garantir que les données sont bien celles que l'on croit être.
2. La confidentialité : consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées.

3. La disponibilité : permettant de maintenir le bon fonctionnement du système d'information.
4. La non-répudiation : permettant de garantir qu'une transaction ne peut être niée.
5. L'authentification : Consiste à s'assurer de l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être.

II.La sécurité dans le cloud computing

La sécurité du Cloud(Cloud Security) est un sous domaine du Cloud Computing en relation avec la sécurité informatique. Elle implique des concepts tels que la sécurité des réseaux, du matériel et les stratégies du contrôle qui sont déployées afin de protéger les données, les applications et l'infrastructure associées au Cloud Computing. Un aspect important du Cloud est la notion d'interconnexion avec divers matériels qui rend difficile et nécessaire la sécurisation de ces environnements.

Pour cela, nous devons définir quelles sont les menaces liées au Cloud et identifier les risques de sécurité afin d'aboutir à un modèle fonctionnel et à l'abri des attaques.

II.2. Les menaces liées au cloud computing

Une menace de sécurité correspond à tout événement susceptible de compromettre l'intégrité d'une architecture Cloud en mettant en péril les données, les serveurs, les infrastructures ou le réseau.

On distingue deux formes de menaces :

1. Menaces involontaires :

Ce type de menace se présente sous forme de catastrophes naturelles telles qu'un tremblement de terre ou un ouragan, mettant en danger un ou plusieurs Datacenters, serveurs ou lignes réseaux ou électriques.

Elle se présente également sous forme de négligences techniques pouvant paralyser les systèmes ou même entraîner la perte temporaire ou permanente de données. Ces menaces sont généralement la résultante de

- Incendies involontaires,
- Catastrophes naturelles,

- Erreurs de manipulation,
- Mauvaise conception pouvant aboutir à la surcharge des réseaux, surchauffe ou perte d'une partie ou de la totalité du matériel, etc…

2. Menaces volontaires :

Plusieurs études menées par des spécialistes tels que ISACA[11] (Information Systems Audit and Control Association) et CSA[12] (Cloud Security Alliance) ont permis d'identifier les points qui constituent les menaces involontaires à la sécurité des données et à celles des applications en cloud.

1. **Violation de données** : les données sensibles tombent dans les mains de concurrents. Une faille sur une application permet à un utilisateur d'avoir accès non seulement à des données liées à cette application, mais également à d'autres.

2. **Perte de données** : une perte de données équivaut à une perte de compétitivité pour une entreprise, voire un manque à gagner. Elle peut faire suite à une suppression accidentelle, une catastrophe naturelle ou un acte malveillant.

3. **Vol de compte ou de trafic de services** : l'hameçonnage et la fraude correspondent à des attaques très répandues et anciennes. Elles ont vu le jour avec le commencement de l'internet et pénalisent lourdement les entreprises ou les particuliers. Un attaquant bien déterminé joue avec les informations qu'il dérobe par des suppressions, des manipulations etc. et parvient ainsi à aborder les applications critiques de 1'entreprise.

4. **Attaque d'initiés internes** :Un utilisateur malveillant, tel qu'un administrateur, peut accéder à des informations confidentielles et avoir de plus en plus d'accès à des systèmes et données critiques.

5. **Gestion de l'identité et accès médiocre**:Les cyber-attaquants qui parviennent à usurper l'identité d'utilisateurs légitimes, qu'ils soient opérateurs ou développeurs d'applications,

11 ISACA : est une association professionnelle internationale dont l'objectif est d'améliorer la gouvernance des systèmes d'information, notamment par l'amélioration des méthodes d'audit informatique

12 CSA : La Cloud Security Alliance (CSA) est une organisation à but non lucratif dont la mission est de promouvoir l'utilisation des meilleures pratiques pour fournir une assurance de sécurité dans le Cloud Computing et de fournir une formation sur les utilisations du Cloud Computing pour aider à sécuriser toutes les autres formes de sécurité informatique.

peuvent lire, modifier et supprimer des données, voler des informations ou espionner, et injecter des applications ou du code malveillant qui semble provenir d'un utilisateur légitime.

6. **Vulnérabilité des technologies partagées** : Les fournisseurs d'infrastructure cloud offrent leurs services de manière évolutive en partageant une infrastructure physique, des plates-formes ou des applications. Parfois, les composants de l'infrastructure peuvent ne pas offrir l'isolement nécessaire pour être utilisés par plusieurs clients, ce qui peut entraîner des vulnérabilités de technologies partagées.

7. **Vulnérabilités des systèmes :**Les vulnérabilités du système sont des bugs exploitables dans les programmes que les pirates peuvent utiliser pour infiltrer un système afin de voler des données, prendre le contrôle ou interrompre le service.

II.3. Vulnérabilité, Menace et risque

Le terme problème de sécurité est un terme plus général qui regroupe les sous termes vulnérabilité, menace et risque. Définissons à présent plus précisément :

- **La vulnérabilité:**est une faute accidentelle ou intentionnelle (avec ou sans volonté de nuire), dans la spécification, la conception ou la configuration du système, ou dans la façon selon laquelle il est utilisé. La vulnérabilité peut être compromise par une menace . Il existe une grande variété de vulnérabilités visant les Services Cloud. Toutefois certaines sont plus connues et plus dangereuses que d'autres : Tel que les Failles d'injection qui se produisent lorsque l'accès d'un utilisateur n'est pas contrôlé.
- **La menace:**est une situation qui peut exploiter une vulnérabilité (Menaces délibérées (attaques) ou involontaires (erreurs ou incidents)). Les menaces engendrent des risques, des couts humains et financiers, perte de confidentialité de données sensibles, etc. les risques peuvent se réaliser si les systèmes présentent des vulnérabilités.

- **Le risque :**est la probabilité d'une menace sous forme d'attaque, en profitant d'une vulnérabilité.

II.4. Les attaques dans le cloud computing

Nous allons d'abord définir qu'est-ce que c'est une attaque, ensuitenous allons définir les attaques les plus potentielles sur le Cloud.

Attaque : est toute action compromettant la sécurité de l'information, c'est la réalisation d'une menace.

Une attaque sur un réseau est de deux types distincts :

- **Passive :**Une personne malintentionnée s'infiltre sur le réseau afin d'observer le flux de données ou intercepter des messages sans interférer entre le serveur et le client (Analyse de trafic, capture d'informations).

- **Active :**L'intrus va modifier ou envoyer de fausses informations au client ou au serveur une fois qu'il se sera mis au centre de la communication (Mascarade, Modification d'informations, Déni de service).

Attaques sur les données et le réseau connues sous le nom de Cyberattaques, elles ciblent la circulation normale des informations et ont principalement pour objectif :

- **L'Interruption** : qui vise la disponibilité des données et des ressources.
- **L'Interception** : qui vise la confidentialité des données.
- **La Modification** : qui vise l'intégrité des données.
- **La Fabrication** : qui vise l'authenticité des données.

Après avoir compris le principe d'attaque en voici les attaques potentielles sur le cloud :

II.4.1. Attaque par Deni de Service (DoS et DDoS) :

Denisde service (Denial of Service) est une attaque qui consiste à produire une interruption ouunralentissement d'un ouplusieurs services parune surcharge réseau dans le but de réussir à empêcher les utilisateurs légitimes d'un service de l'utiliser.

Ce type d'attaquepeut toucher toutserveur d'entreprise ou toutparticulier relié à internet.

Peut-être provoquée par une seule machine ou simultanément par plusieurs machines (DistributedDoS).

Deux méthodes peuvent produire un déni de service :

- Générer un grand volume de trafic en le faisant passer pour un trafic légitime. Ce trafic sature le réseau et empêche le trafic normal de passer.
- Générer un trafic mal formé produisant un comportement erroné des applications.

Il existe de nombreusesfaçonspour effectuer ou réaliseruneattaque DoS, nous citons ci-après les méthodes les plus fréquentes :

- Saturation par UDP[13]
- TCP[14] SYN
- Usurpation d'adresse source/attaque LAND

Ces trois attaques sont celles qui arrivent le plus souvent et partagent des caractéristiques avec bien d'autres types d'attaques.

II.4.1.1.Saturation par UDP

Une attaque parsaturation UDP (User Datagram Protocol-protocol de datagramme utilisateur) peut être initiée en envoyant un nombre important de paquets UDP à des ports aléatoires sur l'hôte ciblé. L'hôte attaqué va :

- Vérifier le trafic provenant de l'application qui écoute sur ce port.
- Vérifier qu'aucune application n'écoute sur ce port.
- Répondre avec un paquet de destination ICMP[15] (Internet control message protocol)inaccessible .

[13]**UDP** :est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport du modèle OSI, quatrième couche de ce modèle. Il a été défini par David P. Reed Le rôle de ce protocole est de permettre la transmission de données (sous forme de datagrammes) de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Aucune communication préalable n'est requise pour établir la connexion). UDP utilise un mode de transmission sans connexion.

[14]**TCP** (Transmission Control Protocol-Protocole de Contrôle de Transmission) :est un protocole de transport fiable,en mode connecté.

[15]**ICMP :**est l'un des protocoles fondamentaux constituant la suite des protocoles Internet. Il est utilisé pour véhiculer des messages de contrôle et d'erreur pour cette suite de protocoles, par exemple lorsqu'un service ou un hôte est inaccessible.

Ainsi, lorsqu'un volume important de paquets UDP est envoyé, le système attaqué renvoie un grand nombre de paquets ICMP et finit par devenir indisponible pour d'autres clients. L'attaquant peut également falsifier l'adresse IP[16] des paquets UDP afin de s'assurer que les nombreux paquets ICMP envoyés en retour ne l'atteignent pas et de garder anonyme son emplacement réseau.

Ceci est illustré dans la figure ci-après :

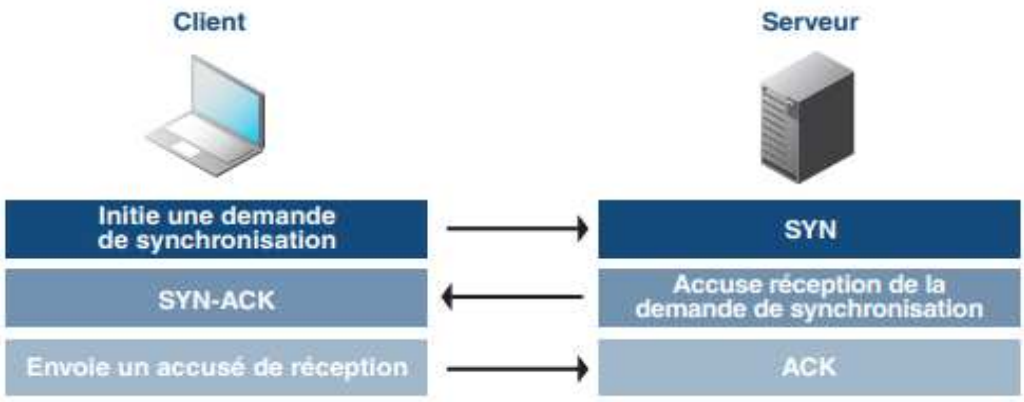

Figure 16: Attaque par saturation UDP (Infoblox, 2013)

II.4.1.1. TCP SYN

Les attaques par TCP SYN sont des attaques DoS qui tentent de saturer le serveur DNS[17] avec de nouvelles demandes de connexion TCP. Un client initie généralement une connexion TCP par le biais de trois messages d'établissement de liaison :

- Le client demande une connexion en envoyant un message SYN (synchroniser) au serveur.
- Le serveur accuse réception de la demande en renvoyant un message SYN-ACK au client.

---

[16] **Adresse IP :** est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque périphérique relié à un réseau informatique qui utilise l'Internet Protocol. L'adresse IP est à la base du système d'acheminement (le routage) des paquets de données sur Internet.

[17] DNS : (Le Domain Name System) le service informatique distribué utilisé pour traduire les noms de domaine Internet en adresse IP

- Le client répond en renvoyant à son tour un message ACK, ce qui a pour effet d'établir la connexion.

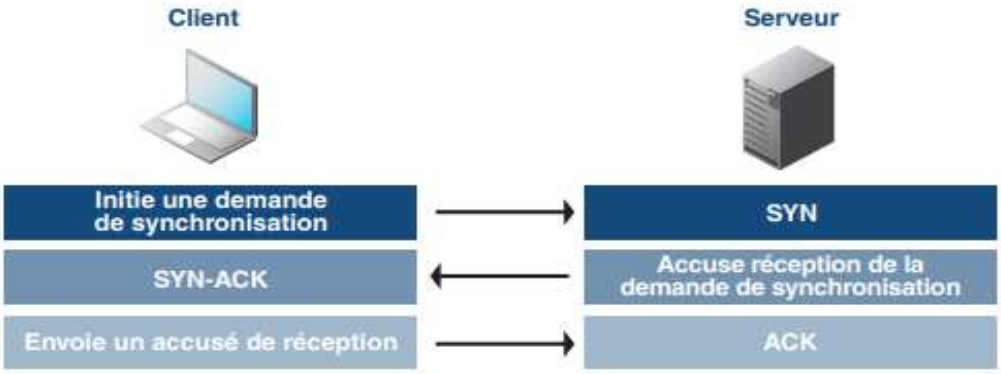

Figure 17: Connexion TCP normale *(Infoblox, 2013)*

Une connexion TCP ne peut s'établir que lorsque ces 3 étapes ont été franchies.

L'attaque SYN consiste à envoyer un grand nombre de requêtes SYN à un hôte avec une adresse IP source inexistante ou invalide.

La victime répond par un paquet SYN-ACK et se met en attente d'un paquet ACK de l'intrus

Chaque demande est mise dans une file d'attente en attendant que l'ACK correspondant arrive.

Les réponses étant destinées à une fausse adresse, la victime ne recevra jamais de paquet ACK et la file est vite débordée.Ayant la file débordée, la victime ne pourra plus répondre à une demande légitime de connexion.

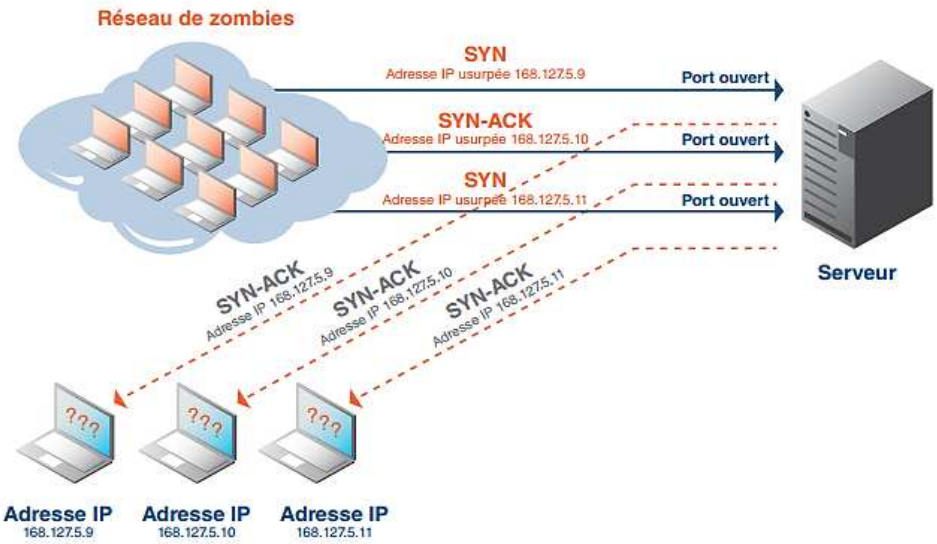

Figure 3: Attaque par saturation TCP/SYN*(Infoblox, 2013)*

II.4.1.3. Usurpation d'adresse source/attaque LAND

LAND (Local Area Network Denial/ Deni de réseau local) est une forme courante d'attaque DoS qui consiste à envoyer des paquets TCP SYN usurpés comportant une adresse source IP et un numéro de port identiques à ceux de la victime.

L'hôte attaqué dans ce cas répond à lui-même continuellement.

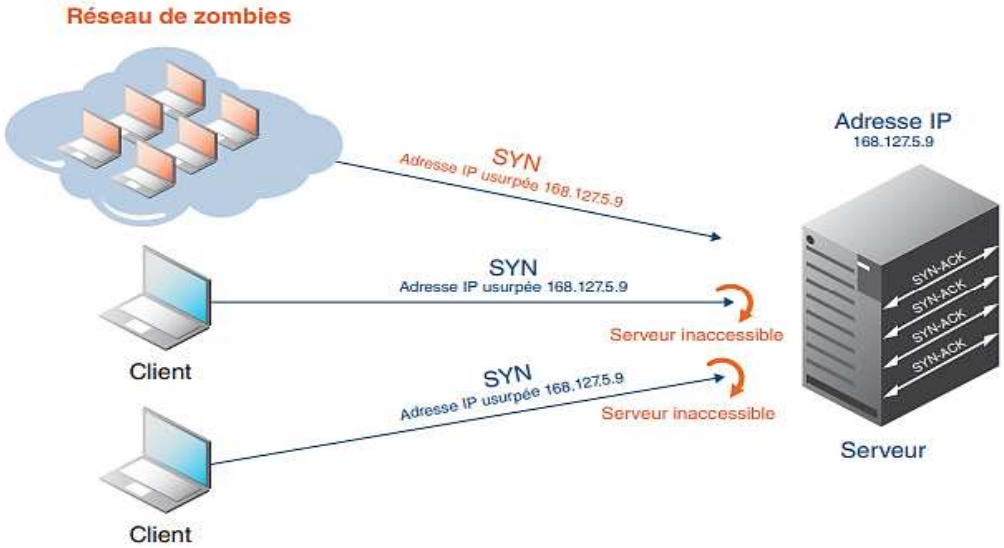

Figure 19: Attaque LAND *(Infoblox, 2013)*

Exemple d'attaque DoS :

- Octobre 2009 L'attaque sur le service hébergé dans le Cloud d'Amazon qui a connu plusieurs interruptions de service en raison d'une attaque DoS.

II.4.2. AttaqueSQL injection :

Une injection SQL est un type d'exploitation d'une faille de sécurité d'une application interagissant avec une base de données. L'attaquant détourne les requêtes en y injectant une chaîne non prévue par le développeur et pouvant compromettre la sécurité du système.

C'est un cas particulier d'un type de vulnérabilité plus général qui peut se manifester lorsqu'un langage est imbriqué dans un autre.

Lorsqu'on évoque l'injection SQL, on parle de faille dans un site Web, mais il n'y a pas que les sites qui sont touchés par cette faille n'importe quelle application dialoguant avec une base de données en utilisant des requêtes sur lesquelles l'utilisateur a une influence peut être vulnérable aux injections SQL.

Exemple de requête :

```php
<?php

// On récupère les variables envoyées par le formulaire
$login = $_POST['login'];
$password = $_POST['password'];

// Connexion à la BDD en PDO
try { $bdd = new PDO('mysql:host=localhost;dbname=bdd','root',''); }
catch (Exeption $e) { die('Erreur : ' .$e->getMessage())  or die(print_r($bdd->errorInfo())); }

// Requête SQL
$req = $bdd->query("SELECT * FROM utilisateurs WHERE login='$login' AND password='$password'");

?>
```

Figure 5 :Une requête SQL qui permet la connexion à un espace membre*(senges, 2018)*

Cette requête aurait pour effet de sélectionner l'utilisateur en question si le nom d'utilisateur et le mot de passe entrés sont dans notre base de données. Si l'un des deux est erroné, la requête ne renverra aucun résultat.

```php
<?php

$req = $bdd->query("SELECT * FROM utilisateurs WHERE login='jean' # AND password=''");

// Qui sera interprété de la façon suivante

$req = $bdd->query("SELECT * FROM utilisateurs WHERE login='jean'");

?>
```

Figure 6 : Une requête SQL injectée*(senges, 2018)*

Le symbole # permet de faire un commentaire en PHP donc tous ce qui suit ce symbole n'est pas pris en compte dans la requête SQL. Grâce à cette injection l'utilisateur va pouvoir se connecter au compte de login sans connaître son mot de passe.

Exemples

- En juin 2011, un groupe d'adolescents est accusé d'une attaque par injection SQL sur le site de Sony. (Les mots de passe et informations personnelles de plus d'un million de clients sont volées et vendues au marché noir).

- En juillet 2012, Yahoo annonce avoir été victime d'une attaque par injection SQL et avoir exposé les informations confidentielles de près d'un demi-million de leurs clients.

II.4.3. Attaque Man in themiddle (MITM)

Ce type d'attaque peut se produire lorsqu'une communication est établie dans deux nœuds système.

Les attaquants du système de communication modifient le contenu du message.

Une attaque interceptée permet à un acteur malveillant d'interrompre, d'envoyer et de recevoir des données entre deux utilisateurs.

La plupart des attaques de type « man in the middle » consistent à écouter le réseau à l'aide d'un outil appelé sniffer[18].

IL existe différentes techniques parmi elles :

- L'empoisonnement d'un serveur DNS(DNS poisoning)
- Le déni de service (DoS) ou l'imposture ARP (AddressResolution Protocol)
- Session hijacking

II.4.3. 1. DNS poisoning

L'empoisonnement d'un serveur DNS(Domain Name Server) ou pollution de cache DNS est une technique permettant de faire passer de fausses requêtes à des serveurs DNS qui les exécuteront en pensant qu'elles sont valides, une fois que l'attaque est réussie un hacker peut rediriger un internaute vers un faux site infesté de virus ou encore pour lui usurper des informations personnelles.

II.4.3. 2. L'imposture ARP[19]

L'imposture ARP (AddressResolution Protocol) est une technique utilisée par les hackers pour détourner des flux de communications transitant entre une machine cible et un routeur au sein d'un réseau local où le protocol ARP est implémenté. Cette attaque permet au hacker d'observer, modifier ou même supprimer les messages (paquets) envoyés par une machine cible à sa passerelle (le routeur).

[18]Sniffer : Un analyseur de paquets est un logiciel pouvant lire ou enregistrer des données transitant par le biais d'un réseau local non-commuté.
[19]ARP : est un protocole utilisé pour traduire une adresse de protocole de couche réseau (typiquement une adresse IPv4) en une adresse de protocole de couche de liaison (typiquement une adresse MAC).

II.4.3. 3. Session hijacking

Une Attaques de Session Hijacking est une technique qui consiste à intercepter une session TCP existante entre deux machines afin de la détourner. L'attaquant dérobe un identifiant de session valide afin de pénétrer dans le système et d'exploiter les données la plupart des authentifications au niveau d'une session TCP s'effectue uniquement lors de l'établissement de cette dernière cela permet pour un attaquant de pouvoir gagner un accès à une machine d'une manière assez facile.

Exampled'attaque Man in the middle

- Mai 2017 L'attaque par phishing qui a touché des milliers de comptesGoogle

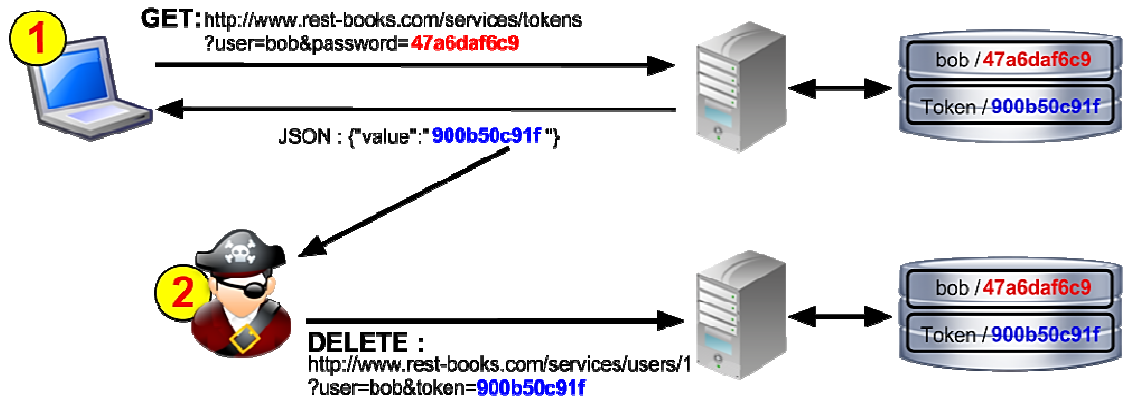

Figure 7 : Attaque Main In The Middle*(Patil, 2016)*

II.4.4. Attaqued'authentification :

L'authentification est le premier rempart aux attaques informatique, cette faille de sécurité regroupe toutes les vulnérabilités pouvant mener à une usurpation de l'identitéd'un utilisateur, d'un service ou d'une application.

Ces points de faiblesse dans les applications Web peuvent ouvrir à des attaquants des accès à des fonctionnalités des applications Web auxquelles ils n'ont pas le droit. Cela peut donc leur permettre de voler des informations ou d'endommager le bon fonctionnement de l'application.

Pour comprendre comment les attaques peuvent être menées, il faut comprendre le mécanisme d'authentification le plus commun des applications Web.

1. L'utilisateur non authentifié demande l'accès à une page Web
2. Le serveur renvoie une page d'authentification
3. L'utilisateur rempli le formulaire en fournissant un identifiant et un mot de passe et revoie ces informations au serveur web
4. Le serveur web fait appel à un service pour vérifier la validité du couple identifiant/mot de passe
5. Si la validité est avérée, le serveur web fournit un identifiant de session à l'utilisateur.
6. L'utilisateur peut utiliser l'application Web tant que la session est ouverte.

Les attaques pour usurper une identité peuvent être regroupées en deux catégories :

- Les attaques contre le système d'authentification qui cherchent à obtenir un droit d'accès
- Les usurpations de session qui permettent de s'affranchir de l'étape d'authentification

Les méthodes d'authentification utilisées :

Authentification simple :se base sur la vérification d'un seul élément.

Exemple : Le couple « nom d'utilisateur/mot de passe »

Authentification forte :combinaison de la vérification de deux facteurs ou plus(un mot de passe, un code NIP, une phrase secrète…etc.).

Exemple : jeton de sécurité +scan de l'iris

Authentification unique (Single-Sign-On) :on s'authentifie une fois pour toute et ensuite on peut accéder à plusieurs applications ou services situés sur des serveurs différents.

Exemples :Kantara initiative, OpenID

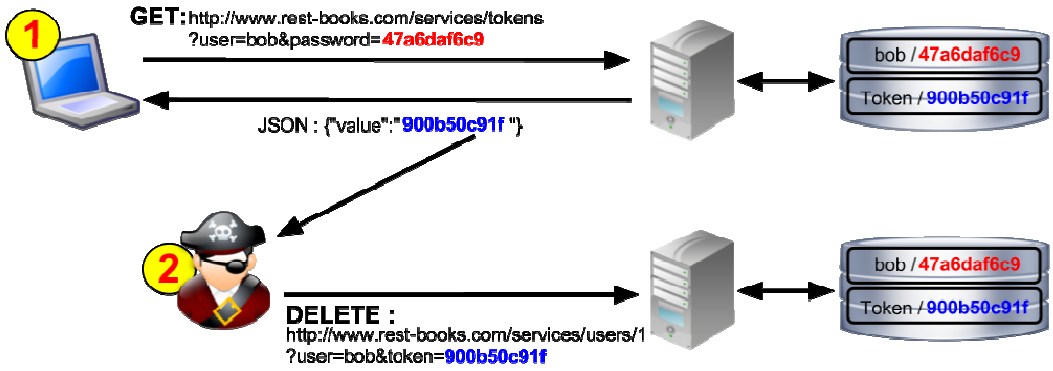

Figure 8 : Attaque d'authentification*(David, 2013)*

Exemple d'attaque authentification :

- 2014 : Publication des photos intimes de plusieurs célébrités.

II.5. Sécurité des données dans le cloud

Comme nous avons vu dans les sections précédentes, les infrastructures cloud sont, comme tout système informatique réparti, exposées à des problématiques de sécurité. En effet, le nombre et la diversité des utilisateurs de ses infrastructures ainsi que la quantité importante de composants matériels et logiciels impliquent la présence de menaces de sécurité.

Dans cette partie nous nous intéressons à la sécurité des données et nous présentons particulièrement quelques mécanismes proposés afin de protéger les données

II.5.1. La sécurité des infrastructures (sécurité physique)

La sécuritéphysique concerne tous les aspects liés à la maîtrise des systèmes (matériels, composants, câbles, etc. ...) et de l'environnement dans lequel ils se situent (locaux, alimentation énergétique, climatisation, etc.).

Le Cloud Computing, par nature, est associé à une « dématérialisation » de l'hébergement. En effet, le lieu d'hébergement du Cloud est généralement multiple, et réparti sur plusieurs data centers.

Sans vouloir être exhaustif, nous retiendronsque la sécurité physique repose essentiellement sur :

- Découper les locaux informatiques en zones de sécurité concentriques, regrouper le matériel les plus sensibles dans les zones mieux protégées

- Déporter du local les accès de maintenance ordinaire (eau, électricité, ascenseurs ...)

- Protéger loin des locaux les supports de sauvegarde (bandes, cassettes, CD ...), si possible du bâtiment. La répartition du système de stockage sur plusieurs centres de données permet de limiter les risques de perte totale du service et d'améliorer la tolérance de panne

- Contrôler l'accès par des systèmes à clé, cartes, digicodes, etc. Faciles à utiliser et dont les listes d'accès sont actualisées en permanence

- Installer des systèmes de surveillance extérieure permanente (caméras, détecteurs de présence, etc.)

- Enregistrer en vidéo les entrées et sorties (très dissuasif).

- Mettre en place une architecture de secours sur un site géographiquement éloigné, avec des équipements redondants permettant de réaliser un PCA[20] (Plan de Continuité d'Activité) sans interruption de service.

- Mettre en œuvre les pratiques suivantes :

✓ Etablir une politique d'accès générale comprenant toutes les exceptions.

✓ Identifier clairement les visiteurs par un badge spécial à durée limitée.

✓ Installer des sanitaires/vestiaires pour les visiteurs.

✓ Ne jamais laisser un visiteur seul se promener dans le bâtiment.

✓ Tout visiteur doit avoir une autorisation d'accès délivrée par un responsable (maintenance, entretien, visites, réunions, etc.).

✓ Concevoir le data center de façon à ce que la présence d'équipes de nettoyage ne soit pas nécessaire dans la salle des serveurs (mobilier antistatique, filtres à particules, etc.).

II.5.2. Le contrôle et la gestion des flux

Il est également important que l'infrastructure cloud se dote d'équipementspermettant de contrôler et gérer les flux. Classiquement, il s'agira de mettre en place despares-feux (firewalls) pour contrôler les flux venant de l'extérieur de l'infrastructure, desproxypour centraliser et contrôler les flux internes allant vers l'extérieur, et des reverseproxypour centraliser et contrôler les flux externes accédant aux ressources internes.

➢ **Pare-feu (firewall) :**

[20]PCA : est un document devant permettre à une entité de fonctionner même en cas de désastre ou de crise majeure

Un pare-feu est un élément du réseau informatique, soit logiciel ou matériel ou une combinaison des deux. Son rôle est de sécuriser un réseau en définissant les communications autorisées ou interdites.

Il permet d'interconnecter deux réseaux ou plus de niveau de sécurité différent par exemple (Internet et un réseau d'une entreprise). Il joue le rôle de sécurité en contrôlant les flux de données qui le traverse que ce soit en entrée ou en sortie, il permet ainsi de filtrer les communications, les analyser et en fin de les autoriser ou de les rejeter selon les règles de sécurité en vigueur.

- **Proxy :**

Un serveur proxy (traduction française de « proxy server », appelé aussi « serveur mandataire ») est à l'origine une machine faisant fonction d'intermédiaire entre les ordinateurs d'un réseau local et internet. Le serveur proxy est utilisé pour le web, il s'agit alors d'un proxy HTTP.

Le principe de fonctionnement basique d'un serveur proxy est assez simple : il s'agit d'un serveur "mandaté" par une application pour effectuer une requête sur Internet à sa place.

Ainsi, lorsqu'un utilisateur se connecte à internet à l'aide d'une application cliente configurée pour utiliser un serveur proxy, celle-ci va se connecter en premier lieu au serveur proxy et lui donner sa requête. Le serveur proxy va alors se connecter au serveur que l'application cliente cherche à joindre et lui transmettre la requête. Le serveur va ensuite donner sa réponse au proxy, qui va à son tour la transmettre à l'application cliente.

Une autre fonctionnalité des proxy est celle de filtrage, ou de contrôle de contenu, pour des raisons de sécurité. Le proxy examine les requêtes émises, et les traite suivant des règles prédéfinies.

II.5.3. La sécurité logique

Précédemment, dans la partie évoquant les principales technologies qui ont œuvré dans le développement du cloud computing, nous retrouvons notamment la virtualisation.

En effet, la sécurité logique que l'on souhaite intégrer est destinée à des plateformes virtualisées.

La sécurité et la confidentialité des données peuvent être gérées de différentes façons d'un point de vue logique :

- Sécurité des serveurs virtuels
- Colocation sécurisée
- Segmentation réseau

II.5.3.1. Sécurité des serveurs virtuels :

Généralement on discerne les bonnes pratiques de sécurité liées à la virtualisation en deux familles.

En premier lieu, il s'agit de sécuriser les systèmes en assurant une gestion des mises à jour de sécurité. La mise à jour de l'hyperviseur et de la partition de gestion, a priori à la charge de l'hébergeur, conduit dans la plupart des cas à un redémarrage du serveur. Pour éviter que les VMs soient indisponibles durant l'opération, un mécanisme de déplacement automatique des VMs vers un autre serveur est possible, voire recommandé.

La sécurisation des systèmes suppose également la réduction des surfaces d'attaque, en fixant au strict minimum les services de la « partition de gestion ». On protège également les fichiersdes disques virtuels par du contrôle d'accès, de l'audit, voire du chiffrement. Idéalement, on agit conformément aux recommandations des fournisseurs de l'hyperviseur (configuration des disques virtuels, installation de composants d'intégration dans les VMs, etc.) et des OS, en mettant en place un contrôle de conformité automatisé.

La seconde famille de bonnes pratiques concerne la notion d'isolation :il est nécessaire d'assurer l'isolement de la machine virtuelle et de la machine hôte.

II.5.3.2. Colocation sécurisée :

La colocation sécurisée consiste en l'hébergement sur le Cloud des applications et données de multiples clients (sociétés, organisations, entités métier…) au sein d'une seule et unique infrastructure physique, mutualisée, tout en respectant la sécurité, notamment au sens de la confidentialité. A juste titre, les sociétés-clientes du Cloud veulent être rassurées sur le fait que leurs données et traitements seront bien isolés et protégés des autres environnements hébergés sur l'infrastructure partagée.

II.5.3.3. Segmentation réseau

Il faut appliquer les mêmes règles dans la virtualisation que dans une architecture physique :(David GELIBERT, 2010)

• Cloisonner les différents rôles (serveurs frontaux, données, applications, pré-production …) sur des VMs différentes via des VLAN[21] différents entre le serveur physique et l'infrastructure du client

• Mettre en place des briques de sécurité (firewall, reverse proxy …) qui assurent les rôles de :

- Routage inter-VLAN : pour que les VMs communiquent sur des ports applicatifs spécifiés.

- Filtrage (analyse port source/destination)

- Sécurité applicative (vérification protocolaire).

II.5.4. La gestion et le contrôle d'accès

Le contrôle d'accès est réalisé pour définir le niveau d'accès de l'utilisateur à la ressource, dont celui-ci a été défini par l'organisation.

L'autorisation d'accès à une ressource informatique repose sur deux fonctions distinctes mais complémentaires : **l'identification** (un utilisateur autorisé dispose d'une identité numérique

[21]VLAN :« Un réseau local virtuel » Est un réseau informatique regroupe de façon logique et indépendante un ensemble de machines informatiques.

pour l'accès à cette ressource) et **l'authentification** (procédure permettant de vérifier les informations d'identification).

II.5.4.1. L'identification

Afin qu'une personne puisse se connecter à des services cloud, la fonction d'identification implique de lui définir une **identité numérique.** Selon l'article « Trust Requirements in Identity Management », l'identité est définie comme « un ensemble de caractéristiques propres par lesquelles une personne ou une organisation est connue ou reconnue. Ces éléments peuvent être définis, comme le nom, l'adresse, la nationalité, ou peuvent être innés comme les empreintes digitales »

L'identité numérique :est l'ensemble des traces numériques qu'une personne ou une collectivité laisse sur Internet qui peut être constituée par : un pseudo, un nom, des images, des vidéos, des adresses IP, des favoris, des commentaires etc.

II.5.4.2. L'authentification

Les facteurs d'authentification qui peuvent être utilisés dans un processus de connexion sont les suivants :

- Ce que l'on connait (facteur mémoriel) : une information que l'utilisateur a mémorisée et que lui seul connait (exemple : un mot de passe, un nom)

- Ce que l'on possède (facteur matériel) : une information que seul l'utilisateur possède et enregistrée dans un support (exemple : « une clé USB »[22]).

- Ce que l'on est (facteur corporel) : une information qui caractérise l'utilisateur avec une empreinte qui lui est propre (exemple : voix, pupille, empreinte digitale).

[22]Une clé USB :est un petit bloc facilement transportable et qui permet de stocker des données informatiques. C'est en quelque sorte le remplaçant de la disquette. La clé USB se branche, comme son nom l'indique, sur un port USB d'un ordinateur.

- Ce que l'on sait faire (facteur réactionnel) : une information ou un geste que seul l'utilisateur peut produire (exemple : une signature).

Les bonnes pratiques en la matière sont :

• La mise en place de mécanismes d'authentification forte. En général, l'authentification forte consistera à combiner deux facteurs d'authentification ou plusieurs (MultiFactor Authentication /MFA). Ces combinaisons peuvent être les suivantes :

- Connaissance d'un mot de passe et possession d'un objet.
- Connaissance d'un mot de passe et identification biométrique.
- Possession d'un objet et identification biométrique.

• L'identification de l'authentification afin de disposer d'une traçabilité des accès.

• La journalisation des authentifications réussies ou échouées.

II.5.5. La protection et la confidentialité des informations

II.5.5.1. Le chiffrement

Il s'agit du principal moyen pour rendre confidentiel les informations d'une organisation.

Le chiffrement est une méthode classique à base de clé publique/cléprivée ; seul le destinataire de l'information peut déchiffrer la donnée qui lui est destinée avecsa clé privée, connue de lui seul, mais pas du fournisseur de la solution Cloud et moins encored'un autre colocataire.

Le Cryptage du Cloud Computing estappliqué sur les données et les bases de données. Les principales technologies de chiffrement permettant la protection des informations sont :

II.5.5.1.1. Secure Socket Layer (SSL)

SSL est un procédé de sécurisation des transactions effectuées via Internet. Il repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur internet.

Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification.

Le SSL est une procédure standard depuis des années nous allons intéresser à quatre points essentielles :

1. Mettre en œuvre SSL chaque fois qu'il ya du trafic confidentiel sur les serveurs Web ou des lignes non garantis.

2. Avoir un moyen systématique de manipulation expiration et la délivrance des certificats SSL de sorte que vous n’avez pas perturbé les opérations commerciales.

3. Mettre en œuvre le protocole SSL pour le trafic Web de la console d'administration.

4. Assurez-vous d'utiliser les normes SSL acceptées par l'industrie. En cas de doute, reportez-vous aux dernières mises à jour de l'Institut National des Standards and Technologie (NIST). Les orientations actuelles dans NIST SP 800-522 recommande SSL v3.

II.5.5.1.2. Les Réseaux privés virtuels (VPN)

Un VPN (Virtual Private Network) est un type de réseau informatique qui permet la création de liens directs entre des ordinateurs distants.

Côté fonctionnement, le VPN repose sur la création d'un tunnel via « un protocole d'encapsulation »[23]entre les deux ordinateurs. Bien que distants, ces deux ordinateurs sont alors connectés à un même réseau local, virtuel.

Côté usage, le VPN gratuit ou payant permet à certains utilisateurs d'accéder à un réseau interne (celui d'une entreprise par exemple) tout en restant éloigné géographiquement de ce réseau. Mais le réseau privé virtuel est aussi utilisé pour masquer son adresse IP en se connectant à l'extérieur de son propre réseau local.

Le VPN participe alors à renforcer l'anonymat d'un utilisateur lorsqu'il navigue sur le Web et peut aussi servir à contourner la mise en place de restrictions et de filtrages géographiques.

II.5.5.1.3. Secure SocketShell (SSH)

Secure Socket Shell, est un protocole réseau qui permet aux administrateurs d'accéder à distance à un ordinateur, en toute sécurité.

SSH désigne également l'ensemble des utilitaires qui mettent en œuvre le protocole. Le protocole Secure Shell assure une authentification forte et des communications de données chiffrées sécurisées entre deux ordinateurs connectés sur un réseau peu sûr, tel qu'Internet.

SSH est largement utilisé par les administrateurs réseau pour gérer à distance les systèmes et les applications, car il leur permet de se connecter à un autre ordinateur sur un réseau, d'exécuter des commandes et de déplacer des fichiers d'un ordinateur à un autre.

II.5.5.1.4. Secure File Transfer Protocol (SFTP)

Il y a des exigences pour transférer des fichiers en toute sécurité vers et depuis votre Cloud privé, il faut s'assurer que vous établissez un processus de SFTP suivant :

1. Établir de provisionnement, et « les processus de certification »[24] de l'utilisateur.

[23]Un protocole d'encapsulation :est un procédé consistant à inclure les données d'un protocole dans un autre protocole.

[24]Les processus de certification : C'est un processus d'évaluation de la conformité qui aboutit à l'assurance écrite qu'un produit, une organisation ou une personne répond à certaines exigences

2. Établir un processus de gestion des utilisateurs d'identifier clairement et l'accès de l'utilisateur de commande.

3. Établir des autorisations d'accès appropriées et dossier isolement pour les utilisateurs SFTP.

4. Appliquer nettoyage de données de dossiers SFTP.

II.5.5.1.5. Transport Layer Security (TLS)

Il s'agit d'un protocole de sécurité qui consiste à assurer la confidentialité des informations échangées entre des applications et leurs utilisateurs sur internet.

TLS fait également en sorte d'éviter que des tiers puissent altérer ou falsifier vos messages. Il est constitué de deux couches :

- **Le Protocole TLS Record** :fournit une connexion sécurisée grâce aux méthodes comme DAE[25] (Data Encryption Standard, ou "standard de cryptage de données" en français). Il peut également être utilisé sans cryptage.

- **Le Protocole TLS Handshake** :permet au serveur et à l'ordinateur de s'identifier l'un à l'autre puis de choisir ensemble un algorithme de cryptage et des clés secrètes avant de commencer à s'envoyer des données ou des messages.

II.5.5.2 Les moyens périphériques

Des moyens complémentaires peuvent être mis en place en plus du chiffrement, afin de renforcer la protection des données stratégiques d'une organisation.

Les appareils de chiffrement sont un moyen pour les fonctions cryptographiques à exécuter sur le réseau. La logique de l'application fait un appel de programmation vers le module de

[25]DAE : est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de (dé)chiffrement. Ce principe est généralement lié au principe d'accès conditionne

chiffrement sur le dispositif pour chiffrer les données avant de les stocker dans la base de données. Voici quelques facteurs clés à considérer lors de l'utilisation d'un appareil de chiffrement :

- Performance : dispositifs de cryptage attachés réseau sont des appareils spécialisés construits dans le but d'exécuter des fonctions cryptographiques.

- Centralisée : L'appareil de chiffrement peut être utilisé par différents locataires sur le Cloud privé.

- Impact applications : Ces solutions offrent plusieurs façons d'appeler la fonction de cryptage, il peut être un appel de programmation qui envoie un champ à chiffrer avant le stockage dans la base de données.

- Les coûts de licence : Ces appareils peuvent être agréés par connecteur, par des fonctions cryptographiques, ou une taxe de l'appareil une fois l'entretien.

II.5.5. Les fournisseurs de solutions de sécurité

Face à l'adoption des services cloud par de nombreuses entreprises, la question de la sécurité cloud préoccupe de plus en plus d'entreprises. Les données, les accès et les workloads[26] doivent être sécurisés. Dans ce contexte, de grandes entreprises et des startups tentent de répondre à cette demande en proposant des solutions de sécurité pour les données, les applications, et bien plus encore

Les fournisseurs de solution de sécurité sur le cloud ont une priorité : maintenir un cycle de production fluide et continu.

Voici quelques fournisseurs de solutions de sécurité :

Avanan : Avanan propose une solution complète de sécurité cloud ouverte aux fournisseursde services cloud

Avg : Le fournisseur d'antivirus AVG propose des solutions de récupération de données,de backup, et de single sign-on ainsi qu'une intégration à Office 365 et VMware

[26]**Workloads :** logiciel ou composant du système d'exploitation qui met en œuvre ces techniques.

Cipher cloud: Est spécialisé dans les solutions de visibilité, de surveillance et de protection du cloud.

Netskope : La solution de Netskope permet de prévenir la perte de données, la visibilité etle renforcement de la sécurité sur le cloud.

Blue Coat: propose la première solution complète de sécurité du cloud grâce au rachat dela société innovante Elastica.

HyTrust : HyTrust offre des systèmes d'authentification, d'autorisation et d'audit pour les environnements virtuels d'entreprises.

LightCyber: permet de détecter les cyber-attaquants grâce aux technologies d'analyse comportementale intégrée à sa solution Active Breach Detection.

II.5.6. Les plateformes utilisées pour sécuriser le cloud

Voici quelques plateformes utilisées pour sécuriser le cloud :

A. Plateforme CASB

Un CASB « Cloud Access Security Broker » est un type de logiciel qui tend à sécuriser les applications SaaS des entreprises (Salesforce, Box,) et IaaS (OCI, AWS, Azure,) de manière à ce que les données de l'organisation soient sécurisées.
Un CASB permet de sécuriser les données de bout en bout, depuis le Cloud au périphérique.
Le Cloud Access Security Broker offre de nombreux services :

- Visibilité sur l'utilisation des applications cloud de l'entreprise et détection du shadow IT.
- Analyse des comportements utilisateurs(UEBA).
- Contrôle des accès utilisateurs.
- Conformité : application des politiques de sécurité et aide à la mise en conformité avec le

RGPD[27].

- Alerte sur les menaces de sécurité.

- Détection des malwares[28].

En pratique, la plateforme CASB s'installe sur un Cloud privé ou dans l'infrastructure des hébergeurs.

B. CloudLock Collaboration Security

CloudLock Collaboration Security permet de :

- Créer une sécurité de l'information aux entreprises

- Protéger les utilisateurs, les données et les applications dans le cloud.

- Combattre plus facilement les violations de données tout en respectant lesexigences réglementaires

C. VMware NSX Cloud

VMware NSX est la plate-forme de virtualisation de réseau destinée au Software-Defined Data Center (SDDC)[29] qui propose un modèle opérationnel de machine virtuelle pour le réseau dans son ensemble.

Grace à NSX Cloud, on peut adopter plusieurs Cloud en toute confiance tout en faisant évoluer les opérations de Cloud, ce qui permet de réduire les dépenses opérationnelles de l'infrastructure de Cloud, et de favoriser la productivité et l'innovation.

[27]RGPD : est l'initiale de Règlement Général pour la Protection des Données et désigne la dernière directive européenne concernant les données personnelles.
[28]malwares : signifiant nuisible / Malveillant. Est un logiciel pouvant être un virus, vers, spywares, keyloggers, chevaux de Troie, backdoors
[29]SDDC : Un centre de données défini par logiciel est un terme marketing qui étend les concepts de virtualisation, tels que l'abstraction, la mutualisation et l'automatisation, à l'ensemble des ressources et des services du centre de données afin de réaliser l'informatique en tant que service.

D. Elastica CloudSOC

C'est une plate-forme pour les applications cloud qui fournit un cycle de vie complet de la sécurité pour SaaS, avant, pendant et après un incident. Elle permet aux organisations de protéger leurs ressources et données, elle prend en charge la détection et la correction des risques, y compris ceux liés aux informations personnelles identifiables.

II.5.7.Les moyens juridiques et contractuels

Dans cette section, nous allons présenter les aspects qui paraissent judicieux à connaître en termes de conformité règlementaire, de standards et de certifications, dont les organisations devront porter leur attention avant de se lancer dans l'utilisation de solutions d'informatique en nuage.

II.5.7.1. Les règlementations*(Boisaubert, 2017)*

Au niveau des règlementations françaises et européennes, nous pouvons citer par exemple :

Le Référentiel Général de Sécurité (RGS) au niveau français, fixant les règles de sécurité pour les échanges numériques avec et/ou entre les autorités administratives. Actuellement, il s'agit de la version 2.0 de juin 2014 qui est applicable depuis le 1er juillet 2014.

Le règlement Electronic IDentification Authentication and trust Services (eIDAS) au niveau de l'Union Européenne, imposant un socle sécuritaire commun sur l'identification électronique et les services de confiance pour les transactions électroniques au sein du marché intérieur entre les citoyens, les entreprises et les autorités administratives de l'Union. Actuellement, il s'agit du règlement eIDAS 910/2014 du 23 juillet 2014 qui est appliqué.

L'arrêté du 3 novembre 2014 relatif au contrôle interne des entreprises du secteur banque et finance (qui remplace le règlement du Comité de la Règlementation Bancaire et Financière (CRBF) 97-02 de février 1997) imposant à ces organisations des exigences en

termes de protection de l'information, dont notamment en cas d'externalisation de leurs activités chez un prestataire.

Au niveau des règlementations applicables en dehors du sol européen ou au niveau mondial, nous pouvons évoquer entre autre :

La loi américaine Sarbanes-Oxley (SOX) de 2002, impose des règles strictes sur la comptabilité et la transparence financière à toute société cotée aux Etats-Unis et à toute filiale d'une société étasunienne cotée sur le sol américain. À noter que l'article 404 de cette loi (Management Assessment of Internal Control) oblige ces organisations à appliquer des règles strictes sur leurs systèmes d'information ;

Les règlementations américaines Health Insurance Portability and Accountability Act de 1996 (HIPAA) **et Health Information Technology for Economic and Clinial Health de 2009 (HITECH)** imposent aux établissements étasuniens de santé des directives afin d'assurer la protection des dossiers médicaux électroniques contre les cyberattaques

En termes de règlementations spécifiques aux données personnelles, nous pouvons lister par exemple :

Le Règlement Général sur la Protection des Données (RGPD) d'avril 2016 applicable depuis mai 2018 visant à la protection des informations personnelles des ressortissants de l'Union Européenne, quel que soit la localisation géographique de la donnée

II.5.7.2. Les normes existantes et les certifications*(Boisaubert, 2017)*

Il existe un grand nombre de normes, de référentiels et de certifications en rapport avec l'informatique en nuage.

A. Les normes ISO :

ISO/CEI 17788 (Information technology – Cloud computing – Overview and vocabulary) : norme définissant les concepts du cloud, les types d'acteurs, les modèles de services et de cloud (2014).

ISO/CEI 17789 (Information technology – Cloud computing – Reference architecture) : norme donnant les définitions de l'architecture fonctionnelle de référence pour élaborer une plateforme d'informatique en nuage (2014).

ISO/CEI 27017 (Information technology – Security techniques – Code of practice for information security controls based on ISO/IEC 27002 for cloud services) : norme indiquant un ensemble de bonnes pratiques sur la sécurité des services cloud (2015).

ISO/CEI 27018 (Information technology – Security techniques – Code of practice for protection of personally identifiable information (PII) in public clouds acting as PII processors) : norme fixant les règles de sécurité pour les fournisseurs de cloud public (2014). Elle s'appuie sur les éléments de la norme ISO 29100 et complète celle de l'ISO 27001.

B. Les recommandations de l'Union Internationale des Télécommunications (International Telecommunications Union – ITU) relatives à l'informatique en nuage, seront classées entre Y.3500 et Y.3599. En voici quelques-unes :

Y.3500 (Information technology – Cloud computing – Overview and vocabulary) : recommandation sur les concepts du cloud, les types d'acteurs, les modèles de services et de cloud (2014) ;

Y.3501 (Cloud computing framework and high-level requirements) : recommandation sur le cadre et les exigences applicables à l'informatique en nuage (2014).

Y.3502 (Information technology Cloud computing Reference architecture) : recommandation sur les définitions de l'architecture fonctionnelle de référence pour élaborer une plateforme d'informatique en nuage (2014).

Y.3512 (Functional requirements of Network as a Service) **:** recommandation sur les exigences fonctionnelles applicables au Réseau en tant que Service (2014).

C. **Les normes développées par l'Americain Institute of Certified Public Accountants (AICPA),** contribuant à renforcer la confiance dans les prestations de service des fournisseurs cloud.

D. **Le référentiel SecNumCloud** de l'Agence Nationale de la Sécurité de l'Information(ANSSI) visant à la qualification des prestataires de service d'informatique en nuage.

E. **La certification Cloud Confidence**

F. **La certification sur l'Hébergement des Données de Santé à caractère personnel**

G. …etc.

II.5.7.3. La contractualisation avec son fournisseur cloud

L'établissement d'un contrat clair entre une organisation cliente et son fournisseur de services d'informatique en nuage est primordial, qui permet notamment d'apporter les garanties nécessaires et suffisantes dans le cadre de la protection de ses données.

Nous pouvons citer par exemple :

Le contrat **SLA (Service Level Agreement) :**« entente de niveau de service » est un document qui définit la qualité de service, prestation prescrite entre un fournisseur de service et un client. Autrement dit, il s'agit de clauses basées sur un contrat définissant les objectifs précis attendus et le niveau de service que souhaite obtenir un client de la part du prestataire et fixe les responsabilités.(Robert Jaques, 2012)

Le SLA tend à devenir un outil essentiel aux clients souhaitant bénéficier d'une sécurité infaillible sur certains de leurs niveaux de sécurité de stockage ainsi que sur la gestion de

leurs données à caractère personnel. De nombreux indicateurs doivent être définis, analysés et contrôlés afin que la performance proposée par le prestataire soit maximisée.

- La disponibilité de l'infrastructure et des services
- La spécification des responsabilités de chacun
- La conformité légale pour la protection et la confidentialité des données
- La restauration des données

Conclusion

La grande diversité des attaques auxquelles des personnes malintentionnées peuvent avoir recours afin de paralyser, infiltrer ou même corrompre un serveur Cloud prouvent que malgré l'évolution des mesures de sécurités établies par des fournisseurs de services tel que Amazon, Google ou Microsoft, et malgré les nombreuses avancées qui se font dans le domaine, l'erreur humaine reste un facteur important.

Il est donc essentiel d'avoir recours régulièrement à des contrôles de sécurité et être capable de protéger les données en elles-mêmes même en cas d'intrusions au moyens de cryptages afin de s'octroyer un niveau de sécurité supplémentaire.

Dans ce chapitre, nous avons pu découvrir les différentes menaces de sécurité qui pèsent sur un Cloud, ainsi que les mécanismes de défense qui peuvent être utilisés pour se protéger de ces menaces.

Dans notre étude, nous allons concevoir deux infrastructures Cloud afin de pouvoir déployer une machine virtuelle. Nous allons également découvrir la solution OpenNebula et la solution OpenStack et ce qu'elles proposent comme moyens de mettre en place ces infrastructures.

Chapitre III : Mise en place des solutions OpenStack et OpenNebula

Introduction

Vu la nécessité de la mise en place d'une solution Cloud Computing appropriée pour but de construire notre infrastructure, qui permet de gérer le provisionnement de machines virtuelles pour fournir un nuage Infrastructure-as-a-service, plusieurs solutions sont disponibles que ce soit des solutions propriétaires ou Open Source.

Dans ce chapitre nous allons nous baser sur deuxsolutions Open Sourcegratuites à savoir OpenNebula et OpenStack. Nous allons définir chaque plateforme et présenter leur fonctionnement ainsi que leurs architectures.Nous nous intéressons, aux principales étapes d'installation de ces deux solutions open source.

I. OpenNebula

Figure 1: logo OpenNebula*(julien, 2019)*

I.1.Présentation d'OpenNebula

Le projet **OpenNebula**est une plate-forme de cloud computing open source qui fournit une solution **IaaS** et prend en charge tous les types d'environnements cloud.

Le projet a été lancé en 2005. Ecrit en C++, C, Java, Ruby[30], Lex[31], et yacc[32], la première version stable est sortie en 2008, sous licence Apache[33]2.0.

OpenNebula gère le stockage, le réseau, la sécurité et les technologies de virtualisation afin de permettre la mise en place dynamique de services multi niveaux (des groupes de machines virtuelles interconnectées) sur les infrastructures distribuées.

Elle offre aux utilisateurs et aux administrateurs l'interfaçage avec des nuages existants (Amazon EC2 Query, Open Grid Forum, Open Cloud Computing Interface et vCloud) et des hyperviseurs Xen, KVM et Vmware.

OpenNebula est intégré dans Debian[34], Ubuntu[35] et OpenSuse[36].

I.2. Architecture OpenNebula

OpenNebula est un gestionnaire qui permet de centraliser l'infrastructure virtuelle. Elle adopte une architecture classique dite de « cluster » avec un **Front-End** et un ou plusieurs **nœuds** qui exécutent et hébergent les machines virtuelles, avec un réseau physique reliant le Front-End aux nœuds.

[30]Ruby : est un langage de programmation libre. Il est interprété, orienté objet et multi-paradigme

[31]Lex : est un programme informatique qui génère des analyseurs lexicaux et a été écrit par Mike Lesk et Eric Schmidt.

[32]Yacc : est un outil de génération d'analyseurs syntaxiques en langage C ,est l'acronyme de Yet Another Compiler Compiler (« Encore un autre compilateur de compilateur ») et il est notamment utilisé dans la construction des jeux d'instructions en langage machine pour les microprocesseurs

33 licence Apache :est une licence de logiciel libre et open source. Elle est écrite par l'Apache Software Foundation, qui l'applique à tous les logiciels qu'elle publie

[34]Debian est une organisation non commerciale qui développe des systèmes d'exploitation basés sur les logiciels libres (modifiables facilement et légalement) de type GNU ou Linux.

[35]Ubuntu est un système d'exploitation (tel Windows ou Mac- OS) basé sur Debian GNU/Linux et sponsorisé par la société Canonical. Ubuntu est une distribution GNU/Linux, c'est-à-dire un regroupement de logiciels libres qui forment un tout cohérent, modulable et adapté à l'utilisateur.

[36]OpenSUSE est une distribution Linux majeure d'origine allemande. C'est une distribution communautaire soutenue par SUSE et d'autres sponsors.

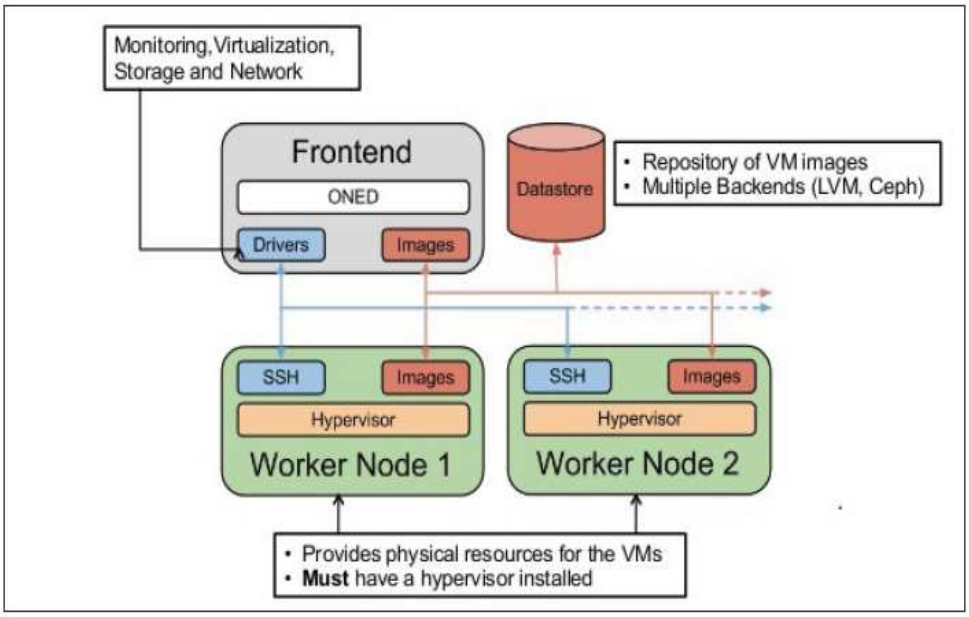

Figure 2: Architecture OpenNebula*(ABDELFATTAH, 2016)*

I.2.1. Front-End

Front-End est le composant central de l'architecture d'OpenNebula qui gère l'ensembledes nœuds de l'infrastructure et qui exécute les services du cluster. Il est composé de:

A) Démon OpenNebula (ONED) : il permet de gérer tous les services du cloud et d'orchestrer les opérations de tous les modules:

– Il gère le cycle de vie des VMs ainsi que le fonctionnement de l'hyperviseur.

– Gère les réseaux virtuels.

– Déploie des machines virtuelles, selon la politique d'attribution, il décidel'emplacement du lancement des machines virtuelles.

– Gère les images des machines virtuelles et leur stockage.

– Supervise l'état des machines virtuelles lancées et leur consommation.

B) Drivers (pilotes): programmes utilisés par le processeur pour l'interfaçage avec un sous-système spécifique, par exemple un système de stockage de fichiers.

Il existe trois (03) types de drivers:

– Pilotes de virtualisation (VMM): interfaçage avec les hyperviseurs(KVM, Vmware ou Xen).

– Pilotes de transfert (TM): interfaçage avec le système de stockage desimages.

– Pilotes d'informations(IM): utilisés pour surveiller les nœuds du cluster.

I.2.2. Nodes (nœuds):

Les nœuds fournissent les ressources nécessaires pour les machines virtuelles, se sont les machines physiques qui hébergent (exécutent) les machines virtuelles, c'est à dire des serveurs hôtes de virtualisation. Chaque nœud dispose des trois élémentssuivants:

A) **Un hyperviseur:** il permet la virtualisation de plusieurs machines virtuelles sur une seule machine physique.

B) **Un bridge:** il permet de relier les interfaces réseaux virtuelles des VMs à l'interface réseaux physique de la machine hôte.

C) **Un serveur SSH:** OpenNebula utilise SSH pour copier les images des machines virtuelles.

I.2.3. Image repository (Référentiel d'images)

C'est le support de stockage qui contient les images de base des machines virtuelles.
En général, il s'agit d'un répertoire de fichiers NFS[37]auquel l'administrateur etl'utilisateur peuvent accéder.

[37] NFS : (Network File System): est un système de fichier distribué permettant à un utilisateur d'accéder à des fichiers sur un réseau d'une manière similaire à la façon dont le stockage local est accessible.

I.3. Caractéristiques :

I.3.1. Gestion des images :

OpenNebulapermet à l'administrateurd'utiliser le système de fichier de son choix sur le systèmede stockage de son choix.

OpenNebuladistingue plusieurs instances de stockage en leur attribuant différent rôles:

- **Les Filesystem Datastore :**ont la responsabilité du stockage des images "template".

- **Les System Datastore :** il est chargé de maintenir une copie des images des VMs en cours d'exécution. Ce dernier permet la migration live d'une VM de son hôte initial vers un nouvel hôte, c'est à dire la relocalisation d'une machine virtuelle d'une machine physique sur une autre sans interruption pour l'utilisateur.(KHENOUS, 2013)

I.3.2. Réseau et adressage :

OpenNebula ne gère pas de serveur DHCP en interne. Il propose alors deux modes:

1) Un mode où il ne gère que les adresses MAC, et il faut alors dans ce cas faire intervenir un serveur DHCP externe permettant l'attribution d'adresses IP aux VMs.

2) Un mode où il propose une règle d'attribution de l'IP à partir de l'adresse MAC, et cette attribution se fait entre l'hyperviseur et la VM.

Pour la mise en place de fonctionnalités réseaux avancées (comme l'isolement de VMs au sein de réseaux virtuels par exemple), OpenNebula propose un système de drivers devant être installé sur les machines hôtes.

Exemples de drivers proposés:

- **fw:** permet l'utilisation de règles simple de pare-feu.

- **802.1Q, VMware, ovswitch, ebtables**: technologies diverses permettant la gestion des ponts entre la connexion physique de la machine hôte et les NICs[38] virtuels des VMs.

I.3.3. Stockage

OpenNebula dispose d'un sous-système de stockage supportant une configuration backend avec différents types de banques de données:

- **Banques de données du système de fichiers:** pour stocker des images disque dans des systèmes de fichiers partagés (NFS, GlusterFS[39]).

- **ISCSI[40]/LVM:** pour stocker des images disque sous une forme périphérique bloc.

- **VMware:** banque de données spécialisée pour l'hyperviseur VMware qui gère le format VMDK[41].

I.3.4. Sécurité

Les mécanismes d'authentification et d'autorisation avec OpenNebula sont basés sur les mots de passe, paires de clés SSH RSA[42], certificats X509[43], LDAP[44]ou Active Directory[45].

38 NIC : (Network Interface Controler): également connu sous le nom d'une carte d'interface de réseau, la carte réseau et adaptateur de réseau local, est un matériel informatique qui relie un ordinateur à un réseau informatique.

39GlusterFS: un système de fichiers libre distribué en parallèle, qui permet de stocker jusqu'à plusieurs pétaoctets (1015octets).

40ISCSI(Internet Small Computer System Interface): un protocole de stockage en réseau basé sur le protocole IP, utilisé pour faciliter les transferts de données sur les intranets et de relier les installations de stockage de données.

41 VMDK (Virtual Machine Disk): un format de fichier ouvert permettant de simuler un disque dur virtuel pour les machines virtuelles telles que VMware, Virtualbox.

42 RSA : RSA est un système cryptographique, ou cryptosystème, pour le chiffrement à clé publique. Il est souvent utilisé pour la sécurisation des données confidentielles, en particulier lorsqu'elles sont transmises sur un réseau peu sûr comme Internet.

43 Certificats X509 : c'est un fichier texte, transmis par l'autorité de certification sur l'ordinateur de client. Il contient la clé publique de l'utilisateur et la signature de la CA ainsi que différents champs normalisés.

44 LDAP (Lightweight Directory Access Protocol): Protocole permettant d'accéder et de gérer des annuaires (les bases d'informations sur les utilisateurs d'un réseau) par le protocoles TCP/IP.

45 Active Directory : permet de fournir des services centralisés d'identification et d'authentification à un réseau d'ordinateurs utilisant le système Windows.

OpenNebula propose ainsi une fonctionnalité de gestion de différents VLANs. Il permet en effet de mettre en place de nombreux procédés visant à sécuriser sonutilisation (tunnels de transferts, authentifications encryptées, séparation des VLANs...)

II. la solution OpenStack

Figure 3: Le logo OpenStack*(Libre, 2012)*

II.1. Présentation d'OpenStack

OpenStackest une plateforme logicielle gratuite et Open Sourcequi permet la construction de tout typede Cloud principalement déployée sous forme d'Infrastructure en tant que service (IaaS) sous licence Apache2.0, écrit en Python, grâce à laquelle des serveurs virtuels et d'autres ressources sont mis à la disposition des clients.

La plateforme est constituée des composants interdépendants qui contrôlent divers pools de ressources de traitement, de stockage et de mise en réseau dans un centre de données. Les utilisateurs la gère via un tableau de bord(Dashboard) ; via des outils de ligne de commande ou via des services web RESTful[46].

OpenStack est packagé dans Ubuntu Server et dans Debian. D'autres distributionstelles que RedHat[47] Enterprise Linux et CentOS[48] sont maintenant supportées.

46 RESTfull : est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web.

47 Redhat : est une distribution Linux produite par Red Hat et orientée vers le marché commercial et les serveurs d'entreprise

48 CentOS :est une distribution Linux qui repose essentiellement sur Red Hat.

OpenStack supporte la plupart des solutions de virtualisation du marché: ESX, Hyper-V, KVM, LXC, QEMU, EMU, Xen et XenServer.

II.2. Architecture OpenStack

OpenStack est composé d'un ensemble de services corrélés dont chaque service a une tâche bien précise.

En effet, ces services sont reliés les uns aux autres autour de deux services(Horizon et Keystone) qui communiquent avec tous les autres modules.

Les services sont pratiquement indépendants mais nécessitent de communiquer ensemble pour assurer la disponibilité, la scalabilité et la sécurité d'une infrastructure OpenStack.

La figure suivante montre une vue schématisée et simplifiée de l'architecture d'OpenStack :

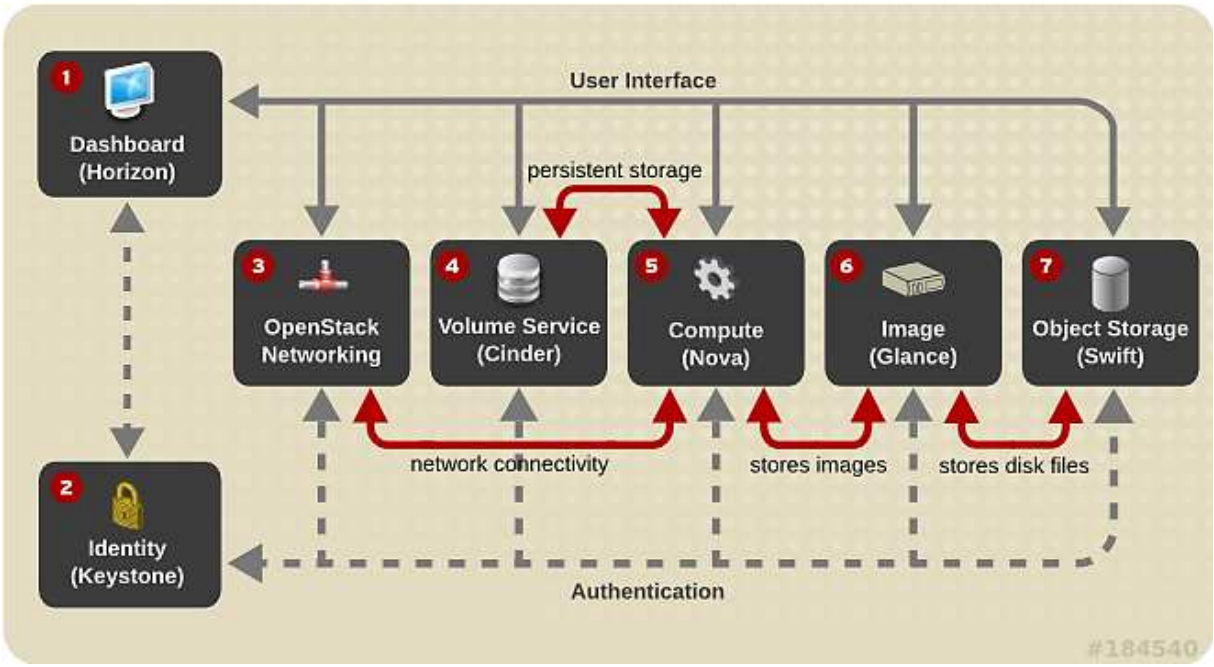

Figure 4: Schéma d'architecture conceptuelle*(DÉON, 2015)*

II.2.1. Nova :

Nova également connu sous le nom d'OpenStack Compute est développé pour fournir un accès à la demande aux ressources informatiques en provisionnant et en gérant de grands réseaux de machines virtuelles. Il gère le cycle de vie des instances serveurs et les tâches incluent la planification, la création et la mise hors service de machines virtuelles à la demande.

Il comporte plusieurs sous-modules ayant chacun une fonction bien précise :

a) **NOVA-API :** Ce démon[49] gère les appels API de l'utilisateur.Ilsupporte l'API native d'OpenStack ainsi que l'API EC2 d'Amazon. Il initie également le démarrage des machines virtuelles et vérifie que certaines règles sont bien respectées (quotas).

b) **NOVA- SHEDULER :** Ce composant s'occupe de récupérer les demandes de création de machines virtuelles en queue et de déterminer sur quelle machine hôte chaque nouvelle instance doit s'exécuter. Le gestionnaire de queue est un système centralisé pour passer les messages entre les démons.

c) **NOVA-COMPUTE :** Ce démon tourne sur les serveurs hôtes. Il gère le cycle de vie des machines virtuelles via l'API de l'hyperviseur.

d) **NOVA-VOLUME :** Ce composant gère la création, l'attachement et le détachement de volumes persistants.

e) **NOVA-NETWORK:**Ce composant de NOVA gère les réseaux (configure les interfaces bridge, adapte les règles de pare-feu…).

f) **QUEUE :** C'est le point de passage obligé pour les instructions échangées entre les services, il existe différents types de files d'attente de messages pour faciliter la communication : Topics, Fanout, Host…

g) **Data Base:** Enregistre la configuration et les états en temps réels pour une infrastructure Cloud : types d'instances disponibles, instances en cours d'utilisation, réseaux disponibles, projets… supporte la plupart des SGBD : MySQL, PostgreSQL.

[49] Démon : un programme informatique ou un processus qui n'est pas contrôlé par l'utilisateur et qui s'exécute en arrière-plan.

II.2.2. Horizon

OpenStack fournit un tableau de bord (Dashboard) qui s'appelle Horizon. Il s'agit d'une application web qui permet aux utilisateurs et aux administrateurs de gérer leurs clouds à travers une interface graphique. Chaque utilisateur authentifié peut déployer une machine virtuelle, assigner des adresses IP, gérer le contrôle d'accès, ajouter une image, créer un volume, etc.

II.2.3. Cinder Block Storage

Le composant OpenStack Cinder a pour rôle de gérer le stockage permanent. Il permet ainsi de créer, modifier et supprimer les volumes et de gérer les types de volumes et les snapshots.

Il repose sur plusieurs briques :

a) **Cinder-api :** ce démon accepte les requêtes API et les transmet à cinder-volume pour l'exécution.
b) **Cinder-volume :** c'est le cœur de cinder. Il réceptionne les demandes de cinder-api et interagit avec la base de données et les autres processus.
c) **Cinder-scheduler :** sélectionne le stockage le plus adapté pour créer le volume.

II.2.4. Swift Object Storage

Stocke et récupère des objets de données non structurées via une API RESTful basée sur HTTP. Le service est hautement tolérant aux pannes avec sa réplication de données et son architecture de type scale-out[50]. Le service écrit les objets et les fichiers vers plusieurs disques, en s'assurant que les données sont répliquées sur un cluster de serveurs.

[50] Scale-out : est une architecture de stockage rattaché au réseau (NAS, Network Attached Storage) dans laquelle il est possible d'augmenter la quantité d'espace disque totale en ajoutant des unités à des baies reliées entre elles et dotées de leurs propres ressources.

II.2.5. Neutron Networking

C'est le module gérant le réseau en tant que service. Ce service permet aux utilisateurs de créer et gérer leur propre réseau et de se connecter à différents types d'architectures grâce aux différents plugins :Brocade, Jupiner, Linux Bridging… etc.

II.2.6. Keystone Identity

Keystone est le service de gestion des identités et des autorisations d'accès, c'est-à-dire qu'il fournit tout le mécanisme pour gérer les utilisateurs, les tokens et le catalogue de droits pour tous les composants d'OpenStack.

Ce module est sollicité par exemple lorsqu'un utilisateur souhaite se connecter sur le portail, lorsqu'un utilisateur souhaite provisionner une machine virtuelle.

Voici les concepts clés de KeyStone :

User : peut-être une personne, un système ou un service. Un utilisateur dispose obligatoirement d'un login, et peut se voir attribuer un Token pour accéder à une ressource. Un utilisateur peut être affecté à un ou plusieurs Tenants/Projets.

Les Credentials : les données qui permettent d'authentifier un utilisateur ou un service, cela peut être sous forme de :

- Login et Mot de passe
- Login et une clé
- Token qui vous a été délivré.

  - **Authentication :** Le gestionnaire d'identité d'Openstack confirme qu'une requête est faite par l'utilisateur prétendant avoir fait la demande en validant un jeu de revendications que l'utilisateur effectue. Après une première confirmation keystone génère et délivre un Token que l'utilisateur utilise comme une carte d'identité.
  - **Token :** est un texte arbitraire (chiffres et lettres) qui est utilisé pour accéder aux ressource. Chaque token possède un champ qui décrit les différents services qui lui sont accessible. Un Token a une durée de vie et peut être révoqué à tout moment.
  - **Tenant (ou projet)** : est un conteneur pour grouper les services ou les utilisateurs.

- **Service :** les services Openstack tel que : Compute (Nova), Object Storage (Swift), Image Service (Glance), Netwok (Quantum), Identity (Keystone) … Les services fournissent un ou plusieurs Endpoints.
- **Endpoint :** Une adresse réseau, généralement décrite par une URL, où un service peut être accessible.
- **Role :** Une personnalité qui assume un utilisateur lors de l'exécution d'un ensemble d'opération. Un rôle comprend un ensemble de droits et privilèges. Un utilisateur assumer ce rôle et hérite ces droits et ces privilèges.

II.2.7. Glance Image

Glance est le module qui stocke et récupère des images disques de machines virtuelles.

Il se compose de quatre parties principales :

- **Glance-api** : ce démon traite les appels à l'API pour la gestion des images. Il permet notamment de lister les images disponibles, récupérer une image et créer une nouvelle.
- **Glance-registry** : ce démon traite, stocke et récupère les métadonnées associées aux images(taille, type….).
- **Base de données** : la base de données est utilisée pour stocker les métadonnées.
- **Répertoire de stockage** : c'est dans ce répertoire que les fichiers d'image sont stockés.

II.2.8. Telemetry Ceilometer

Il permet de collecter différentes métriques sur l'utilisation du cloud. Il permet par exemple de récolter le nombre d'instances lancé dans un projet et depuis combien de temps.

Ces métriques peuvent être utilisées pour fournir des informations nécessaires à un système de facturation ; sont aussi utilisées dans les applications ou par d'autres composants d'Openstack pour définir des actions en fonction de certains seuils comme avec le composant d'orchestration.

II.2.9. Heat Orchestration

Heat est le composant d'orchestration d'Openstack. Il permet de décrire une infrastructure sous forme de modèles. Dans Heat, ces modèles sont appelés des stack. Heat consomme ensuite ces modèles pour aller déployer l'infrastructure décrite sur Openstack. Il peut aussi utiliser les métriques fournies par Ceilometer pour décider de créer des instances supplémentaires en fonction de la charge d'une application par exemple.

II.4. Caractéristiques

II.4.1. Gestion de l'authentification et autorisation :

Lorsqu'un utilisateur se connecte via Horizon, ce dernier envoie une requête HTTP à Keystone pour lui demander s'il peut autoriser la connexion ; Keystone valide les points suivants :

- L'authentification : est-ce que les credentials sont valides ?
- Le contrôle d'accès : l'utilisateur existe-t-il ? Fait-il partie d'un tenant (projet)? Dispose-t-il d'un rôle ?
- L'autorisation : quels sont les droits d'accès de l'utilisateur ?

II.4.2. Gestion des images :

Nova Compute demande à Glance l'image de la VM, il lui fournit pour cela le numéro de l'image (image_id).

Nova Compute dialogue alors avec l'API de Glance, plus exactement via la brique glance-api.

Glance-api retourne une réponse à Nova Compute si l'image existe.

II.4.3. Gestion du réseau :

C'est le module Network (Neutron) qui assure la communication et le déploiement de l'environnement réseau. Neutron exploite les différents SDN[51](Software Defined Network) du marché comme OpenVswitch. Il dispose également d'un plug-in permettant la connexion avec des appliances réseau (firewall, DNS ou IP management).

Neutron fournit une API pour la gestion des topologies réseau dans OpenStack ; en ce sens, Neutron rend autonome OpenStack dans la configuration réseau : les instances créées peuvent disposer d'IP privées et publiques, attribuées de façon automatique. Des routeurs virtuels sont utilisés au sein de l'infrastructure OpenStack.

Neutron utilise les concepts suivants :

- Port : le port représente la carte réseau des machines virtuelles.
- Network : le network représente un réseau virtuel (dans Neutron comme par exemple un tunnel GRE (Generic Routing Encapsulation).
- Subnet : un subnet est un sous-réseau ; il est matérialisé par un adressage spécifique (par exemple, 192.168.1.0/24 représente le subnet ayant une plage d'IP de 192.168.1.1 à 192.168.1.254).
- Router : un routeur dans Neutron est de type virtuel ; il offre une fonction de routage de paquets entre différentes machines de subnets différents.

II.4.4. Gestion du Stockage :

Le stockage est géré par le module Cinder pour la partie stockage en mode bloc et par le module Swift pour le stockage en mode objet.

Le stockage objet (fourni par Swift) a été conçu pour les datastores hautement évolutifs à base d'objets comme les éléments multimédias, les images et les fichiers. La priorité pour ces systèmes est leur capacité à s'adapter à de grandes quantités de données sans dépendre de fonctions de stockage traditionnelles telles que le RAID.

[51] SDN :est un modèle d'architecture réseau qui permet aux administrateurs de réseaux de gérer les services de réseaux par abstraction de fonctionnalités.

Cinder fournit le composant stockage en mode bloc utilisé pour stocker des objets persistants, tels que les VMs et des données régulièrement mises à jour sur place dans des bases de données.

III. La mise en place d'OpenNebula et d'OpenStack

L'objectif de notre étude est la mise en place d'une infrastructure dédiée à la création d'un Cloud privé afin de faire tourner nos machines virtuelles et pouvoir y accéder à distance.

Pour se faire, nous allons dans ce qui suit présenter en détail la mise en place des deux solutions.

III.1. La mise en place d'OpenNebula

Pour commencer, nous allons mettre en place notre hyperviseur KVM (Kernel-based Virtual Machines) qui sera la base de notre Infrastructure afin de pouvoir y héberger nos machines virtuelles.

Nous allons, pour finir, installer OpenNebula afin de pouvoir récupérer et superviser notre hyperviseur KVM.

Une fois notre infrastructure mise en place, nous allons créer une nouvelle machine et y accéder à distance.

III.1.1. Prérequis

Pour pouvoir mettre en place notre Infrastructure, nous aurons besoin de :

Ressources	**Configuration requise**
Mémoire	4 GB
CPU	1 CPU (2 cores)
Stockage	100 GB

Réseau	2 NICS et 1 pont br0
Hyperviseur	VirtualBox
Machine virtuelle	Debian 9,ubuntu 16.04

III.1.2. Les étapes d'installation

1. Installation de la partie Front-end

Pour l'installation de la partie Front-end, nous avons choisi d'exécuter une machine Debian 9 avec un adaptateur réseau NAT ayant l'adresse IP 192.168.117.141.

Sur le terminal, on exécute en tant que super utilisateur :

#wget –q –O- http://downloads.opennebula.org/repo/Debian/repo.key | apt-key add -

#echo ''deb http://downloads.opennebula.org/repo/5.8/Debian/9 stable opennebula'' > /etc/apt/sources.list.d/opennebula.list

#apt-get update

#apt-get install opennebula opennebula-sunstone opennebula-gate opennebula-flow

#usr/share/one/install_gems

Un nouvel utilisateur Oneadmin sera créé. Lui attribuer ensuite un mot de passe Unix et un mot de passe Sunstone afin de pouvoir accéder à l'interface de gestion :

#passwd oneadmin

#echo "oneadmin :motdepasse" >/var/lib/one/.one/one_auth

Il ne reste plus qu'à se connecter au compte oneadmin et lancer le service OpenNebula :

#su – oneadmin

#systemctl start opennebula opennebula-sunstone

2. Installation du nœud KVM :

Le nœud KVM va être installé sur une machine Ubuntu 18.04 avec un adaptateur réseau NAT et l'adresse IP 192.168.117.142

Sur le terminal, exécuter en tant que super utilisateur :

#wget –q –O- http://downloads.opennebula.org/repo/repo.key | apt-key add

#echo ''deb http://downloads.opennebula.org/repo/5.8/Ubuntu/18.04 stable opennebula'' > /etc/apt/sources.list.d/opennebula.list

#apt-get update

#apt-get install opennebula-node

L'utilisateur Oneadmin sera créé et sera ajouté au groupe KVM, ce qui va lui permettre de créer et gérer les machines virtuelles.

Une fois KVM installé, il suffit de lancer le service de virtualisation libvirt-bin

#service libvirt-bin start

Et attribuer un mot de passe à Oneadmin et notre hyperviseur sera opérationnel

#passwd oneadmin

Une fois le Front-End et l'hyperviseur mis en place, il faut connecter les deux machines via un protocole SSH afin qu'OpenNebula puisse récupérer les pools de ressources de KVM.

Pour se faire, il va falloir ajouter notre nœud à la liste des hôtes accessibles par OpenNebula :

#ssh-keyscan 192.168.117.141 192.168.117.142 >> /var/lib/one/.ssh/known_hosts

Puis copier la clé RSA de notre Front-End vers notre hôte afin que les deux machines puissent communiquer entre elles sans avoir à utiliser de mot de passe :

#scp –rp /var/lib/one/.ssh 192.168.117.142:/var/lib/one

Puis vérifier que la clé à bien été mise en place car elle sera le pilier de la communication entre le Front-End et l'hôte

#ssh 192.168.117.141

#ssh 192.168.117.142

Maintenant que le protocole de communication SSH a été mis en place entre les deux machines, il ne reste plus qu'à ajouter l'hôte sur l'interface Sunstone afin de récupérer les informations relatives à notre hyperviseur et pouvoir créer un template et instancier notre machine virtuelle.

Pour se connecter à l'interface Sunstone, on accède à l'adresse***http://192.168.117.141:9869/***puis on introduit le nom d'utilisateur « oneadmin » et le mot de passe.

Figure 5 - Connexion à OpenNebula Sunstone

Il nous faut maintenant ajouter l'hôte KVM que nous avons créé depuis l'interface Sunstone

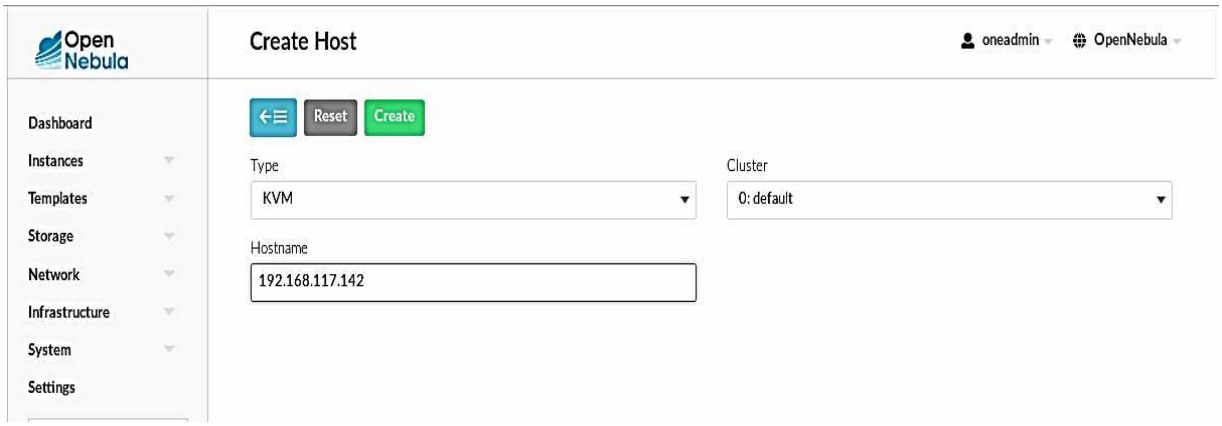

Figure 6 :ajout hôte KVM

Figure 7 : Hôte KVM accessible

On peut à présent créer notre machine virtuelle et configurer les paramètres nécessaires

III.1.3. Les configurations appliquées

Voici les différentes étapes de configuration à suivre afin d'installer OpenNebula :

III.1.3.1. Création d'un template :

Un template sera le modèle sur lequel va se baser OpenNebula pour créer une machine virtuelle. On va y définir les paramètres spécifiques au type de VM que nous voulons déployer.

Notre template va reposer sur quatre paramètres :

1. **Les images :**

Nous devrons créer deux types d'images :

- L'image ISO :

Afin de pouvoir installer un système d'exploitation sur notre machine virtuelle, il va nous falloir créer une image qui sera stockée sur le datastore [52] par défaut d'OpenNebula et en configurer les paramètres :

Dans Storage -> Images, on crée une image ISO comme suit

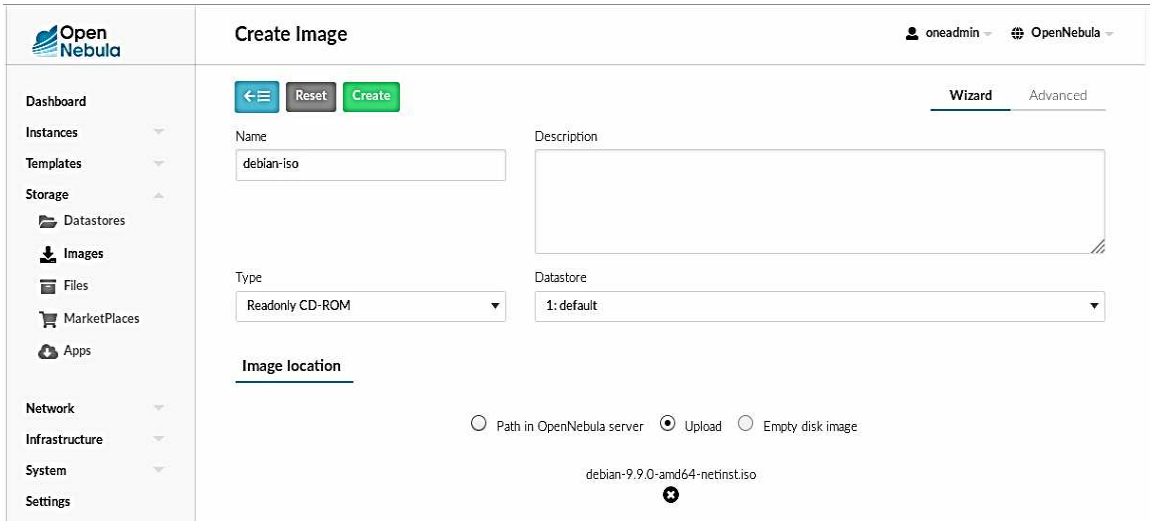

Figure 8 - ajout image iso debian

- L'image du disque :

Il nous faut créer et configurer un disque dur qui sera copié sur l'hôte lors du déploiement de la VM avec les paramètres suivants

[52] **Dtastore** : « depôt de données », est un référentiel servant au stockage permanent d'ensemble de données.

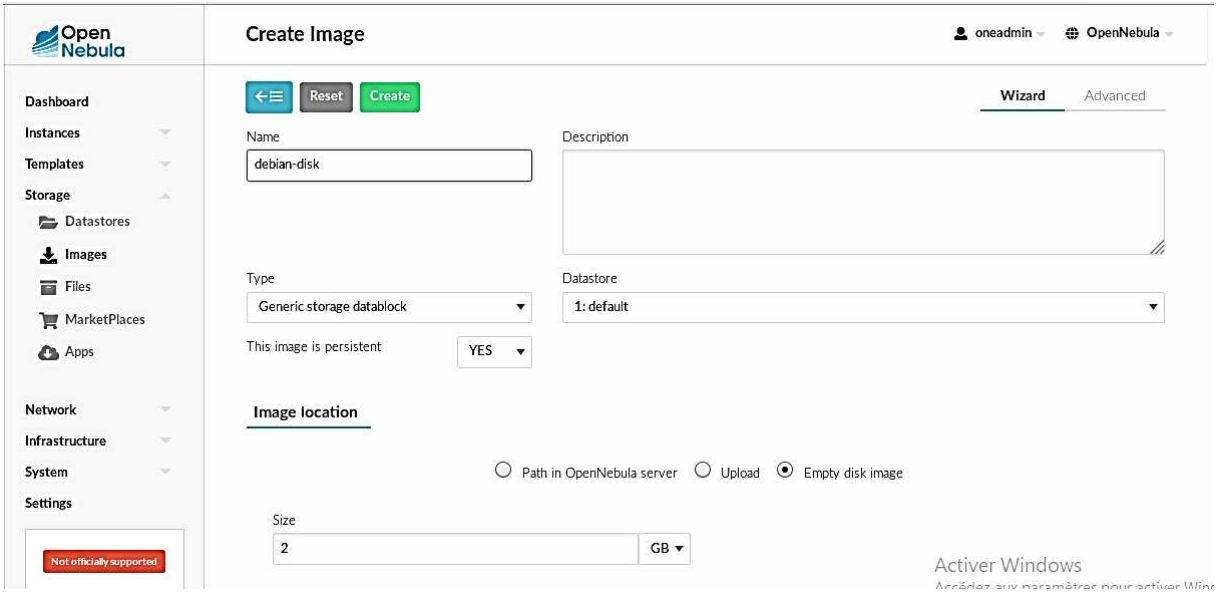

Figure 9 - ajout disque dur debia

Nous obtenons deux disques opérationnels

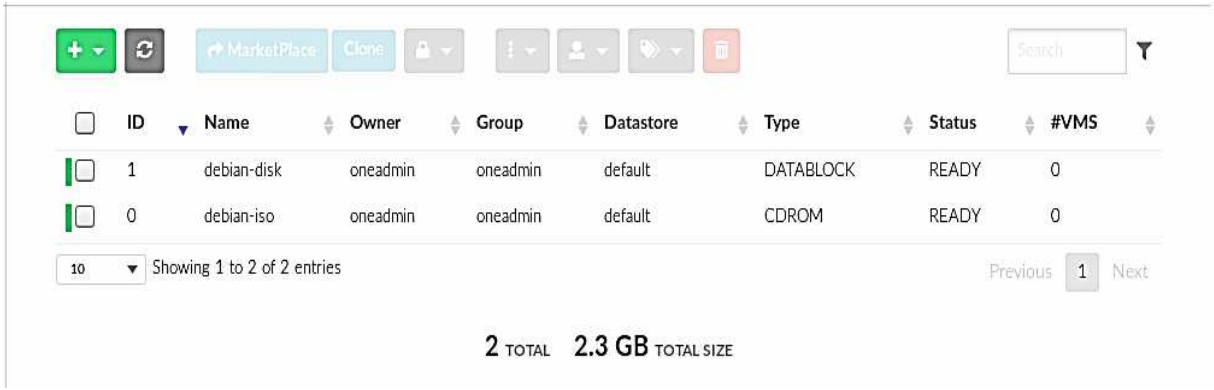

Figure 10 - Images disque dur et disque ISO

2. Le réseau :

Nous voulons que notre VM, lors de son déploiement, soit accessible depuis l'extérieur. Nous avons donc au préalable configuré un pont de connexion br0 sur le système de l'hôte KVM qui va se charger d'exposer notre VM au réseau extérieur.

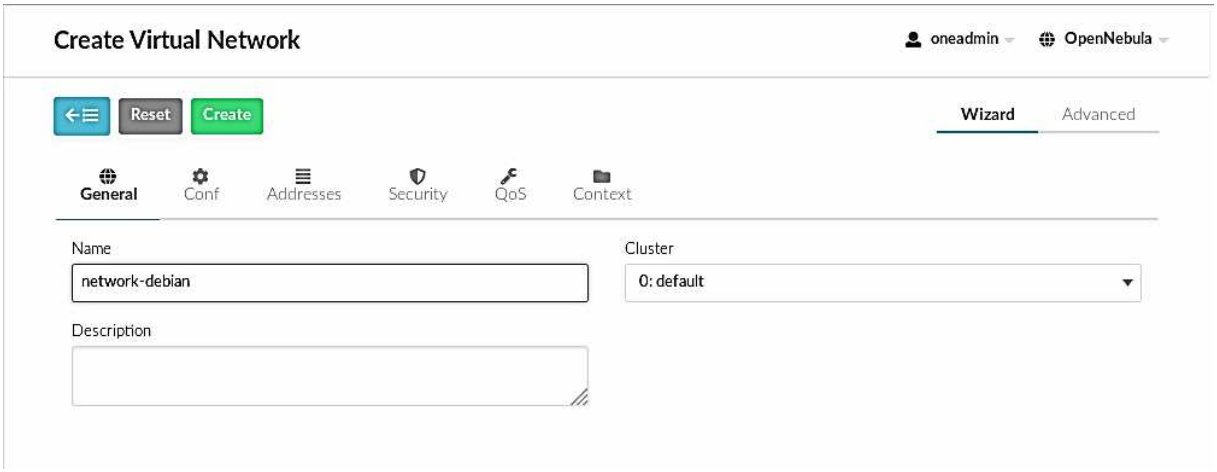

Figure 11 - Ajout paramètres réseau Général

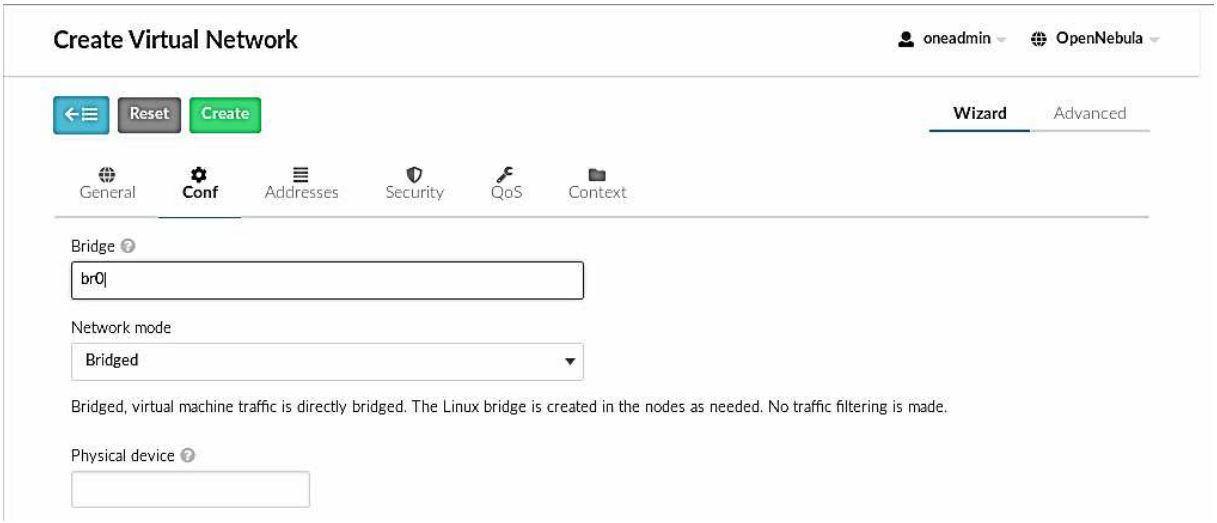

Figure 12 - Ajout paramètres réseau Configuration

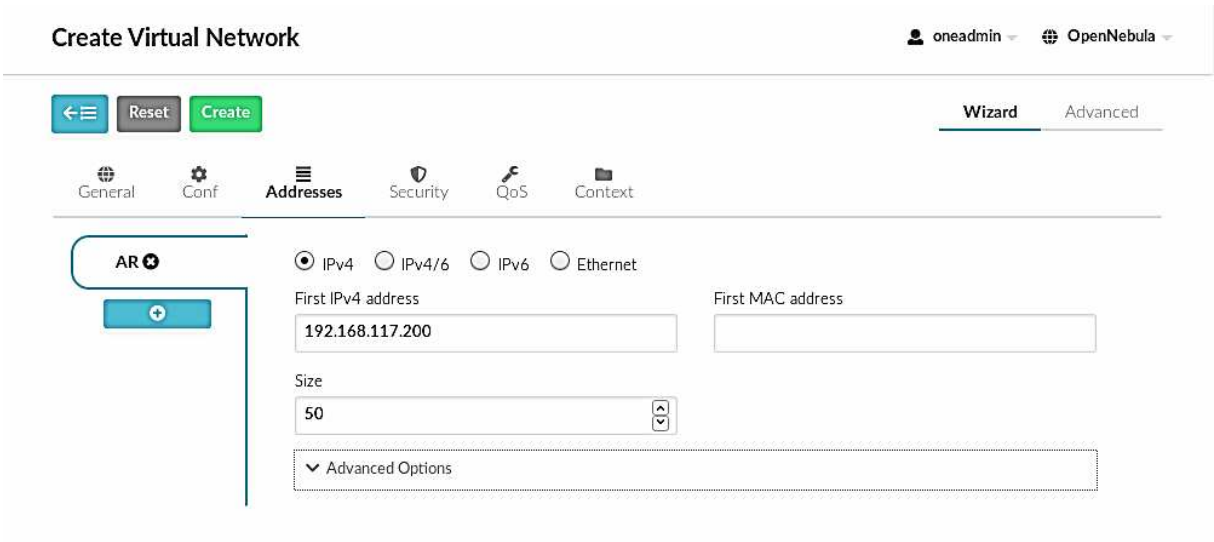

Figure 13 - Ajout paramètres réseau Champ d'Adresses I

3. Créer le template :

Une fois le réseau et le stockage configurés, il ne reste plus qu'à finaliser la création de notre template de base pour la VM.

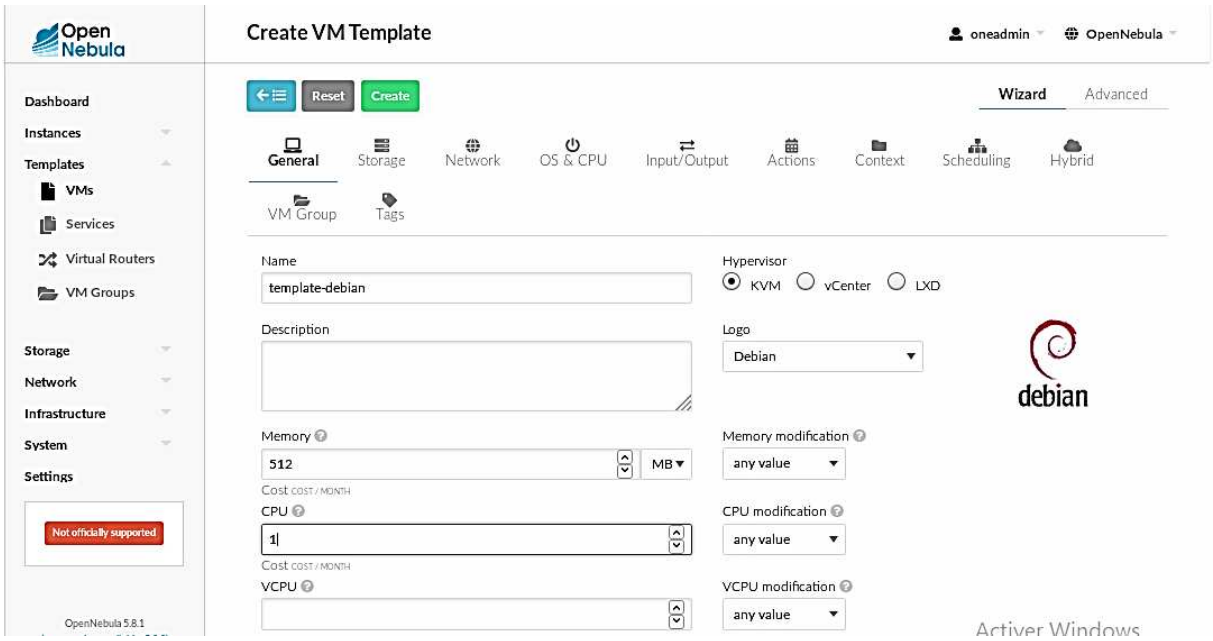

Figure 14 - Création template Général

Nous allons créer deux disques sur le template et leur attribuer respectivement le disque ISO et le disque dur. Le disque ISO devra être en Lecture Seule.

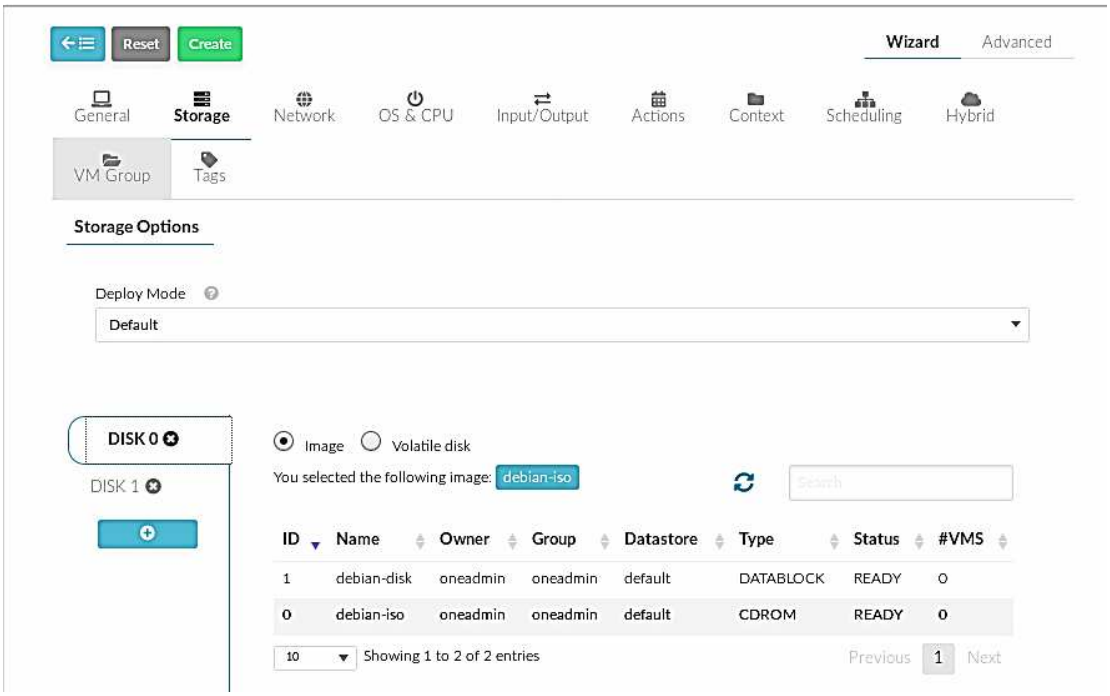

Figure 15 - Configuration stockage template

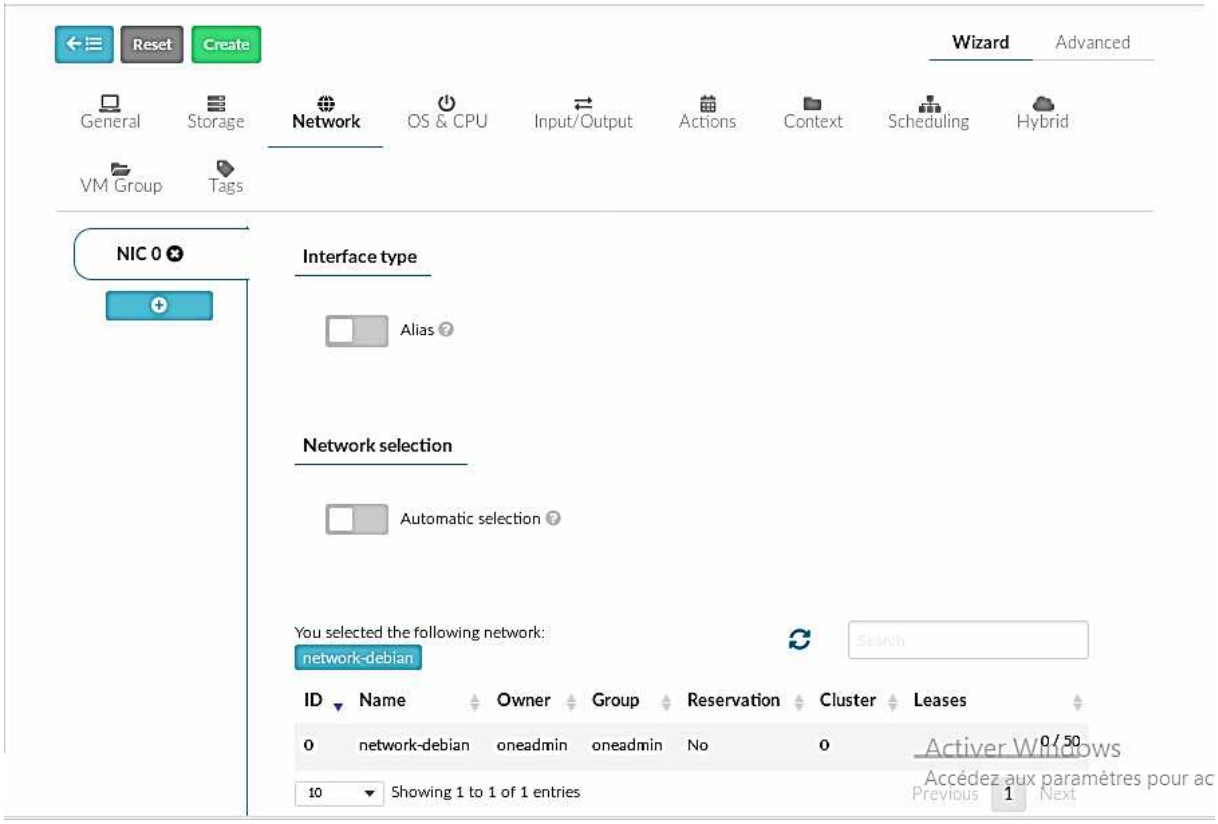

Figure 16 - Configuration réseau template

Il sera également essentiel de configurer l'accès à notre machine depuis l'extérieur via VNC. Nous avons attribué à la VM un port d'écoute ainsi qu'un mot de passe.

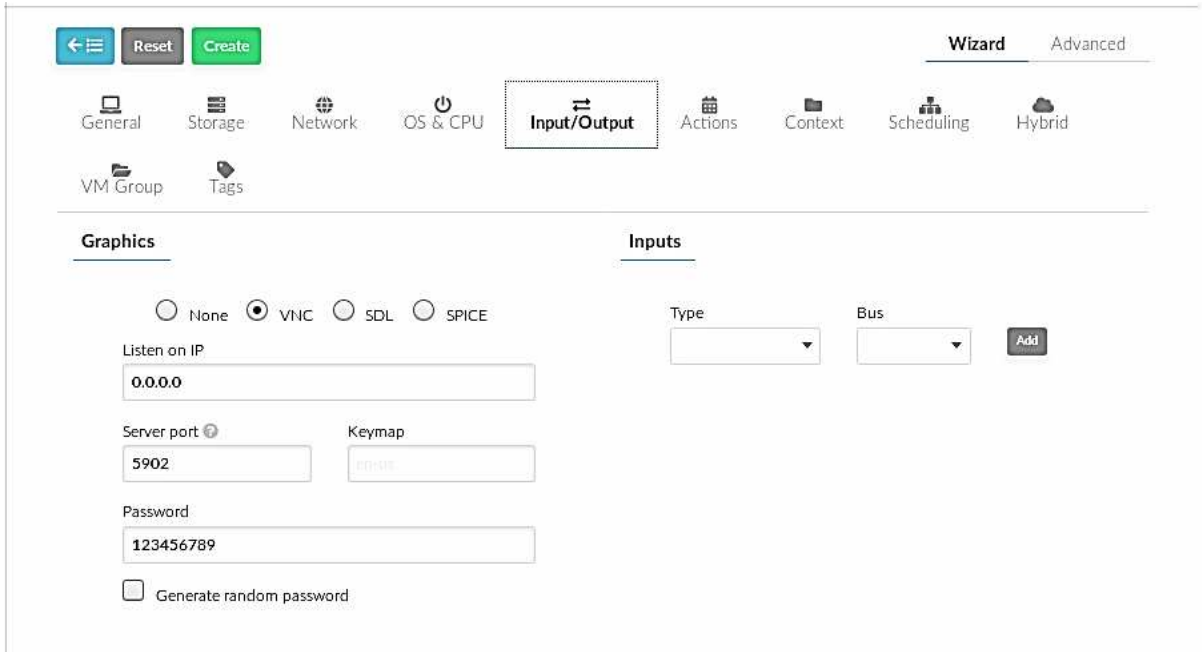

Figure 17 - Configuration VNC template

4. Créer et déployer une VM :

Une fois que le template est créé, il ne reste plus qu'à instancier notre VM.

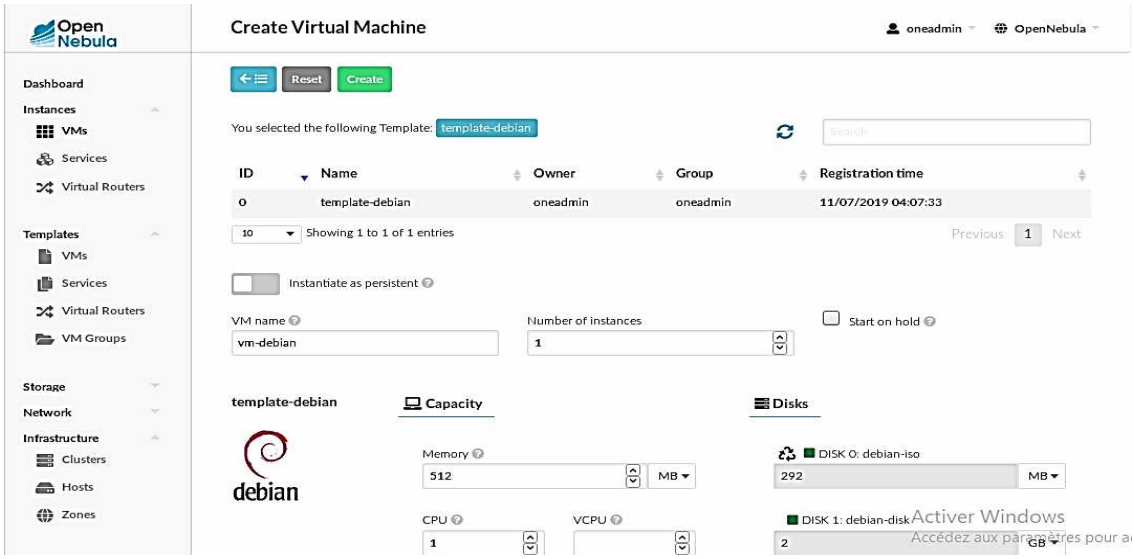

Figure 18 - Création VM

Une fois la VM instanciée, il suffit de la déployer sur l'hôte et d'attendre qu'elle démarre.

Figure 19 - VM déployée et opérationnelle

Notre VM est à présent accessible via l'adresse 0.0.0.0:5902 que nous avons configurés sur le template.

III.2.La mise en place d'OpenStack

Selon la documentation OpenStack, il y a plusieurs architectures de déploiement de cette technologie ; le choix se fit selon les besoins d'utilisation, les ressources disponibles et les exigences des entreprises et utilisateurs.

Le schéma ci-dessous montre les différentes architectures existantes :

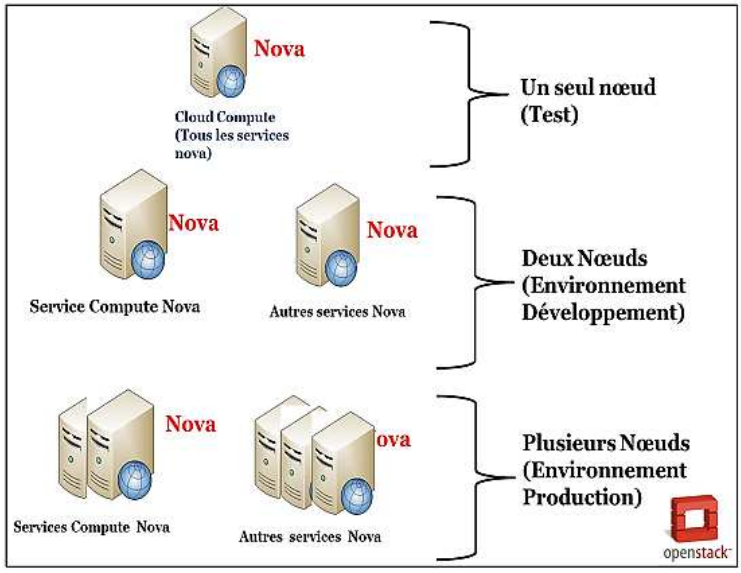

Figure 20: Les différentes architectures possibles*(Mehdi, 2018/2019)*

Comme on le voit bien OpenStack peut s'installer de deux (02) manières :

1. **Un seul Nœud (Single Node) :** dans cette configuration, tous les services sont installés et exécutés sur un système unique. Ce type de déploiement est adapté à des fins d'évaluation. Un tel déploiement n'est cependant pas adapté pour une utilisation dans un environnement de production.

2. **Plusieurs Nœuds (Multi Node) :** les composants d'OpenStack seront installés sur des nœuds différents.

OpenStack propose aussi plusieurs types d'installation :

- Installation via des scripts.
- Installation via les packages.

Dans notre cas nous allons installer OpenStack sur un seul nœud et pour mieux comprendre l'installation des différents composants nous avons donc choisi une installation manuelle via les packages.

III..2.1. Les prérequis

Ressources	Configuration requise
Mémoire	8 GB
CPU	2CPU (4 cores)
Stockage	100 GB
Réseau	2 NICS et 1 pont br0
Hyperviseur	VirtualBox
Machine virtuelle	CentOS7

III.2.3. Les étapes d'installation :

Voici les différentes étapes à suivre afin d'installer OpenStack :

III.2.3.1. Installation des paquets de base et préparation du système

Après avoir terminé l'installation de la distribution CentOS7, nous passons en tant qu'utilisateur root jusqu'à la fin de l'installation:

openstack-VirtualBox: sudo su

1. La mise à jour du système

Avant de commencer l'installation des paquets, la Mise à jour du système est nécessaire :

 # apt-get update

 # apt-get upgrade

 # apt-get dist-upgrade

3. Installation des bridges

Le bridge est nécessaire pour configurer les interfaces Ethernet en mode pont et permet le partage de connexion entre ces interfaces, il permet également de relier une interface réseau physique à une interface réseau virtuelle

Nous allons installer le paquet bridge-utils qui va permettre de créer et gérer les bridges :

 # apt-get install bridge-utils

4. Configuration réseau

Nous allons configurer le réseau de l'environnement openstack par la modification du fichier /etc/network/interfaces comme suit :

#L'interface réseau loopback

Auto lo

Iface loi net loopback

#l'interface réseau promaire

Auto eth0

Iface eth0 inet dhcp

#l'interface public

Auto eth1

Iface eth1 inet static

Address

Netmask 255.255.0.0

Network

Broadcast

#l'interface privée

Auto br100

Iface br100 inet manual

Bridge_ports eth2

Bridge_stp off

Bridge_maxwait 0

Bridge_fd 0

Up ifconfig eth2 up

Nous redémarrons les services réseau pour que ces changements soient pris en charge:

 # /etc/init.d/networking restart

5. Installation et Configuration de MySQL

Chaque composant OpenStack possède sa base de données *MySQL*, contenant toutes les données modifiables à chaud (ID des images disques, des instances virtuelles, réseaux, identités...). Les données de configuration fixes sont stockées dans des fichiers texte:

 # apt-get install mysql-server pythonmysqldb

Pour indiquer à MySQL que le serveur doit écouter sur toutes les interfaces et pas seulement sur la boucle locale, nous allons modifier sa configuration dans le fichier */etc/mysql/my.cnf* comme suite:

 # sed –i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf

Ensuite, nous redémarrons le serveur *MySQL* :

 # service mysql restart

6. Installation et configuration du serveur NTP[53]

NTP (Network Time Protocol) est un programme de synchronisation de l'heure qui permet la bonne synchronisation du cloud:

 # apt-get install ntp

Puis, nous allons configurer le serveur NTP en modifiant le fichier /etc/ntp.conf:

 # sed –i 's/server ntp.ubuntu.com/server ntp.ubuntu.com \nserver 127.127.1.0\nfudge 127.127.1.0 stratum 10/g' /etc/ntp.conf

Une fois la configuration terminée nous redémarrons le service:

 # service ntp restart

7. Installation de RabbitMQ

RabbitMQ est un courtier de messages se basant sur le standard AMQP afin d'échanger les informations avec différents clients. En bref c'est le service qui permet la communication entre les composants d'OpenStack

 # apt-get install rabbitmq-server

III.2.3.1. Installation et configuration des composants OpenStack

Pour la mise en place d'OpenStack, nous allons opter à l'installation des composants:Keystone, Glance, Nova et Dashboard, tel que nous n'allons pas procéder à l'installationde Swift, Quantum et Cinder car la configuration de ces derniers complexifie beaucoup letemps de la mise en place, et nécessitedu matériel performant

De plus, les fonctionnalités avancées de Quantum de Swift et de Cinder ne sont pasvraiment utiles lors de l'installation sur un seul serveur et pour une installation de test.

Pour cela nous avons opté à:

- Installer et configurer nova-network qui prend en charge le comportement de Quantum mais avec moins de fonctionnalités.

[53] Serveur NTP :

- La configuration de nova-volume au lieu du composant Cinder.
- Enfin, l'utilisation de nova-objectstore qui `a un rôle similaire au composant Swift

1. Installation et configuration de Keystone

Le composant Keystone est chargé de la gestion des utilisateurs et des services.

- **La gestion des utilisateurs** : s'articule autour de trois (03) objets:

  - L'objet User représentant l'utilisateur final
  - L'objet Tenant que l'on peut représenter par un projet, une organisation au sein duquel les instances seront regroupées et administrées par les utilisateurs.
  - L'objet Role qui définit le rôle de l'utilisateur sur un Tenant. Un utilisateur peut avoir un ou plusieurs rôles sur différents Tenants.

- **Gestion des services et points d'accès :**
  - La gestion des différents services, comme Glance pour les images ou Swift pourle stockage
  - La définition des points d'accès `a ces différents services, les url et ports pour y accéder

Nous commençons par l'installation du paquet keystone:

 # apt-get install keystone

Après avoir terminé l'installation de keystone nous passons à sa configuration :

- **Création de la base de donnée MySQL**

La création de la base de donnée MySQL pour Keystone se fera par les étapes suivantes:

Lancer la console MySQL par la commande :

 # mysql -u root –p

Puis nous introduisons le mot de passe de l'utilisateur root maysql .

La commande suivante crée une base de données nommée "keystone":

mysqlCREATE DATABASE keystone;

Enfin, nous allons créer un utilisateur nommé ”keystone” avec un mot de passe ”openstack” et lui donner tous les droits sur la base de données ”keystone”:

mysql GRANT ALL ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'openstack';

mysql FLUSH PRIVILEGES;

mysql Quit;

Et nous modifions le fichier */etc/keystone/keystone.conf* comme indiqué ci-dessous:

Nous allons décomenter les huit premières lignes et nous allons rajouter la ligne *”connection =mysql://keystone:openstack@172.16.0.1:3306/keystone”* pour créer une connexion à labase de données *”keystone”*

Ensuite, nous redémarrons keystone:

root@openstack-VirtualBox:/home/openstack# service keystone restart

Et enfin, nous allons synchroniser la base de données *”keystone”:*

root@openstack-VirtualBox:/home/openstack# keystone-manage db sync

- **La création des rôles**

Les rôles pouvant être attribués à un utilisateur sont:

- **Admin:** donne le droit de modifier la configuration des services (ex: allouer uneplage d'adresse IP, un quota d'espace disque pour un projet etc...).
- **Member:**permet de gérer le contenu du projet (création d'instances de machines,ajout d'un disque virtuel à l'une d'elles etc...)

Commençons par l'installation du paquet suivant

 # apt-get install python-keystoneclient

Pour lancer une commande keystone, l'authentification se fait avec les arguments user/password de la manière suivante:

 # keystone _username admin –password ADMIN --tenant name admin --auth url *http://localhost:5000/v2.0*

Ensuite, nous ajoutons la ligne source ~/.openrc à la fin du fichier .bashrc

 # echo ”source ~/ .openrc” .bashrc

1. La création du rôle admin

 # keystone role-create --name admin

2. La création du role member

 # keystone role-create --name Member

3. La création du rôle KeystoneAdmin

 # keystone role-create --name KeystoneAdmin

4. La création du rôle KeystoneServiceAdmin

 # keystone role-create --name KeystoneServiceAdmin

- **La création des tenants**

- **Tenant admin:**permet à ses membres d’administrer les services.

 # keystone tenant-create --name=admin --description ”” --enabled true

- **Tenant service :** le Tenant interne des services

 # kystone tenant-createname=service -description ”” --enabled true

- **Ajout des utilisateurs dans les tenants**

Récupération de l’identificateur du tenant admin :

#TENANTID=$(keystonetenant-listawk ’/admin/{print $2}’)

Création de l’utilisateur admin:

 # kyston user-creat –name admin --tenant id $TENANT ID --pass ADMIN --email root@localhost --enabled true

Récupération de l'identificateur du Rôle admin:

 # ROLE ID=$(keyston rolelist | awk / admin/{print $2}'

Récupération de l'identificateur de l'utilisateur admin**:**

 # USER ID =$(keyston user-list | awk/ admin /{print $2}')

Affectation du rôle admin pour l'utilisateur admin dans le tenant admin:

 # keyston user-role-add –user $user ID –role£ROLEDID-tenante id $TENANT ID

De la même façon nous attribuons le rôle KeystoneAdmin et KeystoneServiceAdmin pour l'utilisateur admin dans le tenant admin, ainsi le rôle admin pour les utilisateurs glance et nova dans le tenant service, après avoir créé les utilisateurs glance et nova.

- **Création des services**

1. Le service keystone

 # keystone service-create --name keystone--type identity --description 'OpenStack Identity Service'

2. Le service Glance

 # keystone service-create --name glance --type image --description 'OpenStack Image Service'

3. Le service nova compute

 # keystone service-create --name nova --type compute --description 'OpenStack Compute Service'

4. Le service nova volume

 # keystone service-create --name volume --type volume --description 'Volume Service'

- **Définition des points d'accès pour les services**

1. Le point d'accès du service Keystone

Récupération de l'identificateur du service de keystone

 # ID=$(keystone service-list | awk '/ keystone / {print $2}')

Nous allons configurer les URLs publiques et privées qui permettent l'accès aux différents services.

#PUBLIC="http://172.16.0.1:5000/v2.0"
#ADMIN="http://172.16.0.1:35357/v2.0"
#INTERNAL=$PUBLIC

 # keystone endpoint-create --region RegionOne --service id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL

2. Le point d'accès du service Glance

Récupération de l'identificateur du service de Glance

 # ID=$(keystone service-list | awk '/ glance/ {print $2}')

Définition de l'adresse ainsi le port correspondant pour l'accès 0 ce service :

 # PUBLIC="http://172.16.0.1:9292/v1"
 # ADMIN=$PUBLIC
 # INTERNAL=$PUBLIC

Création du point d'accès :

 # keystone endpoint-create --region RegionOne --service id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL

3. Le point d'accès du service nova compute

Récupération de l'identificateur du service de Nova

#ID=$(keystone service-list | awk '/ nova / {print $2}')

 # PUBLIC=”http://172.16.0.1:8774/v2/$(tenant id)s”

 # ADMIN=$PUBLIC

 # INTERNAL=$PUBLIC

 # keystone endpoint-create --region RegionOne --service id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL

4. Le point d’accès du service nova volume

Récupération de l’identificateur du service du service volume

 # ID=$(keystone service-list | awk ’/ volume / {print $2}’)

 # PUBLIC=”http://172.16.0.1:8776/v1/ %(tenant id)s”

 # ADMIN=$PUBLIC

 # INTERNAL=$PUBLIC

 # keystone endpoint-create --region RegionOne –service id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL

2. Installation et configuration de Glance

Glance est le service chargé de distribuer les images de disque dur système utilisées par les machines virtuelles.

➢ Préparation de la base de données Mysql

La commande suivante crée un utilisateur nommé ”glance” et sa base de données nommée ”glance”, avec un mot de passe ”openstack”:

mysql mysql -u root -p
mysql CREATE DATABASE glance;
mysql GRANT ALL PRIVILEGES ON glance.* TO ’glance’@’%’ IDENTIFIED BY ’openstack’;
mysql FLUSH PRIVILEGES;
mysql Quit;

➢ Installation

Nous allons installer les paquets suivants :

 # apt-get install glance glance-api glanceclient glance-common glance-registry python-glance

- **Configuration de glance pour l'utilisation de keystone**

Dans le fichier /etc/glance/glance-api-paste.ini, nous modifions La section [filter:authtoken] comme suit :

Admin tenant name =service

Admin user =galanc

Admin-pasword=ADMIN

Nous allons modifier le fichier /etc/glance/glance-api.conf pour y ajouter les lignes suivantes :

[paste deploy]

flavor = keystone

Puis, le fichier /etc/glance/glance-registry.conf pour modifier la connexion à la base de données "glance"

sql connection = mysql://glance:openstack@172.16.0.1:3306/glance

Et nous ajoutons à la fin du fichier:

[paste deploy]

flavor = keystone

Nous allons modifier le fichier /etc/glance/glance-scrubber.conf pour ajouter les lignes suivantes:

sqlconnection=mysql://glance:openstack@172.16.0.1:3306/glance
sql idle timeout = 3600

Enfin, le fichier /etc/glance/glance-registry-paste.ini pour modifier les lignes suivantes:

Admin tenant name =service

Admin user =galanc

Admin-pasword=ADMIN

Nous allons synchroniser la base de données MySQL

 # glance-manage version control 0

 # glance-manage db sync

- Utilisation de glance

Maintenant nous allons vérifier si tout fonctionne correctement. Nous allons télécharger une première image pour tester:

 # wget http://uecimages.ubuntu.com/releases/precise/release/ ubuntu-12.04-server-cloudimg-amd64-disk1.img

Nous ajoutons maintenant l'image téléchargée aux images Glance:

 # glance add name="Ubuntu 12.04 cloudimg amd64" is public=true container format=ovf disk format=qcow2 ≺ubuntu-12.04-server-cloudimg-amd64-disk1.img

3. Installation et configuration de Nova

Le composant Nova, permet la gestion des instances des machines virtuelles, de leurs espace disque et du réseau

Préparation de la base de données Mysql

La commande suivante crée un utilisateur nommé "nova" et sa base de données nommée "nova", avec un mot de passe "openstack":

mysqlmysql -u root -p

mysqlCREATE DATABASE nova;

mysqlGRANT ALL PRIVILEGES ON nova. TO 'nova'@'%' IDENTIFIED BY 'openstack';*

mysqlFLUSH PRIVILEGES;

mysqlQuit;

- Installation

Nous allons installer les paquets suivants :

 # apt-get install nova-api nova-cert novacommon nova-compute nova-compute-kvm nova-docnova-network nova-objectstorenova-scheduler novnc nova-consoleauth nova-volume nova-console python-nova pythonnovaclient.

Nous modifions le fichier /etc/nova/api-paste.ini comme indiqué ci-dessous:

Admin tenant name =service

Admin user =nova

Admin-pasword=ADMIN

Puis, nous redémarrons tous les services nova:

 # for a in libvirt-bin nova-network novacompute nova-api nova-objectstore nova-scheduler nova-volume nova-cert novaconsoleauth novnc; do sudo service "$a" stop; done

 # for a in libvirt-bin nova-network novacompute nova-api nova-objectstore nova-scheduler nova-volume nova-cert novaconsoleauth novnc; do sudo service "$a" start; done

Nous synchronisons la base de données :

 # nova-manage db sync

Et nous redémarrons de nouveau tous les services

 # for a in libvirt-bin nova-network novacompute nova-api nova-objectstore nova-scheduler nova-volume nova-cert novaconsoleauth novnc; do sudo service "$a" stop; done

 # for a in libvirt-bin nova-network novacompute nova-api nova-objectstore nova-scheduler nova-volume nova-cert novaconsoleauth novnc; do sudo service "$a" start; done

- Utilisation de nova

Images disques: pour lister les images disque fournies par le service Glance, nous utilisons la commande suivante:

 # nova image-list

Réseaux : nous allons maintenant créer les réseaux privés et publics. Les adresses seront enregistrées dans la base MySQL

- Réseau public

#nova-manage floatingcreate- ip range=172.0.1.0/24

- Réseau privé

 # nova-manage network create private --fixed range v4=10.0.1.0/24 --num networks=1 --bridge=br100 --bridge interface=eth2

4. Installation du Dashboard(Horizon)

L'interface graphique Dashboard a été d´enveloppée pour simplifier l'administration du serveur et des projets. L'accès se fait à partir d'un navigateur web pointant à l'adresse du serveur.

Les différents services (Keystone, Glance, Nova,...) doivent être installés et configurés avant de l'utiliser. Une grande partie des commandes est alors à portée d'un clic de souris.

Nous allons installer les paquets suivants :

 # apt-get install openstack-dashboard apache2 libapache2-mod-wsgi

Nous pouvons configurer OpenStack Dashboard en modifiant le fichier /etc/openstack-dashboard/local settings.py

OPENSTACK HOST = "172.16.0.1"

OPENSTACK KEYSTONE URL = "http://%s:5000/v2.0" % OPENSTACK HOST

OPENSTACK KEYSTONE DEFAULT ROLE = "Member"

Et nous redémarrons le serveur web Apache pour vérifier que tout fonctionne bien :

 # sudo service apache2 restart

➢ Utilisation

Nous ouvrons un navigateur web à l'adresse: "http://172.16.0.1/horizon/". La première interface dashboard d'authentification apparait:

Figure 21 :L'interface d'authentification du dashboard

L'administrateur et les utilisateurs peuvent s'authentifier auprès de cette interface. L'administrateur s'authentifie avec le nom d'utilisateur "admin" et le mot de passe "ADMIN"

Une fois l'authentification est faite, nous retrouvons sur la vue d'ensemble du projet

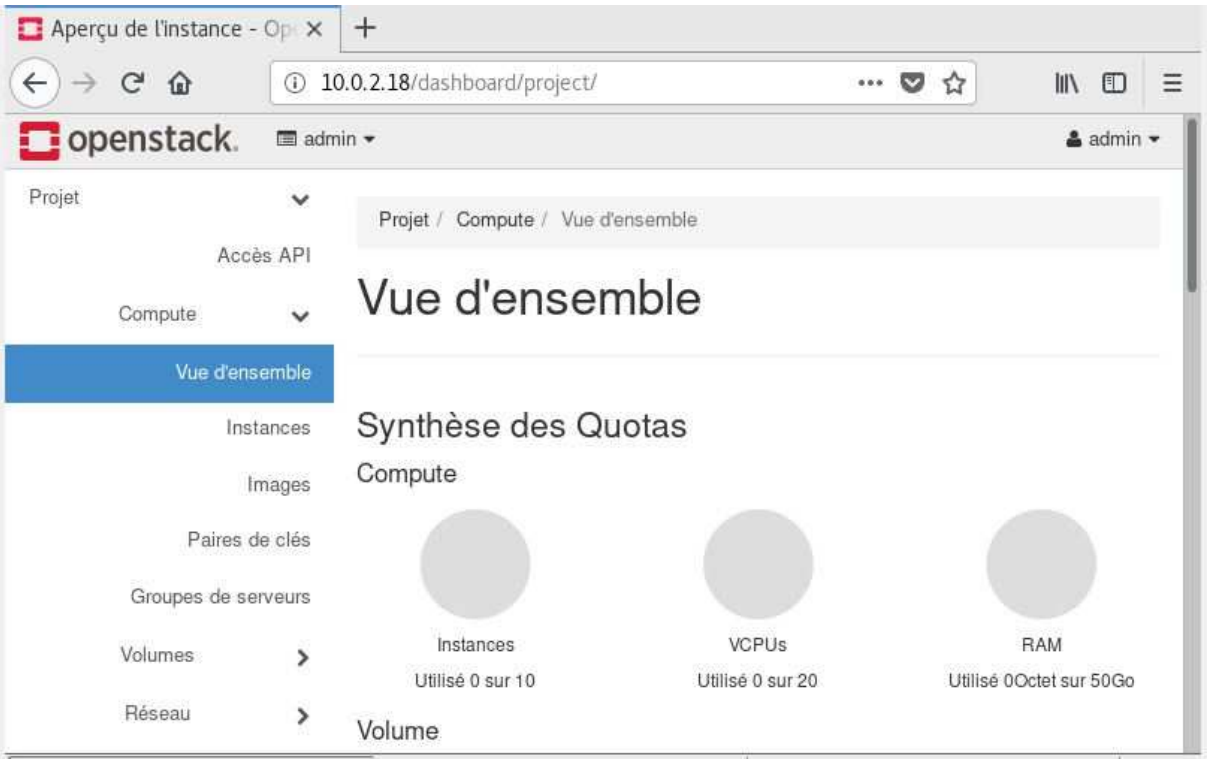

Figure 22 une vue d'ensemble

Cette vue d'ensemble nous donne l'utilisation des quotas alloués comme le nombre d'instances, le nombre de cœurs, la mémoire vive, l'espace disque ou les adresses IP publiques. Cependant, cette vue n'est pas représentative des ressources disponibles sur le cloud. Il ne s'agit l`a que des quotas autorisés

Les étapes de configuration de la machine virtuelle.

1. **Allocation d'une adresse flottante**

Une adresse IP flottante permet un accès externe depuis des réseaux extérieurs ou Internet à une machine virtuelle Openstack.

Afin de créer des adresses IP flottantes, nous allons nous connecter avec les informations d'identification d'utilisateur et accéder à l'onglet Projet-> Calcul - - Accès et sécurité -> IP flottantes, puis nous cliquons sur *Attribuer une adresse IP au projet.*

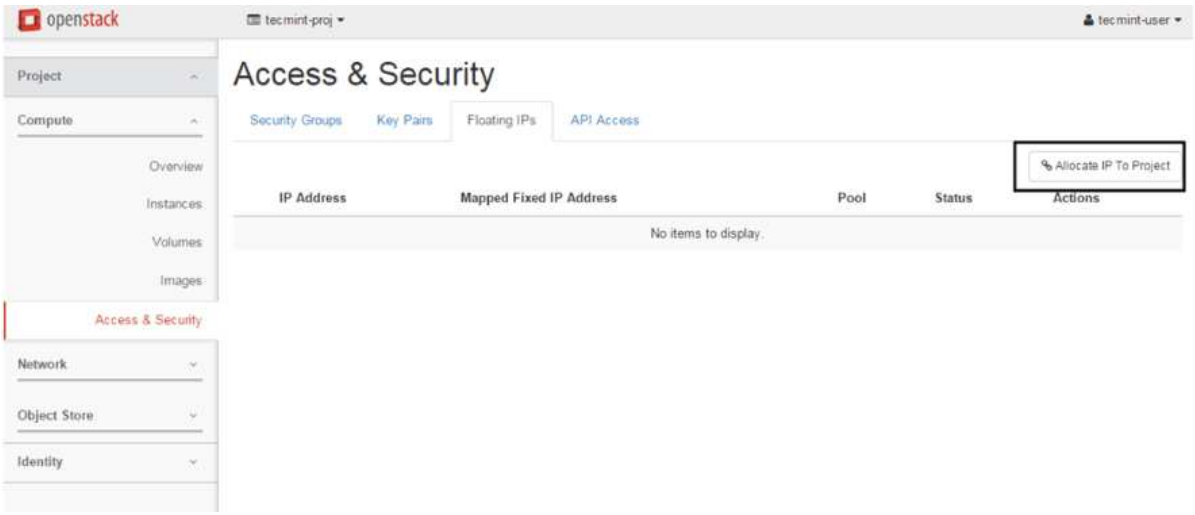

Figure 23: Affectation d'adresse IP flottante à un projet dans OpenStack

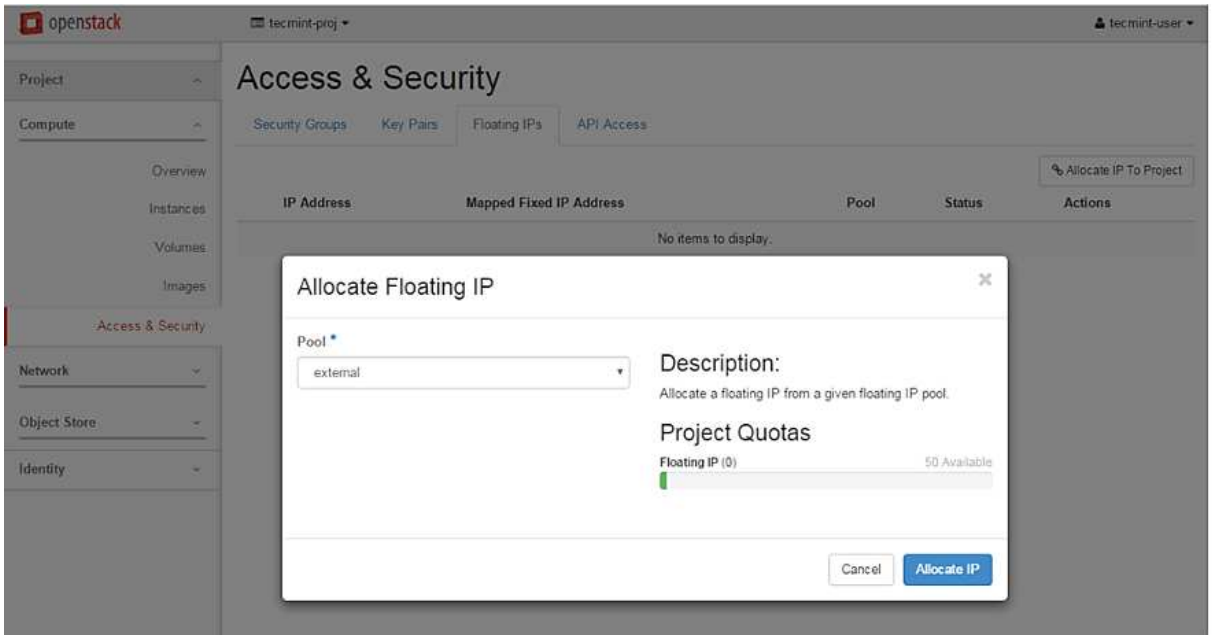

Figure 24: Allocation une adresse IP flottante à un pool externe

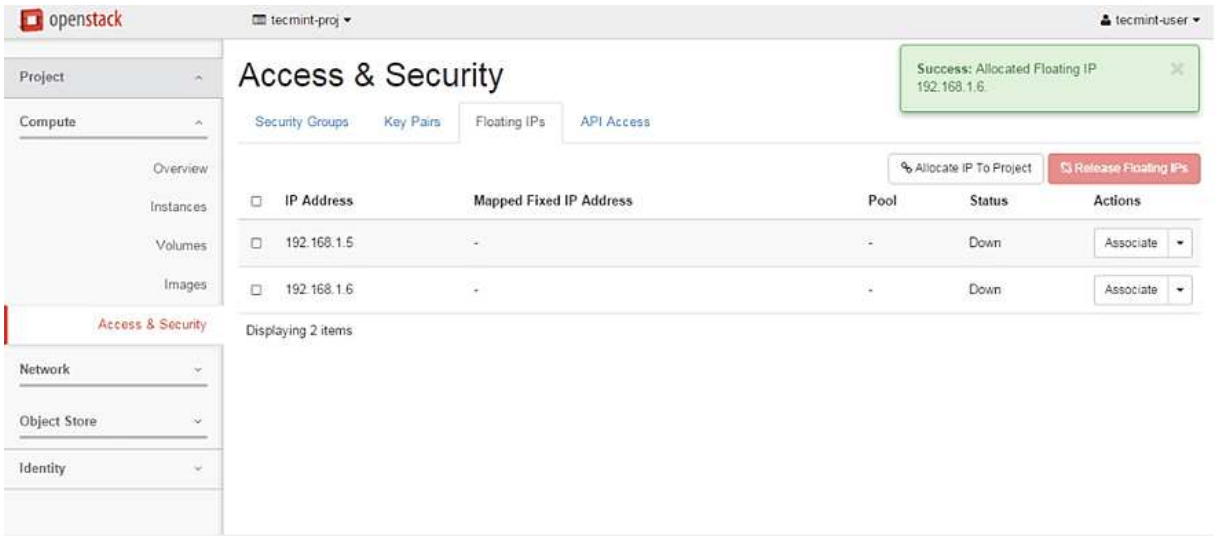

Figure 25: Confirmation de l'ajout

2. La création d'une image

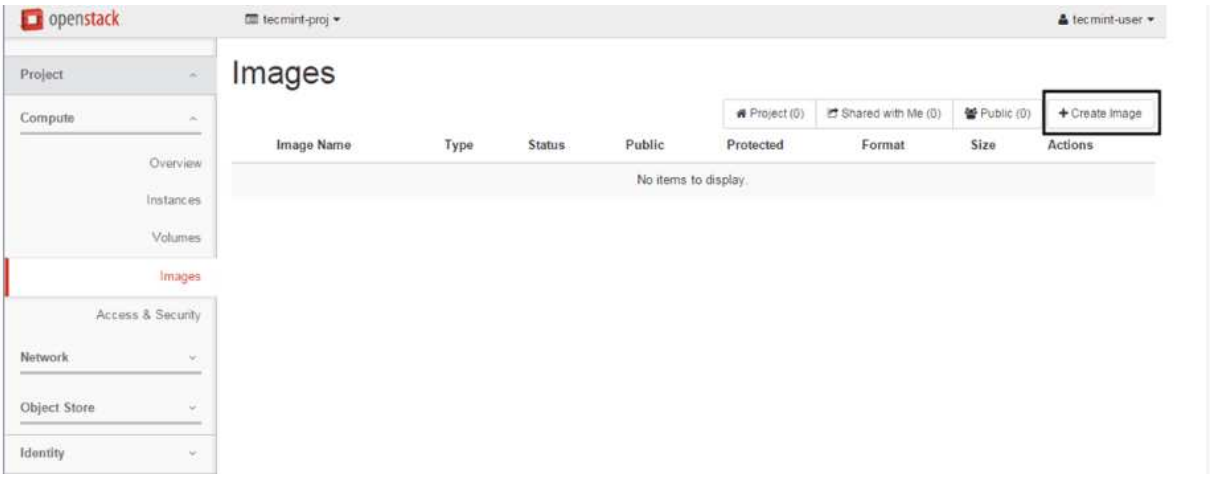

Figure 26 : Création d'une image

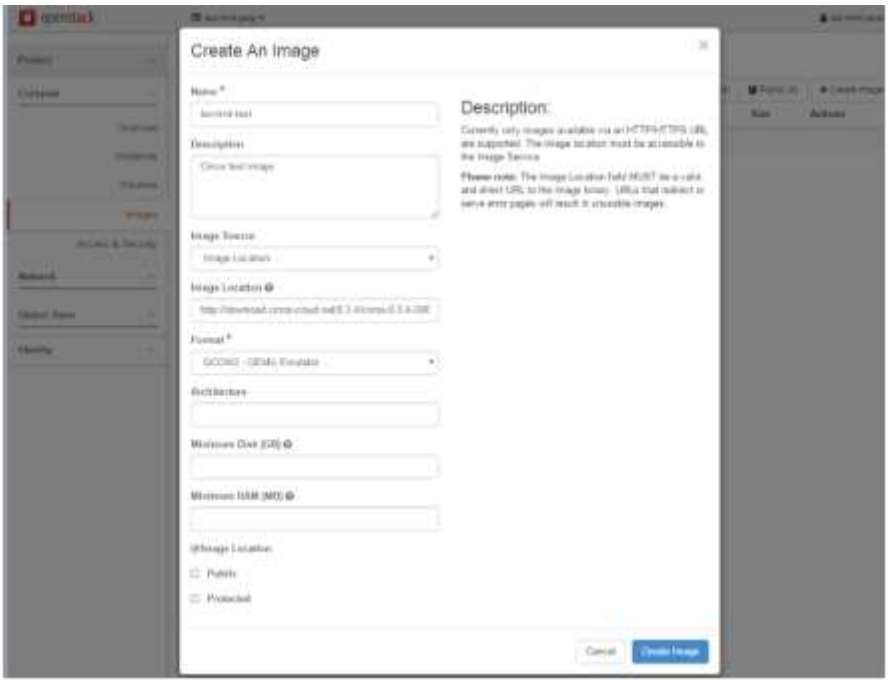

Figure 27: Ajout des détails de l'image OpenStack

•

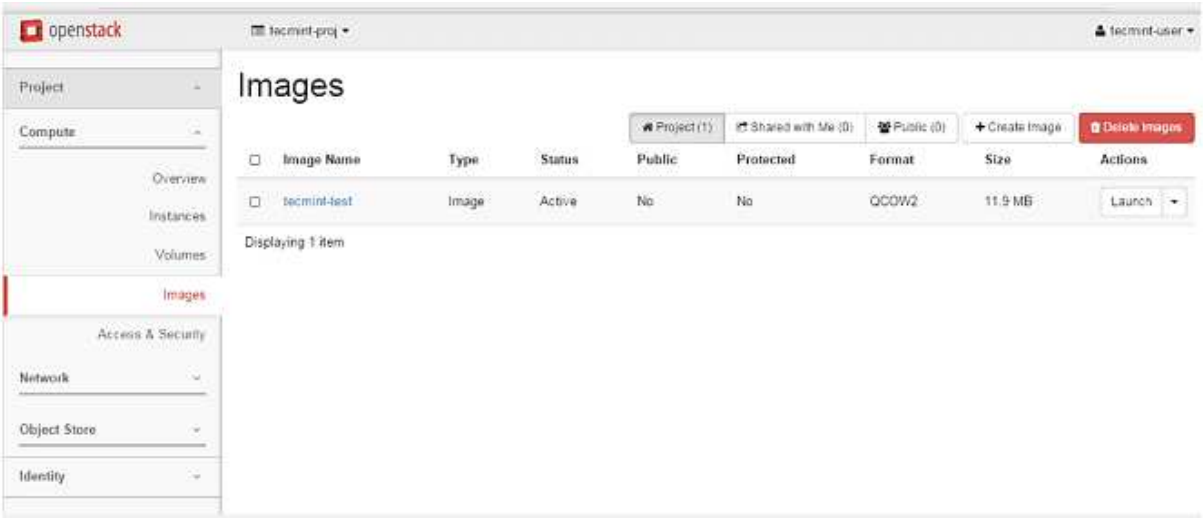

Figure 28: Image OpenStack créer

3. Lancer une instance d'image dans OpenStack

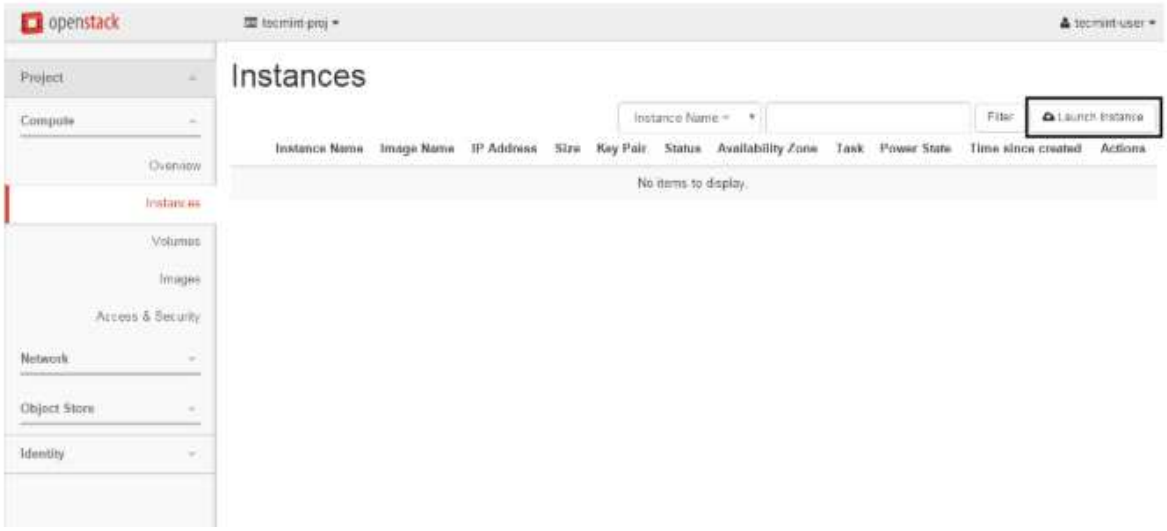

Figure 29 : Lancement d'une instance d'image dans Openstack

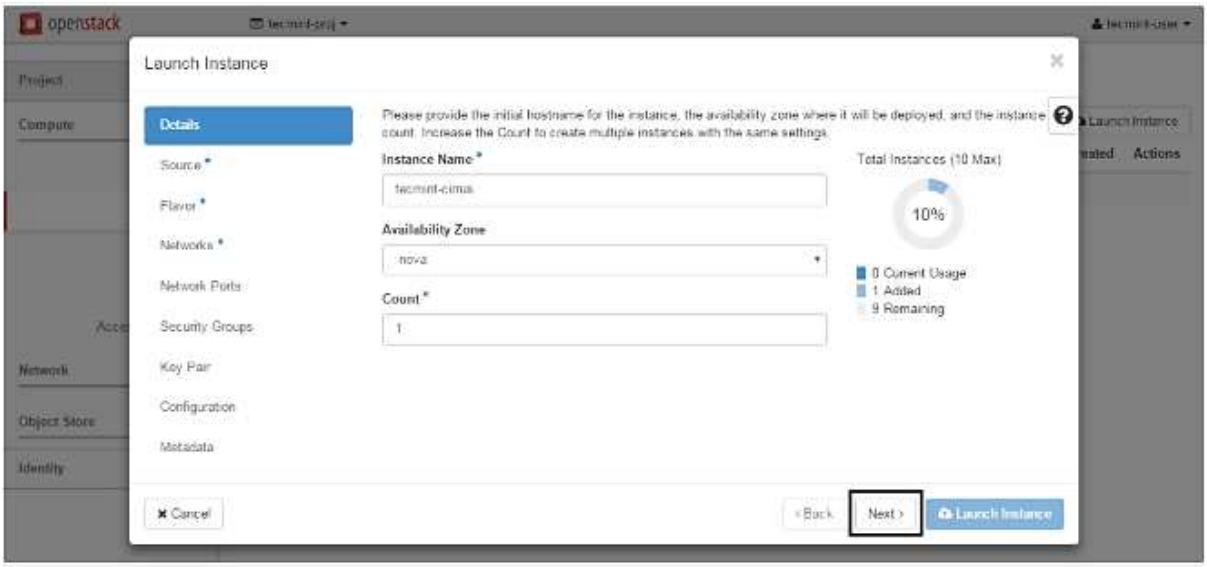

Figure 30 : Ajout d'un nom d'hôte à une instance OpenStack

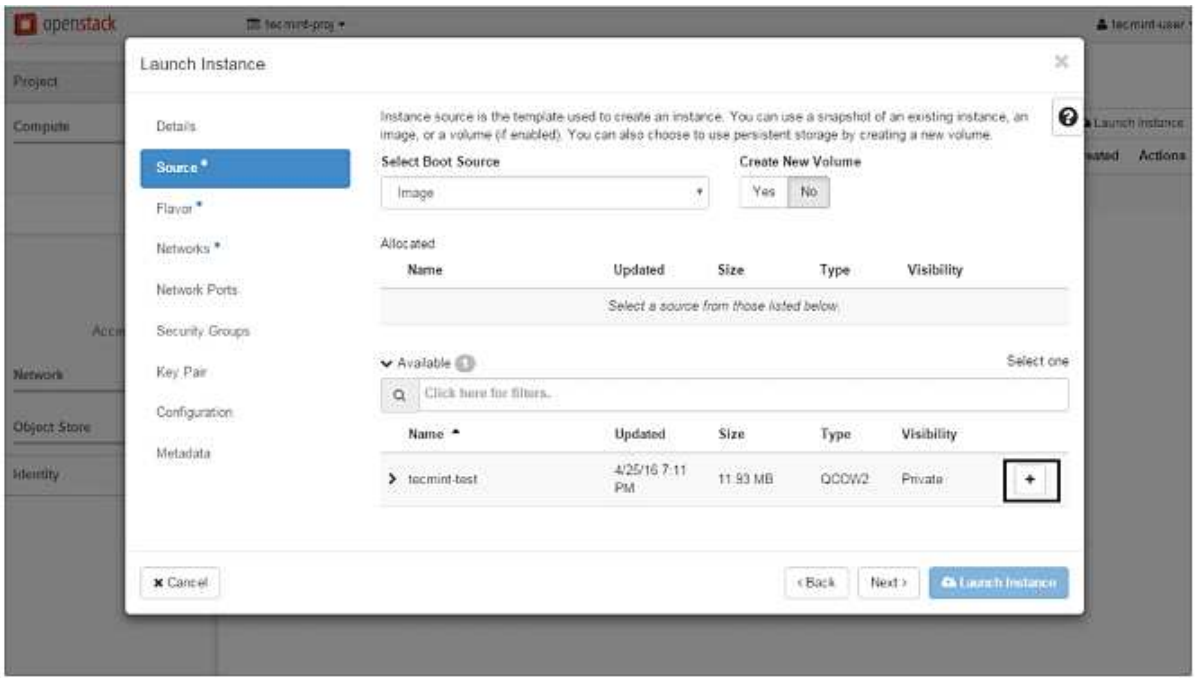

Figure 31 : démarrage de l'instance OpenStack

Allocation ressources de la machine virtuelle:.

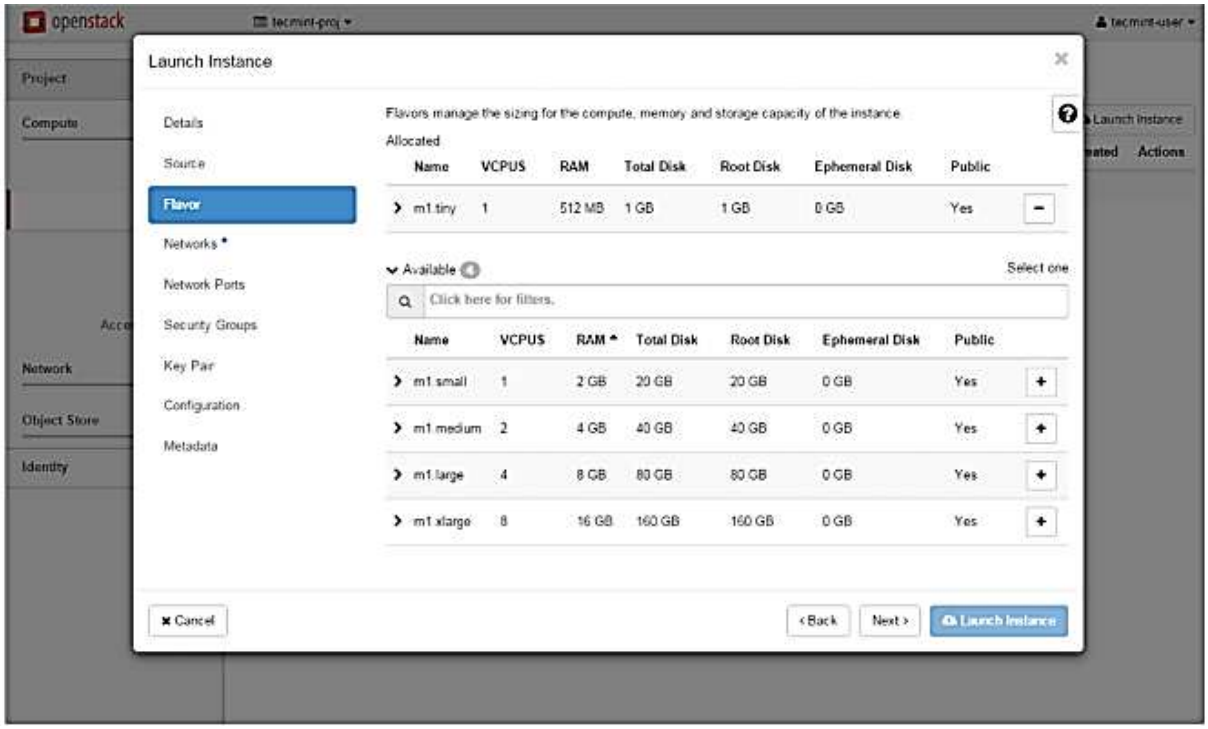

Figure 32 : Ajout des ressources à l'instance OpenStack

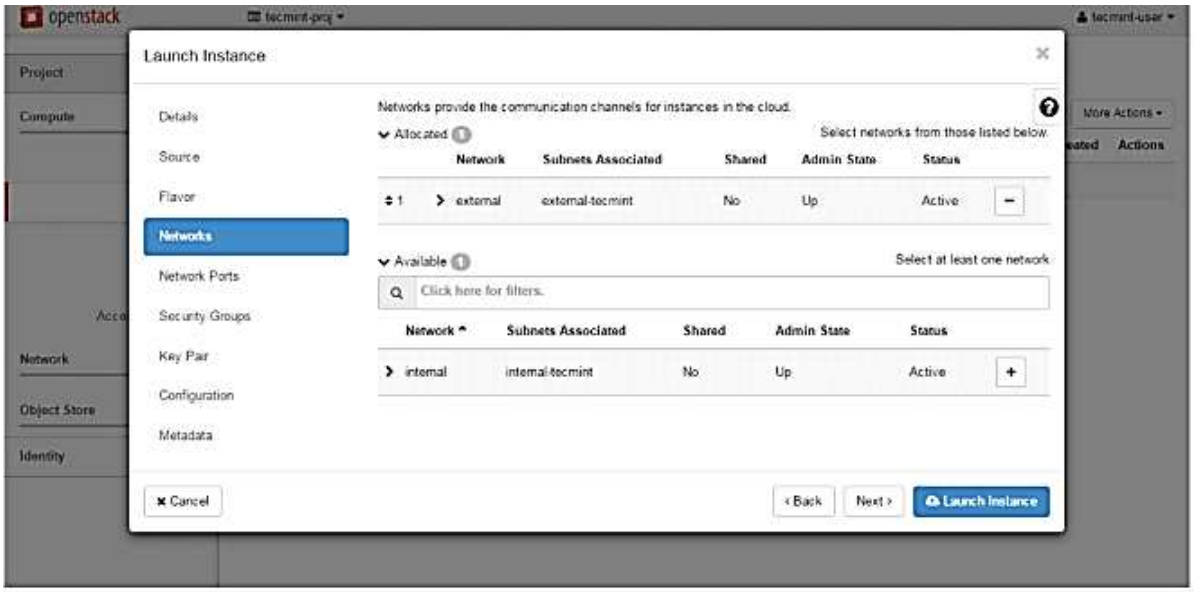

Figure 33 :Ajout d'un réseau à une instance OpenStack

Une fois l'instance lancée, on lui associe une adresse IP flottante en sélectionnant l'une des adresses IP flottantes créées précédemment.

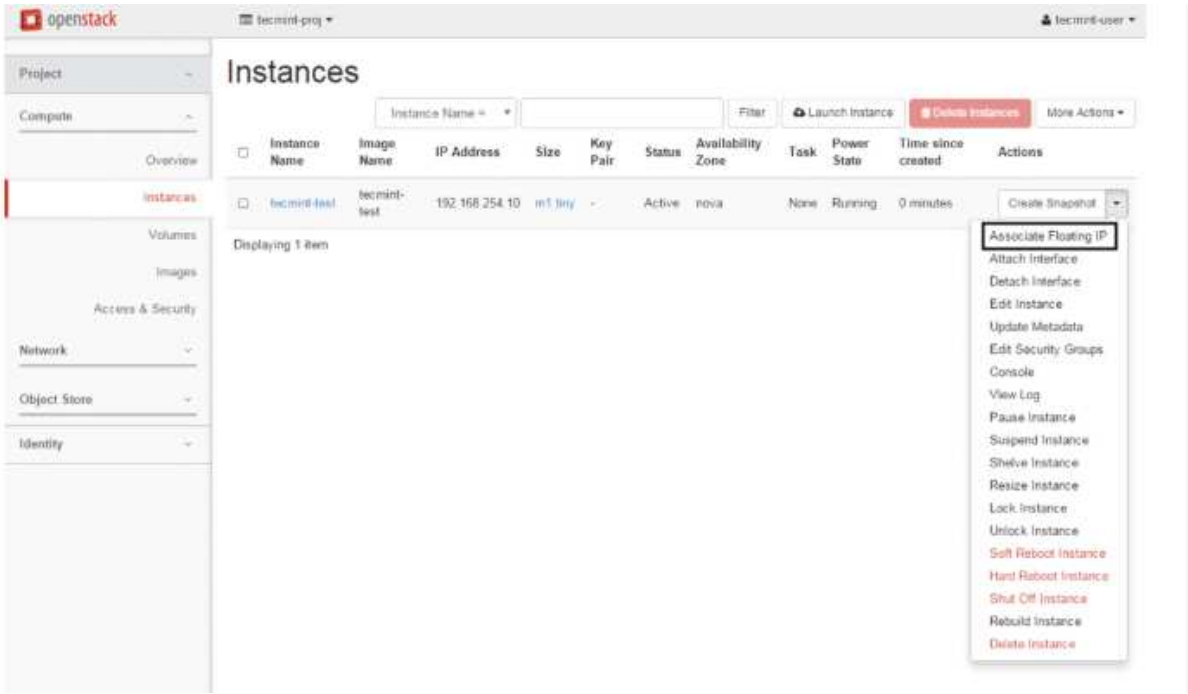

Figure 34 : l'ajout d'une adresse IP flottante associée à une instance OpenStack

Pour tester la connectivité réseau la machine virtuelle, on émette une commande ping contre l'adresse IP flottante de l'instance à partir d'un ordinateur distant de votre réseau local.

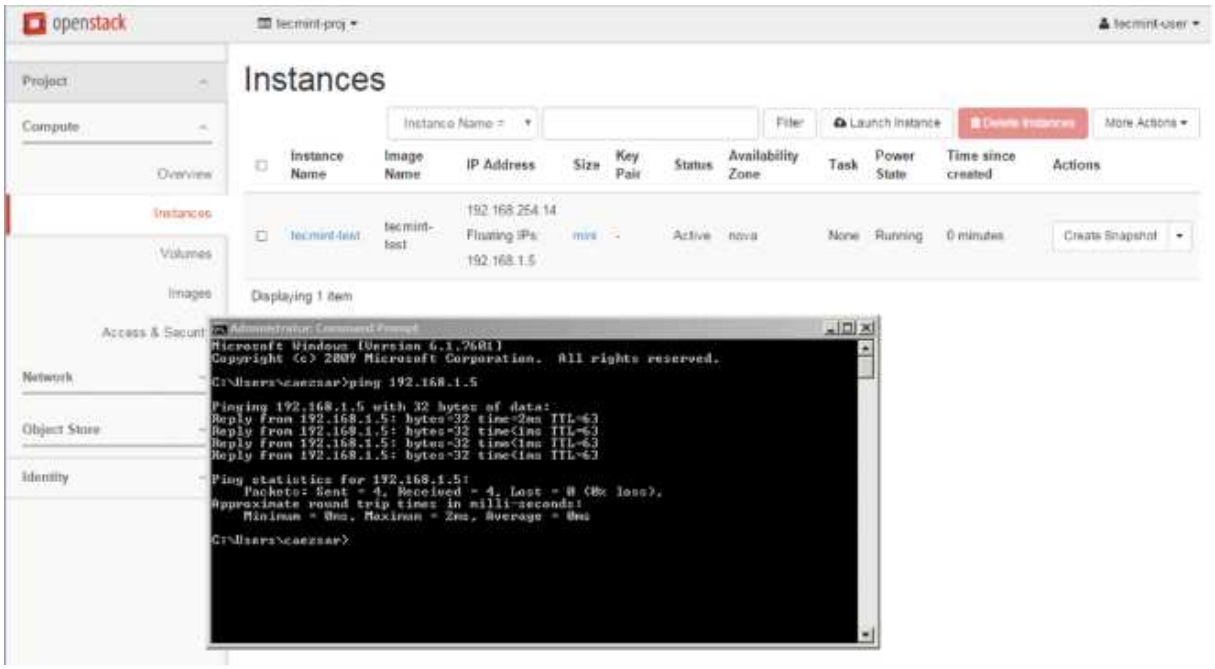

Figure 35 :Vérification le réseau de la machine virtuelle dans OpenStack

On ajoute les serveurs de noms DNS requis, enregistrez la configuration, démarrez et connectez-vous à la console de l'instance pour vérifier si la nouvelle configuration a été appliquée en envoyant une requête ping à un nom de domaine

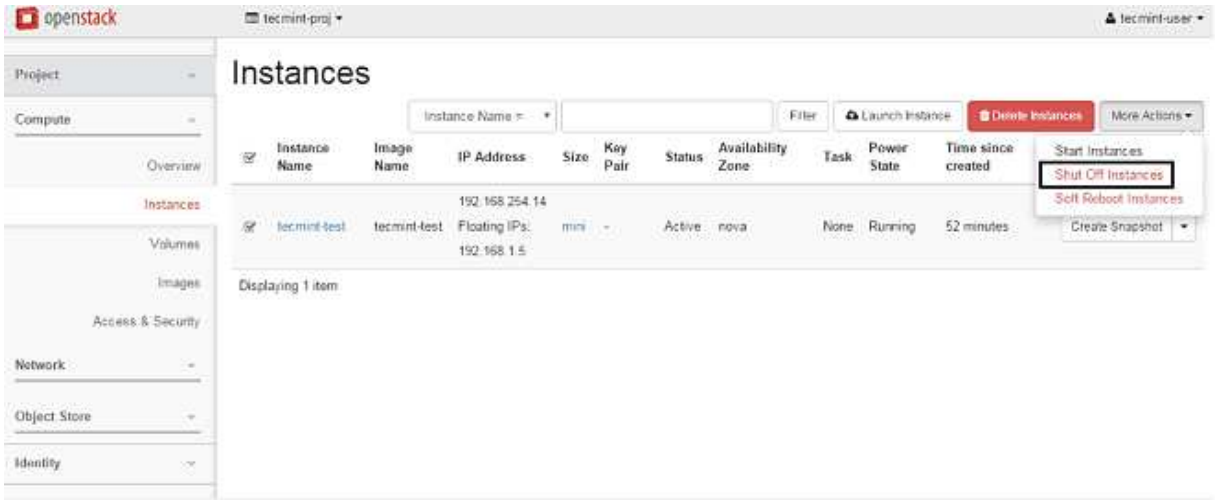

Figure 36 : Instance d'arrêt

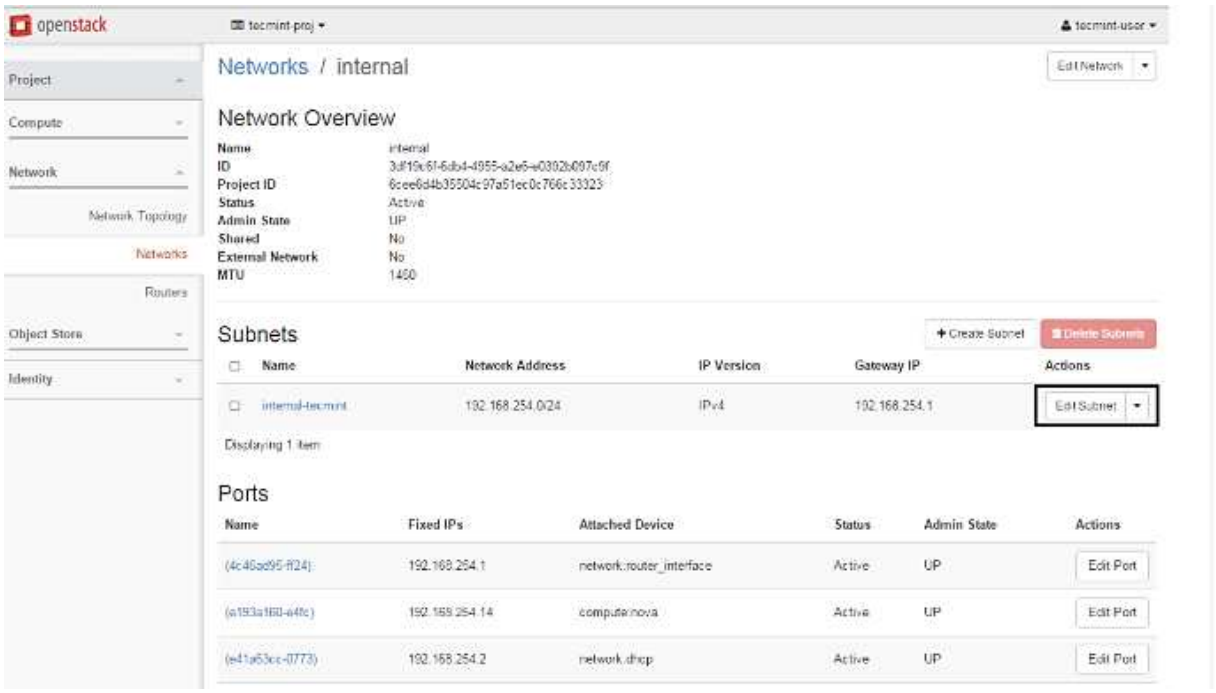

Figure 37 : Modification du sous-réseau de l'instance

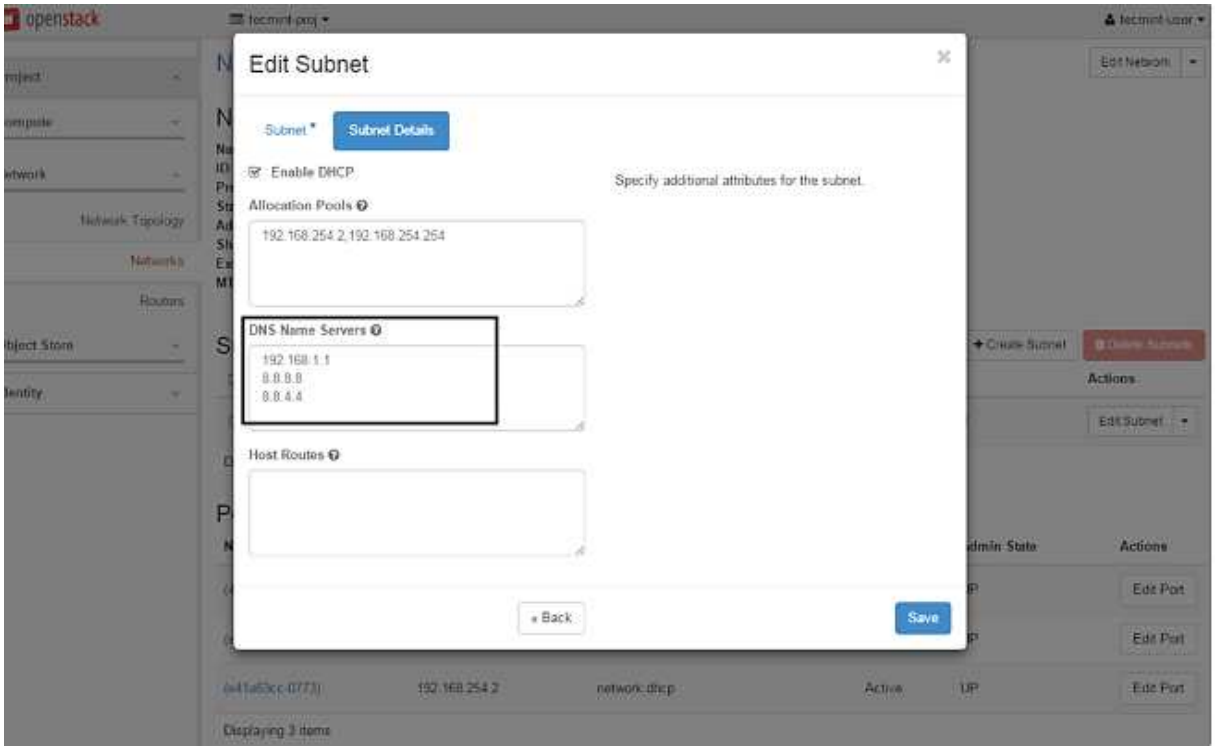

Figure 38 : Ajout des serveurs DNS à l'instance

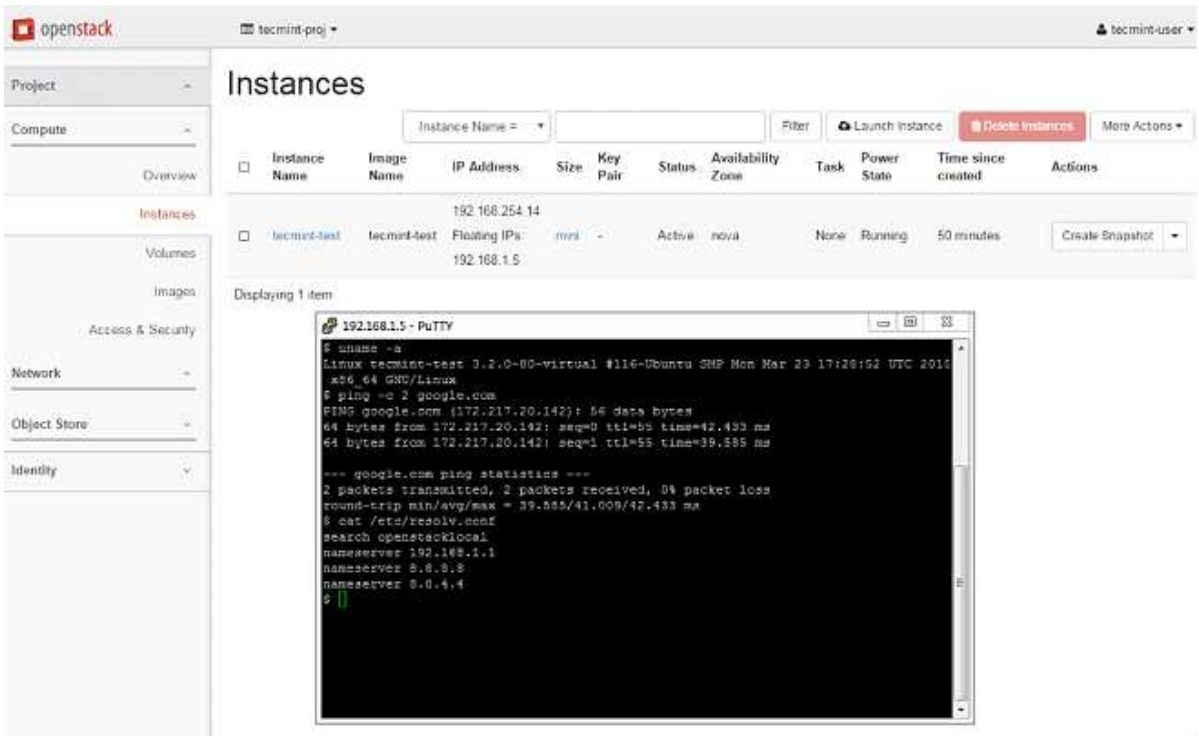

Figure 39 : La Vérification de la connectivité réseau de l'instance

Conclusion

Dans cette partie nous avons étudié deux solutions Open source : OpenNebula et OpenStack qui permettent la mise en place d'une solution Cloud Computing de type IaaS. Ces plateformes permettent d'accéder facilement à différents services fournis par le Cloud.

Dans le chapitre suivant nous intéressons à présenter les failles de sécurité que nous avons trouvé lors de la mise en place de ces plateformes et nous allons proposer des solutions afin d'atténuer ces failles et pour finir nous allons établir un comparatif récapitulatif d'OpenNebula et d'OpenStack.

Chapitre IV : Les solutions proposées pour remédier aux failles de sécurité dans OpenNebula et OpenStack

Introduction :

Après avoir défini et détaillé les concepts des plateformes « OpenStack » et « OpenNebula » et les avoir implémenté sur les machines virtuelles, nous nous intéressons dans ce chapitre à la présentation les failles de sécurité que nous avons rencontrées lors de leur installation et proposer des solutions qui feront face à ces failles. Nous allons finir par un récapitulatif comparant les deux failles de sécurité et les solutions associées.

I. Sécurité sur OpenStack :

OpenStack comprend différents ensembles de composants offrant des fonctionnalités propres à chaque module. Ces composants peuvent être ciblés individuellement par un attaquant. Par conséquent, pour sécuriser OpenStack dans sa gestion des requêtes/réponses du serveur, nous devons évaluer les vulnérabilités de sécurité présentes.

Dans cette partie nous examinons certaines des vulnérabilités dans OpenStack qui pourraient causer un préjudice grave si elles ne sont pas corrigées à temps.

I.1. Vulnérabilité liée à la fixation de session

La vulnérabilité liée à la fixation de session a été découverte dans le module *Horizon*. Cela s'est produit lors de l'utilisation de la session de *cookie* signé par défaut.

Lors de la mise en place d'une connexion entre un client et le serveur Horizon, un cookie propre à la session est généré. Ce cookie est généré et stocké par le navigateur de l'utilisateur et lorsqu'il est configuré pour utiliser des sessions côté client, le serveur n'a pas connaissance de l'état de connexion de l'utilisateur. Les jetons sont stockés dans l'identifiant(ID) de session dans le cookie.

Le paramètre par défaut dans le module *Horizon* consiste à utiliser des cookies signés pour stocker l'état de la session côté client. Cela crée la possibilité que si un attaquant est capable de capturer le cookie d'un utilisateur, il peut effectuer toutes les actions avec cet utilisateur, même si celui-ci s'est connecté en dehors.

Si un attaquant peut voler le cookie, il peut effectuer toutes les actions en tant qu'utilisateur cible, même après sa déconnexion.

Un attaquant peut intercepter une requête de l'utilisateur vers le serveur, et à l'aide de son adresse IP, il peut récupérer le cookie de session généré et se faire passer pour l'utilisateur au près du serveur. Pour pallier à ce risque, nous allons configurer OpenStack pour initier une connexion à l'aide d'un certificat SSL que nous allons générer et qui servira à authentifier l'utilisateur auprès du serveur et à encrypter les cookies de session même après déconnexion.

I.2. Connexion non sécurisée :

La connexion de notre infrastructure se fait en http, cela signifie que tous les échanges avec entre le navigateur et l'application d'utilisateur se font en texte non crypté et donc lisible par n'importe quelle personne qui espionne le réseau.

I.3. Les solutions sur OpenStack

I.3.1. Utilisation de session en cache

Pour stocker les données de sessions en utilisant le système de cache de Django, il s'agit d'abord de s'assurer que le cache est configuré.

local_settings.py

SESSION_ENGINE = 'django.contrib.sessions.backends.signed_cookies'

Si SESSION_ENGINE est défini sur une valeur autre que 'django.contrib.sessions.backends.signed_cookies', cette vulnérabilité n'est pas présente. Si SESSION_ENGINE n'est pas défini dans local_settings.py, vérifiez-le dans settings.py.

Les étapes pour configurer les sessions memcache sont les suivantes :

1. Assurez-vous que le service memcached est exécuté sur le système.

2. Assurez-vous que python-memcached est installé

3. Configurez le backend du cache memcached dans local_settings.py

CACHES = {

'défaut': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION': '127.0.0.1:11211',

}

}

Assurez-vous d'utiliser l'adresse IP et le port réels du service memcached.

4. Ajoutez une ligne dans local_settings.py pour utiliser le backend du cache:

SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

I.3.2. Solution de connexion https via ssl

Pour résoudre le problème de connexion non sécurisée nousavons utilisé une configuration avec un certificat SSL/TLS auto-signé.

Pour mettre en place cette configurer nous allons suivre les étapes suivantes :

- D'abord nous commençons par l'installation du module apache mod_ssl car le chiffrement SSL/TLS pour apache est fourni par le paquet mod_ssl. Pour se faire on exécute la commande suivante :

 # yum install mod_ssl

On notera l'apparition d'un fichier de configuration ssl.conf dans /etc/httpd/conf.d

- Avant de continuer, on effectue une copie de sauvegarde du fichier de configuration par default :

$ cd /etc/httpd/conf.d

$ sudo cp ssl.conf ssl.conf .orig

- Ensuite, on crée un certificat et une clé qui seront respectivement stockés dans les répertoires ***private*** et ***certs :***

 # openssl req -x509 -nodes -days 365 -newkey rsa :2048 -keyout /etc/pki/tls/private/openstack.key -out /etc/pki/tls/certs/openstack.crt : cette ligne de code va nous permettre de générer une clé privée RSA

x509 : format de la clé

days : validité de la clé en jours

newkey rsa : création de la nouvelle clé de taille 2048bits

keyout : nom et destination de la clé générée

out : nom et destination du certificat généré

- On accède au répertoire de configuration de serveur apache :

 # cd /etc/httpd/conf.d/

Ensuite on édite le module open_ssl :

 # vim 15-horizon_vhost.conf

On active l'hébergement virtuel basé sur un nom sur le port 443 (car le protocole ***https*** utilise ce port) en ajoutant les lignes suivantes juste avant le commentaire de contexte d'hôte virtuel SSL :

NomVirtualHost *: 443

On ajoute ainsi les lignes suivantes :

SSLEngine on (pour activer le protocole ssl)

SSLCertificateFile /etc/pki/tls/certs/openstack.crt (spécifier le chemin vers le certificat)

SSLCertificateKeyFile /etc/pki/tls/private/openstack.key (spécifier le chemin vers la clé)

- Après avoir édité et enregistré le fichier ssl.conf, on redémarre le serveur http Apache

 # httpd –t

 # systemctl restart httpd

- Maintenant que nous avons corrigé la configuration de base de la configuration du module, nous devons nous assurer d'autoriser le trafic SSL sur le pare-feu (firewall). Dans ce cas nous exécutons la commande ci-après :

 # firewall.cmd --permanenent --add-service=httpds

Et pour finir on recharge la configuration firewall :

 # firewall.cmd --reload

II. La sécurité sur OpenNebula :

- Lors de la création de l'utilisateur 'oneadmin' pendant l'installation du front-end et du nœud, une clé RSA a été générée, permettant au front-end et aux hôtes de communiquer par SSH de manière sécurisée. Néanmoins, lorsque qu'un hôte est ajouté, la clé du front-endest copiée vers l'hôte afin de permettre l'établissement d'une connexion SSH sans mot de passe. (une clé rsa est générée dans le fichier .ssh de oneadmin et cette clé est distribuée à tous les hôtes qu'il fédère).
- L'authentification d'un hôte auprès du serveur ssh d'opennebula se fait via la clé générée et donc, le front-end n'a plus besoin de demander un mot de passe à la machine qui initie la connexion. Si un noeud est infecté, l'attaquant pourra accéder à n'importe quel autre noeud de l'infrastructure via SSH sans qu'un mot de passe ne lui soit demandé.

II.1. Solution proposée sur OpenNebula

La solution mise en place pour sécuriser les accès vers OpenNebula se divise en deux parties:

1) Mettre en place un protocole d'authentification via un certificat SSL afin d'identifier les utilisateurs qui tentent de se connecter au Front-end.
2) Mettre en place un protocole de chiffrement des données du navigateur afin d'encrypter les informations que le navigateur stocke durant l'établissement d'une session via Sunstone.

a. Mise en place du protocole d'authentification avec OpenNebula:
On commence par éditer le fichier /etc/one/auth/x509_auth.conf
nano /etc/one/auth/x509_auth.conf
et on specifie le chemin vers le repertoire des certificats que nous approuvons afin qu'OpenNebula puisse leur faire confiance.
:ca_dir:/etc/one/auth/certificates

On génére ensuite une clé de chiffrement RSA d'une longueur de 2048 bits
#openssl genrsa -out privatekey.pem 2048

On crée le certificat qui va contenir notre clé publique avec en entrée notre clé de chiffrement et en spécifiant la durée de validité du certificat en jours
#openssl req -new -x509 -key privatekey.pem -out publickey.pem -days 365

Une fois le certificat généré, on va changer le mot de passe de l'utilisateur Oneadmin par le certificat que nous venons de créer
oneuser chauth oneadmin x509 --x509 --cert publickey.pem

Nous allons hasher notre certificat afin d'encoder le nom. Ceci nous servira à dissimuler l'origine du certificat présent dans le repertoire de certificats de confiance
#openssl x509 -noout -hash -in publickey.pem

Nous allons ajouter le certificat à notre répértoire de certificats de confiance
cp publickey.pem /etc/one/auth/certificates/78d0bbd8.0

Il ne nous reste plus qu'à s'authentifier à l'aide du certificat
oneuser login oneadmin --x509 --cert publickey.pem --key privatekey.pem
#export ONE_AUTH=/home/oneadmin/.one/one_x509

Un jeton de connexion sera généré. Il aura une validité d'une heure avant de clore la session et de déconnecter l'utilisateur.

b. Mettre en place un protocole de chiffrement des données du navigateur:
Pour pouvoir sécuriser nos sessions de connexion et pouvoir s'assurer que nos données ne puissent pas être lues en cas d'interception, nous allons mettre en place un serveur web ainsi qu'un serveur de certificat afin d'obtenir une session de connexion en HTTPS

On commence par installer ssl-cert, notre serveur de certificat qui va nous permettre de générer des certificats
apt-get install ssl-cert

Ensuite on génére un certificat et une clé
#/usr/sbin/make-ssl-cert generate-default-snakeoil

Il ne nous reste plus qu'à installer et configurer un serveur web pour qu'il puisse utiliser le certificat que nous avons généré
apt-get install nginx
nano /etc/nginx/sites-enabled/sunstone
et on modifie les parametres comme suit

```
upstream sunstone  {
server 127.0.0.1:9869;
}

server {
        listen 80;
        server_name 10.0.2.15;

       return 301 https://$server_name:8443;
}

server {
        listen 8443;
```

```
    server_name cloudserver.org;

    ssl on;
    ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;
    ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key;

    location / {
        proxy_pass http://sunstone;
    }
}
```

III. Etude Comparative des solutions « OpenStack » et « OpenNebula »

III.1. Hyperviseur et instance :

On rappelle que l'hyperviseur est une plateforme de virtualisation qui permet de faire fonctionner plusieurs systèmes d'exploitation sur une même machine physique.

Le tableau suivant liste les différents hyperviseurs disponibles pour les deux solutions considérées :

Hyperviseur	OpenNebula	OpenStack
KVM	•	•
XenServer		•
Xen	•	•
EMU		•
LXC		•
VMWare	•	•
Hyper-V		•
Docker		•
PowerKVM		•

Tableau 1 : Hyperviseurs supportés par les solutions open source

III.2. Communauté, Développement, Documentation :

Outre les fonctionnalités et spécificités techniques de ces solutions, il ne faut pas négliger la communauté qui encourage les deux autres éléments importants : le développement et la documentation.

Choisir une solution dont la communauté est faible ou le développement n'est pas dynamique ou encore une documentation insuffisante mener le projet à l'échec avant la production ou imposer d'importantes difficultés pouvant aller jusqu'à l'obligation de changer l'infrastructure complète en production. Il est important donc d'analyser ces caractéristiques.

Le tableau ci-après résume la situation :

	OpenNebula	**OpenStack**
Communauté	Faible	Grande
Développement	Moyen	Très élevé
Documentation	Faible	Forte

Tableau2 : Comparaison des communautés, de l'activité de développement et de la documentation pour les solutions IaaS open source

III.3. Comparatif de mise en place :

OpenNebula	OpenStack
• Installation rigide. Pas assez de liberté de configuration (une seule méthode de configuration). • Ne demande pas beaucoup de ressource	• Installation souple (ajout et configuration des modules selon le besoin). • Gourmant en ressources

Tableau3 : Comparaison de la mise en place de chaque plateforme.

III.3. Comparatif de la sécurité :

Au travers de notre travail, nous avons pu remarquer que:

- OpenNebula et OpenStack permettent de gérer les privilèges des utilisateurs et des groupes pour ainsi garantir que les ressources sont accessibles uniquement par les utilisateurs autorisés.
- OpenNebula et OpenStack permettent de déployer des machines virtuelles sur des typologies réseau très variées. Il est ainsi possible de sécuriser les accès réseaux en fonction des besoins
- Ils permettent d'attribuer une paire de clé RSA aux machines virtuelles qui communiquent avec l'exterieur pour ainsi sécuriser les accès via SSH.
- La communication entre le Front-End d'OpenNebula et ses noeuds se fait via SSH. Les clés RSA qui vont pouvoir authentifier un utilisateur distant sont entreposées dans le repertoire /.ssh/ de l'utilisateur. Verouiller l'accès à ce repertoire bloque toutes les communications. Etant donné qu'OpenNebula s'installe sur un système d'exploitation Linux, il n'est pas écarté qu'une faille de sécurité propre à Linux puisse permettre d'accéder au répertoire et récupérer les clés.
- OpenStack permet la communication entre ces différents composants grace au serveur RabbitMQ permet de valider, transformer et rediriger transmis entre les modules d'OpenStack. Il agit comme médiateur entre les émetteurs et les récepteurs en leur permettant de communiquer efficacement avec un couplage minimum entre eux. Chaque accès à un composant passe par une authentification et une autorisation du composant Keystone qui va générer un jeton de connexion une fois la demande d'authentification approuvée.

Conclusion

Ce chapitre nous a permis de présenter les solutions que nous avons adoptées pour traiter et éviter les failles de sécurité trouvées sur OpenStack et OpenNebula. Ces solutions reposent principalement sur le chiffrement SSL dans le but de protéger les mots de passe enregistrés en clair dans la base de données et assurer une connexion sécurisée pour les utilisateurs.

Pour finaliser notre travail nous avons fait une comparaison entre ces deux solutions d'un point de vue technique et d'un point de vue sécuritaire. Nous avons essayé de relever les defaillances de conception qui pourraient nuir à la sécurité de l'infrastructure.

Conclusion générale et perspective

Le Cloud Computing est un nouveau concept de déploiement de systèmes informatique, il offre beaucoup d'avantages en termes de puissance de calcul, de temps de réponse et de réduction des coûts. Les utilisateurs peuvent bénéficier pleinement des services Cloud qui permettent de satisfaire leurs besoins à la demande. Toutefois, comme chaque avancée technologique, externaliser ses ressources informatiques apporte aussi sa part de risques, notamment en termes de sécurité des données, car si l'utilisateur ne peut pas avoir ses propres ressources de manière sécurisée, à tout moment et à partir de n'importe quel emplacement géographique, alors l'efficacité, les avantages et même la définition du Cloud Computing seront mis en péril.

Au cours de ce mémoire, nous avons fait une étude sur la sécurité d'un Cloud Computing, nous avons commencé par donner les définitions de base nécessaires à la compréhension du Cloud, ses différents types (privée, public, hybride), ses services (IaaS, PaaS, SaaS) et ses éléments constitutifs (la virtualisation, datacenter), et nous avons par la suite présenté et détaillé les différents mécanismes de sécurité d'un Cloud Computing.

L'objectif de notre projet était d'étudier et de mettre en place deux solutions libres de Cloud Computing pour pouvoir comparer réellement ces dernières. En effet, nous avons puinstaller et configurer OpenStack et OpenNebulapar la suite nous avons examiné leur aspect sécuritaire en étudiant quelques failles de sécuritéqui leur sont liées.

Ce projet étant très ambitieux, nous avons rencontré de nombreux problèmes lorsde l'installation et de la configuration d'OpenStack et d'OpenNebula, notamment en ce qui concernele réseau et l'infrastructure. Tousces problèmes nous ont montré la complexité de configurer une solution de CloudComputing, et leur résolution nous a souvent retardé.

Ce travail a été pour nous une chance de découvrir un environnement aussi complexe vis-à-vis de la miseen place d'une solution Cloud, ce qui nous a permis d'approfondir nos connaissances dans le domaine de la virtualisation et du Cloud Computing.

En perspective, il serait plus intéressant de déployer une infrastructure IaaS comme OpenStack ou OpenNebula sur un serveur physique et pouvoir implémenter des solutions de sécurité comme le cryptage de données ou la gestion des accès réseaux sans avoir à faire aux limitations que nous imposent les environnements de para-virtualisation comme VMware workstation ou VirtualBox.

Bibliographie

ABDELFATTAH, H. (2016). ANALYSE DE PERFORMANCE PAR PLANS EXPÉRIMENTAUX D'INTERGICIELS. MONTRÉAL.

Ben Mahmoud. (2015). virtualisation&cloud computing. Institut Supérieur des Études Technologiques de Tozeur.

Benkemoun, A. (2009). Récupéré sur https://www.antoinebenkemoun.fr/2009/07/la-virtualisation-materiel-assistee/

BERKANI Nassima, M. S. (2015/2016). La sécurité des données dans le Cloud Computing. Béjaïa.

Biulding. (2018). Récupéré sur http://www.nemengine.com/case/san/9.html

Boisaubert, K. (2017). Protection des données et cloud computing. Institut Léonard de Vinci, France: Institut Léonard de Vinci.

David. (2013). Récupéré sur https://blog.ineat-conseil.fr/2013/01/restful-authentication/

David GELIBERT, F. S. (2012). *La sécurité et la virtualisation .* Lyon: Polytech Lyon.

David GELIBERT, F. S. (2010). *sécurité du cloud informatique.* Lyon.

DÉON, S. (2015). *OpenStack.*

Foulon, D. (2016). *Étude et mise en place d'une plate-forme de cloud.*

GERVAISE, D. G. (2012). *Sécurité et Virtualisation.* Lyon: Polytech Lyon.

Infoblox. (2013). *Vaincre les attaques DoS/DDoS en temps réel.*

julien. (2019). Récupéré sur https://pandorafms.com/docs/index.php?title=File:Opennebula-logo.png

KARTIT, Z. (2016). Contribution à la sécurité du Cloud Computing : Application des algorithmes de chiffrement pour sécuriser les données dans le Cloud Storage. Rabat: Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat – Maroc.

KHENOUS, H. M. (2013). [mémoire] Etude et Mise en oeuvre d'une solution Cloud. Bejaia .

Libre, J. D. (2012). Récupéré sur http://jeudisdulibre.be/2013/02/07/mons-le-21-fevrier-presentation-dopenstack/openstack-logo5/

Marlise, M. M. (2010). les reseaux SAN comme solution de stockage et de protection des données. Cameroun.

Mehdi, S. H. (2018/2019). Mise en place et étude de la sécurité d'une plateforme IaaS basée sur OpenStack. Oran, Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF.

Pascal Faure. (2015). *Guide sur le Cloud Computing et les datacenters .*

Patil, J. (2016). Récupéré sur https://www.quora.com/What-is-a-man-in-the-middle-attack

PETER, L. (2016). Récupéré sur https://www.supinfo.com/articles/single/1765-introduction-aux-differents-types-hyperviseurs

RELAZA, T. J. (2016). Sécurité et disponibilité des données stockées dans les nuages. Toulouse: Université Toulouse 3 Paul Sabatier.

Robert Jaques, R. S. (2012). SLA – Service Level Agreement. Genève.

Seddiki, M. S. (2015). Allocation dynamique des ressources et gestion de la qualitée de service. *Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)*. Université de Lorraine: Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA).

senges, C. (2018). Récupéré sur https://openclassrooms.com

Zohra, F. (2016). gestion de confiance dans le cloud computing. oran: Laboratoire d'informatique d'oran.