Table des matières

LISTE DES FIGURES :

LISTE DES TABLEAUX :

ACRONYMES:

VPN= Virtual Private Network

ERP= Enterprise Resource Planning

AD =Active Directory

VM= Virtual Machine

VMM= Virtual Machine Manager

VIM= Virtual Infrastructure Manager

CPU= Central Processing Unit

SOA= Service Oriented Architecture

Iaas= Infrastructure as a Service

PasS= Platform as a Service

SaaS= Service as a Service

KVM= Kernel-based Virtual Machine

VLAN= Virtual LAN

SDK= Software Development Kit

HA= High Availability

VNC= Virtual Network Computing

NVP= Network Virtualization Platform

SAN= Storage Area Network

SPOF= Single Point Of Failure

NAT= Network Address Translation

DHCP= Dynamic Host Configuration Protocol

LE CAHIER DE CHARGE:

Missions

Mise en place d'une plateforme de virtualisation et déploiement d'une solution Cloud privé open source.

- Etude technique de la plateforme existante.
- Etude des différentes solutions de virtualisation
- Etude de différentes solutions Cloud open source disponibles.
- Choix de la solution de virtualisation.
- Choix de la solution Cloud Open source à déployer
- Déploiement des machines virtuelles et migration des serveurs physique existants vers la plateforme virtuelle
- Déploiement de la solution Cloud
- Test de bon fonctionnement de la plateforme

Technologies: VMware Esxi, vSphere, vCenter, OpenStack, Snapshot,…

INTRODUCTION GENERALE:

Surfer dans les nuages à toujours été le rêve des hommes. Ce domaine était longtemps réservé aux poètes, et leur offrait la possibilité de s'évader des problèmes quotidiens qui les assaillaient. Mais depuis le jour où les avions et autres objets volants, conquirent le grand espace ciel, une grande majorité de gens pouvait sillonner plusieurs types de nuages. Ce rêve est également devenu réalité, dans le domaine de l'informatique. Actuellement, on parle beaucoup de Cloud Computing, et ce n'est pas par hasard.

En effet, le matériel et les systèmes informatiques associés évoluent pour satisfaire les besoins de calcul qui s'accroîent chaque jour. Pour cela, les chercheurs et les technologues travaillent dur et péniblement afin de simplifier la gestion des infrastructures informatiques, la continuité d'activité et la réduction du coût qui représente une stratégie des entreprises d'aujourd'hui pour se focaliser sur leur métier spécifique marqué par une concurrence aiguë.

Les solutions présentes sur le marché actuellement sont souvent complexes et très coûteuses. En parallèle, les évolutions quotidiennes au niveau des infrastructures réseaux et les systèmes d'information ont conduit à l'arrivée de nouveaux périphériques et technologies, ce qui rend la tâche de déploiement et de gestion encore plus difficile à assurer. De toutes ces contraintes est né le besoin de « la virtualisation » qui est survenu comme une solution révolutionnaire à un grand nombre de défis auxquels l'entreprise devait faire face.

Assurément, remplacer un parc de serveurs physiques par une architecture virtualisée constitue une véritable révolution dans la conception des systèmes d'information. L'objectif de cette technique est d'augmenter la flexibilité et d'améliorer l'utilisation des ressources matérielles à moindre coût, tout en assurant la performance et la disponibilité des services. Cette notion a évoluée encore plus, et a fait apparaître un nouveau terme : le Cloud Computing. Dans les dernières années, cette terminologie, qui se base sur la virtualisation, a envahi le monde de la commercialisation, et s'est présentée comme une solution inédite arrivant jusqu'aux clients distants. Toutefois, cette tendance divulgue des défaillances, qui ont été souvent ignorées lors de ces déploiements en raison du coût, ou la complexité de mise en place.

Les entreprises ont besoin de mettre en place plusieurs outils et services pour pouvoir exécuter leurs travaux, afin de bien assurer la gestion des projets, leurs qualités et la productivité des équipes ainsi que l'évolution sûre du produit pour une meilleure compétitivité sur le marché.
C'est sous cette optique, que nous allons réaliser au niveau de ce PFE, une plateforme de virtualisation qui permet de migrer de l'architecture physique existante vers un environnement virtuel et déployer un Cloud privé.

Organisation de rapport :

Chapitre1- Contexte de projet : Ce chapitre explicite les notions sur lesquelles s'appuie le projet. Il présente l'organise d'accueil, ces services et son domaine d'activités, la problématique, critique de l'existant et la solution proposée, ainsi que les spécifications des besoins et étude préliminaire

Chapitre2- Etat de l'art : Ce chapitre présente les notions fondamentales, le mécanisme et les avantages de la virtualisation, ainsi des généralités à propos du Cloud Computing, historique, les catégories des services, les modèles de déploiement, les avantages, les inconvénients du Cloud et les différentes solutions disponibles pour la création d'un environnement Cloud.

Chapitre3- Processus de développement et méthodologie : ce chapitre représente le modèle de développement sashimi, une étude comparative entre les différentes solutions Cloud open source disponibles et étude des différents hyperviseurs, de plus les phases de conception, d'implémentation et les tests.

Finalement, on finit ce travail par une conclusion générale, limitations et perspectives.

Chapitre 1: Contexte général et spécification des besoins

Chapitre 1: Contexte général et spécification des besoins

I. Thématique

L'efficacité de l'utilisation du Cloud Computing au sein des entreprises n'est plus sujet à discussion vue qu'il a fait cette preuve durant les dernières années, d'où l'enthousiasme du groupe SERVICOM pour implémenter un pour, en premier lieu, centraliser ces serveurs et du coup faciliter la gestion et la maintenance du système d'information et en plus améliorer la qualité de l'échange entre les différentes branches du groupe.

1. Présentation de l'entreprise d'accueil : SERVICOM

Figure 1: Logo Sevicom

Créée en mars 2003, les principaux secteurs d'investissement du groupe SERVICOM sont :

- les travaux d'infrastructure en télécommunications,
- les travaux du génie civil (routes, construction, etc.),
- la climatisation et les ascenseurs.

Depuis mai 2009, SERVICOM est cotée à la bourse de Tunis. Le siège de l'entreprise est en Tunisie mais ses investissements couvrent la région du Maghreb et la méditerranée.

2. Mission

Forte de son savoir-faire et de son expertise en matière de travaux d'infrastructure en télécommunications, SERVICOM propose une offre globale de compétences s'étalant de l'amont à l'aval et couvrant aussi bien l'ingénierie que les travaux, la maintenance et l'exploitation des Réseaux Télécom.

Figurant parmi les entreprises leaders du secteur, SERVICOM est le prestataire de prédilection de la plupart des institutions publiques, opérateurs téléphoniques et promoteurs immobiliers.

Les quatre départements de SERVICOM sont :

- Réseaux en cuivre et en fibre optique.
- Réseaux d'entreprise.
- Equipements d'énergie.
- Serveurs et accessoires

Tableau 1: Présentation SERVICOM

Adresse	1, Avenue Abou El Kacem Echebbi
Code postal	2080
Ville	Ariana
Pays	TUNISIA
Téléphone	71822327
Fax	71822327
E-mail	Contacts@servi.com.tn
Site web	www.servi.com.tn
PDG	Mr Majdi Zarkouna
Année de création	2003
Effectif	80
Secteur d'activité	Réseaux & Télécommunications

3. Organisation et structure :

La structure de SERVICOM se manifeste par sa nature hierachico-fonctionnelle. En effet, par son caractère, elle est organisé hiérarchiquement mais les impératifs de son activité font qu'il est organisé aussi par fonctions et ce, dans un souci de compétitivité et d'efficacité.

J'ai effectué mon stage au sein de la **direction Informatique**, dans le **département gestion des projets.**

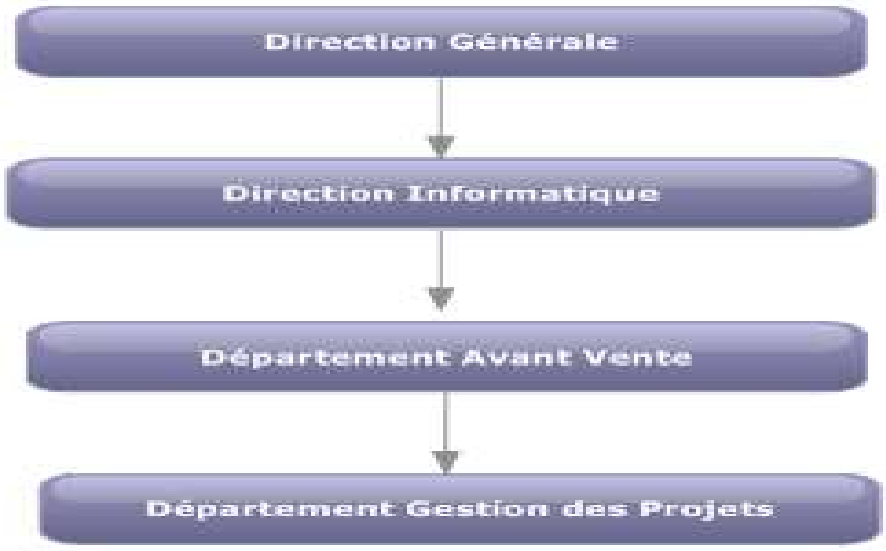

Figure 2: Organigramme de département gestion des projets

Présentation de la direction informatique

La Direction Informatique a le rôle d'étudier, mettre en place, et maintenir des systèmes informatiques fiables (matériel et logiciel) permettant d'autoriser les procédures administratives et techniques, développer des applications de gestion permettant d'accroître les rendements et d'aider à la prise de décision conformément aux besoins exprimés par l'utilisateur et par les clients

Département Gestion des Projets

Ce département prend en charge plusieurs missions notamment :

- Contribuer à la définition de la stratégie de développement du SI en cohérence avec celle de l'entreprise.
- Développer et maintenir l'infrastructure technique des systèmes d'informations.
- La réalisation et l'exécution des projets gagnés dans le secteur public et privé
- Assurer les transferts des compétences aux clients
- Maintenir les maintenances curatives et préventives des clients
- Assurer la production informatique, et instaurer progressivement une gouvernance des SI au niveau de SERVICOM.

II. Présentation du projet

Ce projet est réalisé dans le cadre de notre projet de fin d'étude, il nous permet d'exploiter les avantages du Cloud ainsi ceux de la virtualisation.

Dans ce projet, nous allons mettre en place une infrastructure virtuelle et déployer un Cloud privé SERVICOM.

1. Problématique

La solution proposée doit offrir tous les outils dont le groupe **SERVICOM** a besoin pour assurer le bon déroulement des tâches et la communication inter/intra départements.

2. Critique de l'existant

Nous avons étudié en premier lieu l'architecture déjà mis en place pour pouvoir identifier les lacunes du système ainsi que l'inventaire de l'existant qui pourra être utilisé par la suite dans ce projet.

Le groupe a deux sites principaux : l'un est situé à l'Ariana qui est un siège social et le second site est situé à la Charguia I qui est un siège technique. La communication entre les deux sièges est assurée par une liaison VPN. Les sites disposent d'un « Active Directory », d'un « ERP » et d'un « WORKFLOW » qui leur sont propres.

La figure qui suit illustre l'architecture déjà en place.

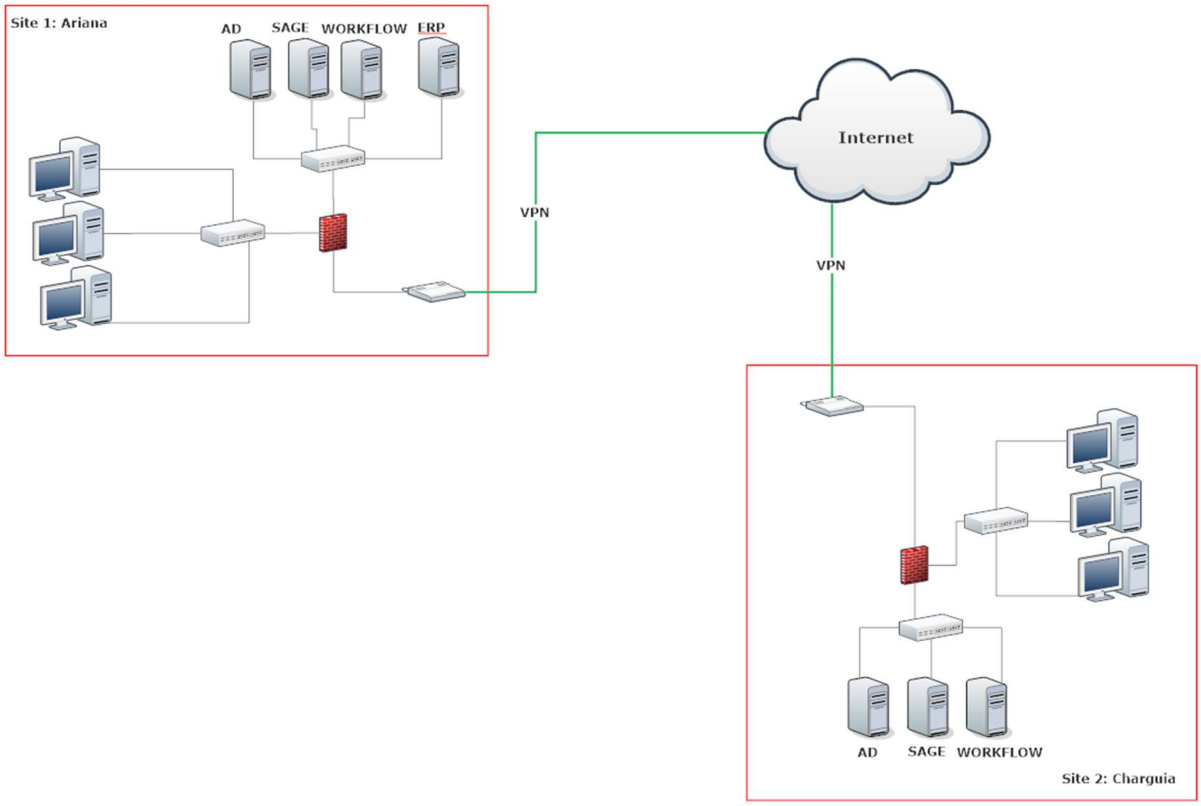

Figure 3: Architecture déjà mise en place

Le système d'information qui été déjà en place présente certaines lacunes. Le fait d'avoir deux AD, deux ERP et DEUX WORKFLOW propres à chaque site rend la coordination entre les deux ambigus. En plus, il faut assurer deux staffs techniques pour l'administration et la maintenance des serveurs.

3. Solution proposée

Nous allon centraliser le système d'information dans le second site. Le regroupement de tout les serveurs dans un seul site nous permet d'installer une architecture basée sur la virtualisation et le cloud computing ce qui nous permettera entre autre d'assurer la haute disponibilité des services et des outils dont la societé a besoin.

4. Plannig adopté

La planification est parmi les phases d'avant projet les plus importantes. Elle consiste à déterminer, ordonnancer les tâches du projet et à estimer leurs charges respectives. Mon projet s'est articulé en quatre Mois (Mars Avril Mai Juin juillet Aout septembre : 2015).

- Ressources humaines

  Les gens qui ont participé à ce travail sont :
  Encadrant-interne Mr . HELMI BEDOUI
  Encadrant-externe Mr . MHAMDI MOHAMED
  Stagiaire : Maher JENDOUBI

- Ressources logiciels

  Durant mon projet j'ai utilisé une panoplie d'outils informatiques à savoir :

  Vmware ESXI, VMware vSphere Client, VMware vCenter, Windows Server 2008R2 Windows 7 Pro, Ubuntu 14.04 LTE, SQL Server 2008, OpenStack.

- Ressources budgétaires

Pour mener à bien mon projet, j'ai commencé par un planning prévisionnel, sauf que j'ai passé beaucoup plus de temps dans la partie Conception et Processus de développement et méthodologie

- Chronogramme

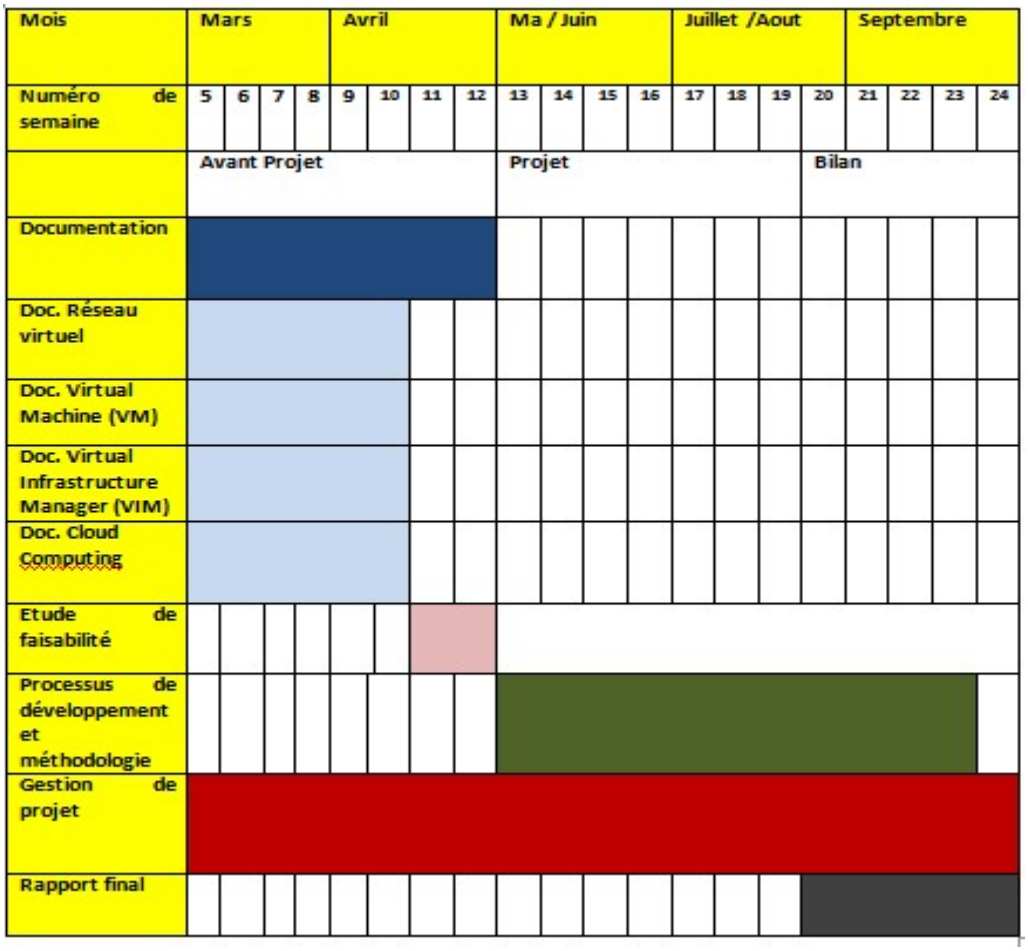

Figure 4: Planning prévisionnel de travail

III. Spécification des besoins et études préliminaire

L'étude préliminaire consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte. Elle prépare les activités plus formelles de capture des besoins fonctionnels et techniques.

1. Projet à réaliser

L'objectif initial de notre projet consiste à concevoir une plateforme virtuelle pour migrer des serveurs physiques existants dans un environnement virtuelle et permettant de réaliser le déploiement de la plateforme Cloud Computing privée.

1.1. Recueil des besoins

On a effectué plusieurs recherches pour identifier les besoins de la solution et ceci afin de répondre aux attentes des utilisateurs. Les besoins se divisent en deux grandes catégories : les besoins fonctionnels et les besoins non fonctionnels.

1.2. Besoins fonctionnels

Pour réussir la tâche du recueil des besoins fonctionnels, nous avons commencé par interviewer les différents employés de la société selon un questionnaire semi-directif. Ces interviews nous ont permis d'identifier les besoins fonctionnels suivants :

- La migration vers une plateforme virtuelle pour optimiser les ressources et garantir la fiabilité du système
- La migration des serveurs physiques vers des machines virtuelles
- L'évolutivité de l'infrastructure
- La continuité de l'activité
- La gestion des sauvegardes des machines virtuelles déjà installées.
- Le déploiement d'une plateforme du Cloud Computing privée pour offrir des services à toutes les filiales du Groupe Servicom
- Suggérer la meilleure architecture pour le déploiement d'OpenStack

1.3. Besoins non fonctionnels

- Intégration d'un outil de supervision pour toute la plateforme de virtualisation
- Développer et vendre des applications Cloud
- La création d'un module de facturation pour les clients Cloud

IV. Conclusion

Dans ce premier chapitre, nous avons présenté l'entreprise d'accueil à savoir le groupe SERVICOM. Puis, nous avons présenté le système déjà mis en place et nous avons proposé une solution. En fin nous avons distingué les différents besoins que notre application a imposée. Ces besoins seront détaillés dans le chapitre de conception.

Chapitre 2 : Etat de l'art

Chapitre 2 : Etat de l'art

I. Introduction

Au niveau de ce chapitre, nous allons présenter les notions fondamentales de la virtualisation et du Cloud Computing. Nous commencerons par expliquer son principe, les différents services qu'il fournit, ses modèles de déploiement ainsi que ses avantages et inconvénients. Par la suite, nous allons présenter les différentes solutions du Cloud Computing existantes et nous allons choisir la solution que nous allons adopter.

II. La virtualisation

La virtualisation [1] recouvre l'ensemble des techniques matérielles et ou logiciels qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d'exploitation, plusieurs instances différentes et cloisonnées d'un même système ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes.

Les intérêts de la virtualisation sont : l'utilisation optimale des ressources, l'économie sur le matériel par mutualisation, l'allocation dynamique de la puissance de calcul, la facilité d'installation, de déploiement et de migration des machines virtuelles.

1. Mécanisme

- Un système d'exploitation principal appelé « système hôte [1] » est installé sur un serveur physique unique. Ce système sert d'accueil à d'autres systèmes d'exploitation.
- Un logiciel de virtualisation appelé « hyperviseur » est installé sur le système d'exploitation principal. Il permet la création d'environnements clos et indépendants sur lesquels seront installés d'autres systèmes d'exploitation « systèmes invités ». Ces environnements sont des « machines virtuelles ».
- Un système invité est installé dans une machine virtuelle qui fonctionne indépendamment des autres systèmes invités dans d'autres machines virtuelles. Chaque machine virtuelle dispose d'un accès aux ressources du serveur physique (mémoire, espace disque…).

2. Avantages

- La virtualisation des systèmes informatiques présente de nombreux avantages :
- Déploiement rapide des applications
- Niveaux de service supérieur et disponibilité accrue des applications
- Taux d'utilisation supérieur des investissements dans l'infrastructure
- Évolutivité rapide et flexible
- Réduction des coûts énergétiques, d'infrastructure et des installations
- Diminution des frais de gestion
- Accès aux applications et aux données des bureaux en tout lieu
- Sécurité informatique renforcée

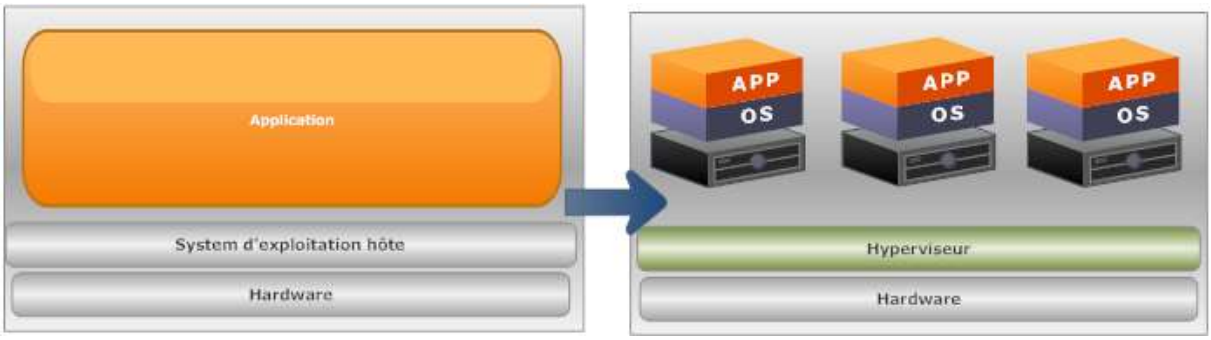

Figure 5:Comparaison entre architecture physique et virtualisée

3. Virtual Machine Manager (VMM) / Hyperviseur

La mise en œuvre d'une machine virtuelle (Virtual machine ou VM) nécessite l'ajout d'une couche logicielle à la machine physique. Cette couche d'abstraction se place entre le matériel et le système d'exploitation et s'appelle hyperviseur ou moniteur de machine virtuelle (VMM). L'hyperviseur agit comme un arbitre entre les systèmes invités: il attribue du temps processeur et des ressources à chacun, redirige les requêtes d'entrées-sorties vers les ressources physiques, veille au confinement des invités dans leur propre espace.

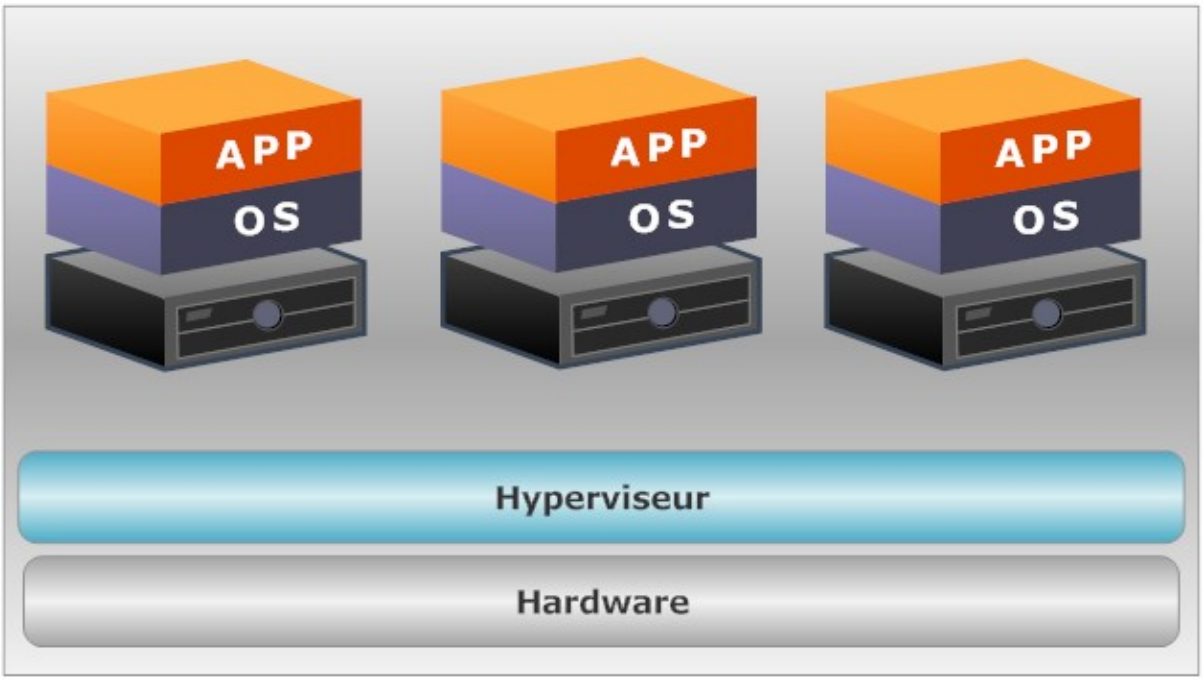

Figure 6:Hyperviseur

4. Virtual Machine (VM)

Une machine virtuelle est un environnement d'exécution virtuel créé à partir d'une image présentant un modèle pour instancier cette machine. Le cycle de vie d'une machine virtuelle comporte six phases : create, suspend, resume, save, migrate et destroy. Les machines virtuelles tournant sur un même noeud physique sont gérées et contrôlées par le VMM.

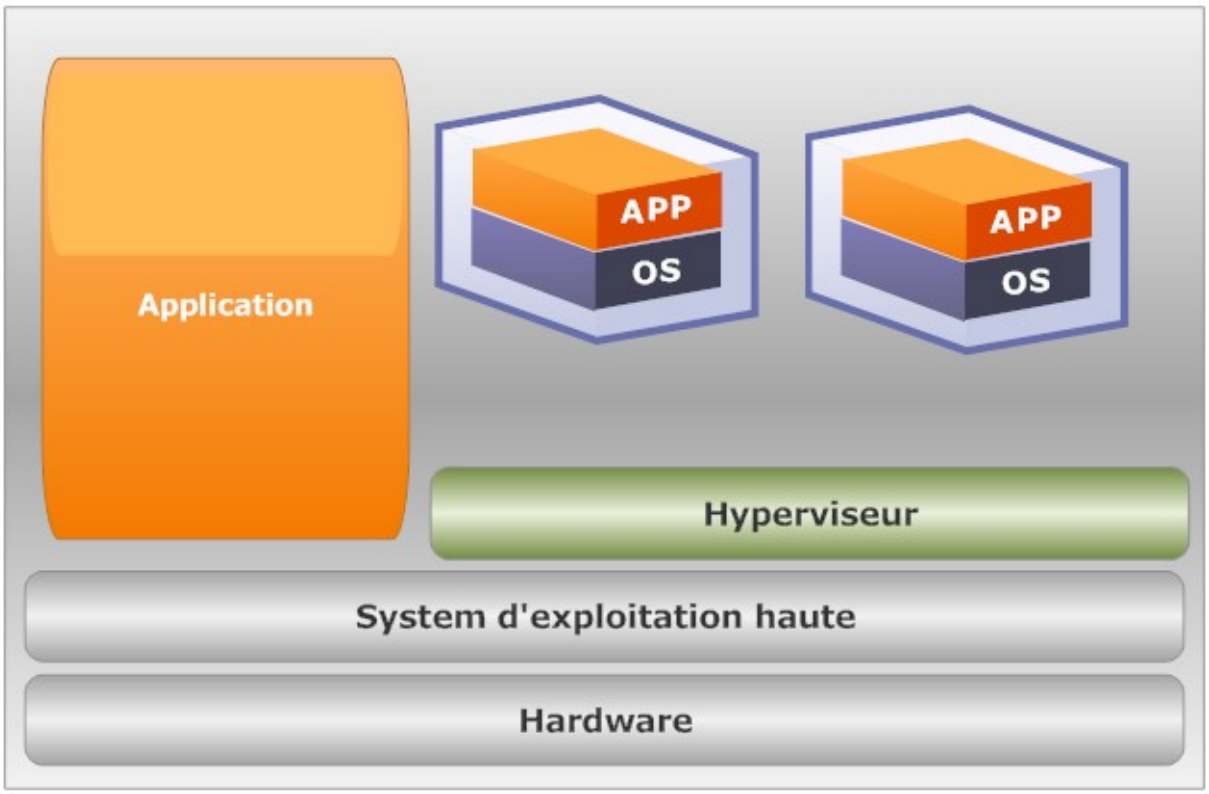

Figure 7: Machine virtuelle

5. Virtual Infrastructure Manager (VIM)

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

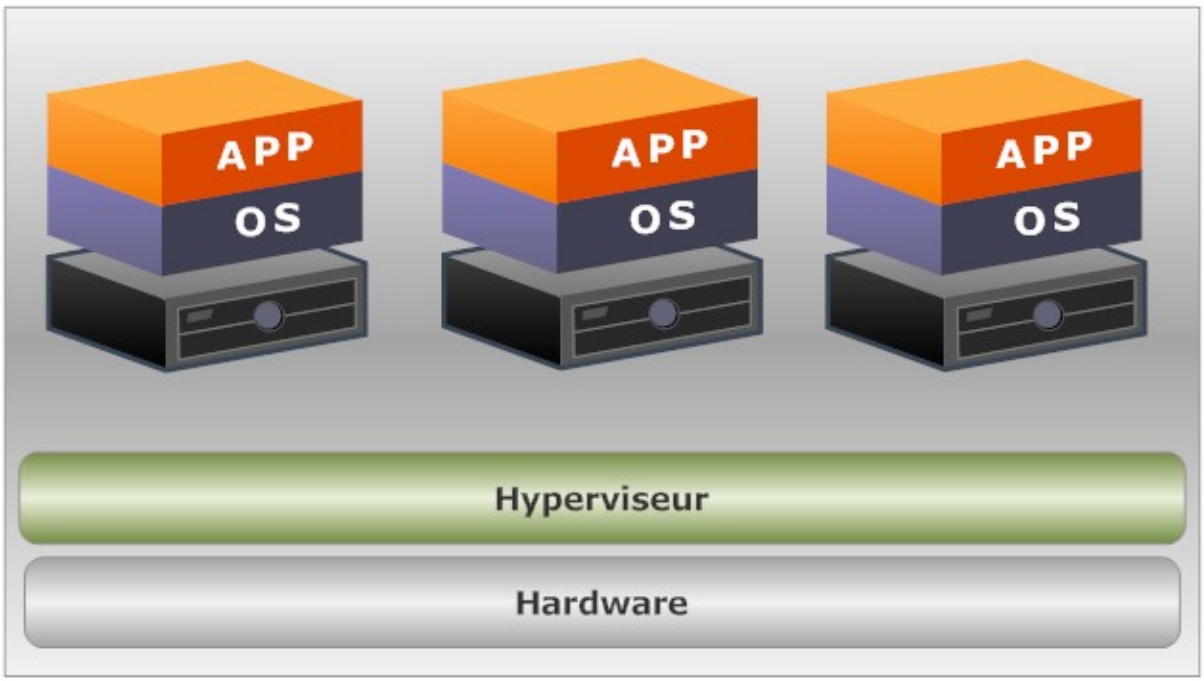

Figure 8: Infrastructure virtuelle

La figure 8 décrit l'infrastructure virtuelle avec ces différentes entités nécessaires pour le déploiement et la gestion des machines virtuelles. Elle est basée effectivement sur la présence de l'hyperviseur et du gestionnaire de l'infrastructure.

6. Réseau virtuel

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

Dans les hyperviseurs de base, une machine virtuelle possède une ou plusieurs cartes d'interfaces réseau virtuelles. Le trafic réseau sur toute interface virtuelle est commuté par l'hyperviseur vers l'interface réseau physique en utilisant une solution logicielle, matérielle ou les deux ensembles.

Un commutateur virtuel (bridge) comme présenté dans la figure 9 est un logiciel intégré à l'hyperviseur qui se comporte comme un commutateur matériel. Ce logiciel permet aux machines virtuelles (VM) de s'intégrer au réseau avec d'avantage de souplesse, notamment sans se soucier du nombre de cartes physiques.

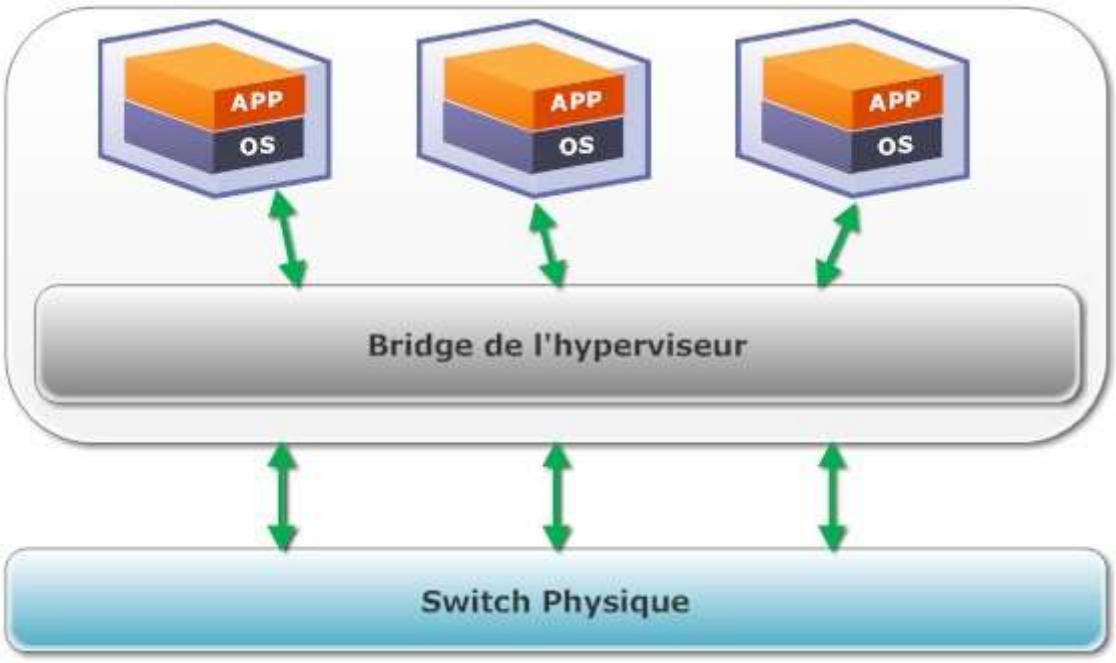

Figure 9: Concept du réseau virtuel

Dans les hyperviseurs de base, une machine virtuelle possède une ou plusieurs cartes d'interfaces réseau virtuelles. Le trafic réseau sur toute interface virtuelle est commuté par le commutateur vers l'interface réseau physique.

III. Les concepts et définitions du Cloud Computing

1. Définitions et généralités

Le Cloud Computing [2], ou informatique dans les nuages, est une manière de fournir et d'utiliser les aptitudes des systèmes informatiques, basée sur les nuages (cloud en anglais). Ce dernier est un parc de machines, d'équipements réseau et de logiciels maintenus par un

fournisseur, que les consommateurs peuvent utiliser en libre-service via un réseau informatique, le plus souvent Internet.

Le National Institute of Standards and Technology a donné une définition succincte qui détermine les principes de base du cloud computing: « L'informatique dans les nuages est un modèle permettant d'établir l'accès via un réseau de télécommunication à un réservoir partagé de ressources informatiques standards configurables (réseau, serveurs, stockage, applications et services) qui peuvent être rapidement mobilisées et mises à disposition en minimisant les efforts de gestion ou les contacts avec le fournisseur de service. »

Donc, le Cloud Computing est un concept qui consiste à la mise à disposition de ressources de technologies de l'information sous forme de services à la demande. De cette façon, les applications et les données ne se trouvent plus sur des serveurs locaux ou sur le poste de l'utilisateur, mais dans le Cloud composé d'un certain nombre de serveurs distants, interconnectés au moyen d'une large bande passante.

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

La figure suivante illustre le concept du cloud computing.

Figure 10: Cloud computing[3]

2. Historique

Le Cloud computing est un concept assez récent. Sa première énonciation date de 1960 (John McCarthy), mais sa réelle mise en application a commencé au début des années 2000. Salesforce.com fut le premier hébergeur de Cloud en 1999, suivi en 2002 par Amazon.

Le Cloud Computing met en œuvre l'idée de l'informatique utilitaire du type service public, proposée par John McCarthy en 1961 qui suggère que la technologie informatique partagée pourrait construire un bel avenir dans lequel la puissance de calcul et même les applications spécifiques pourraient être vendues comme un service public.

L'apparition du Cloud Computing vient d'une évolution de certaines technologies telles que la virtualisation du matériel informatique, les services web, ou l'architecture orientée services SOA (Service Oriented Architecture).

La virtualisation a été la première pierre de l'ère du Cloud Computing. En effet, cette notion permet d'optimiser les ressources matérielles en les partageant entre plusieurs environnements dans le but de pouvoir exécuter plusieurs systèmes « virtuels » sur une seule ressource physique et fournir une couche supplémentaire d'abstraction du matériel.

Le Cloud computing est donc la juxtaposition de ces technologies pour passer à la vitesse supérieure sur l'exploitation de données à travers Internet.

3. Caractéristiques

Le Cloud computing s'appuie sur une architecture client-serveur, il possède des caractéristiques qui le distinguent des autres technologies:

- Les ressources et les services fournis aux clients sont souvent virtuels et partagés par plusieurs utilisateurs ;
- Les serveurs qui forment le Cloud sont hébergés dans des data ''centers'' externes. La plupart de ces centres comportent des dizaines de milliers de serveurs et des moyens de stockage pour permettre des montées en charge rapides ;
- Les services sont fournis selon le modèle "pay-per-use" où la facturation est calculée en fonction de la durée et de la quantité de ressources utilisées.

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

4. Les catégories des services

Le Cloud computing peut être décomposé en trois catégories de services

- Infrastructure en tant que service (IaaS, Infrastructure as a Service).
- Platforme en tant que service (PaaS, Platform as a Service).
- Application en tant que service (SaaS, Software as a Service).

La figure ci-après illustre les différentes catégories de services d'un Cloud computing.

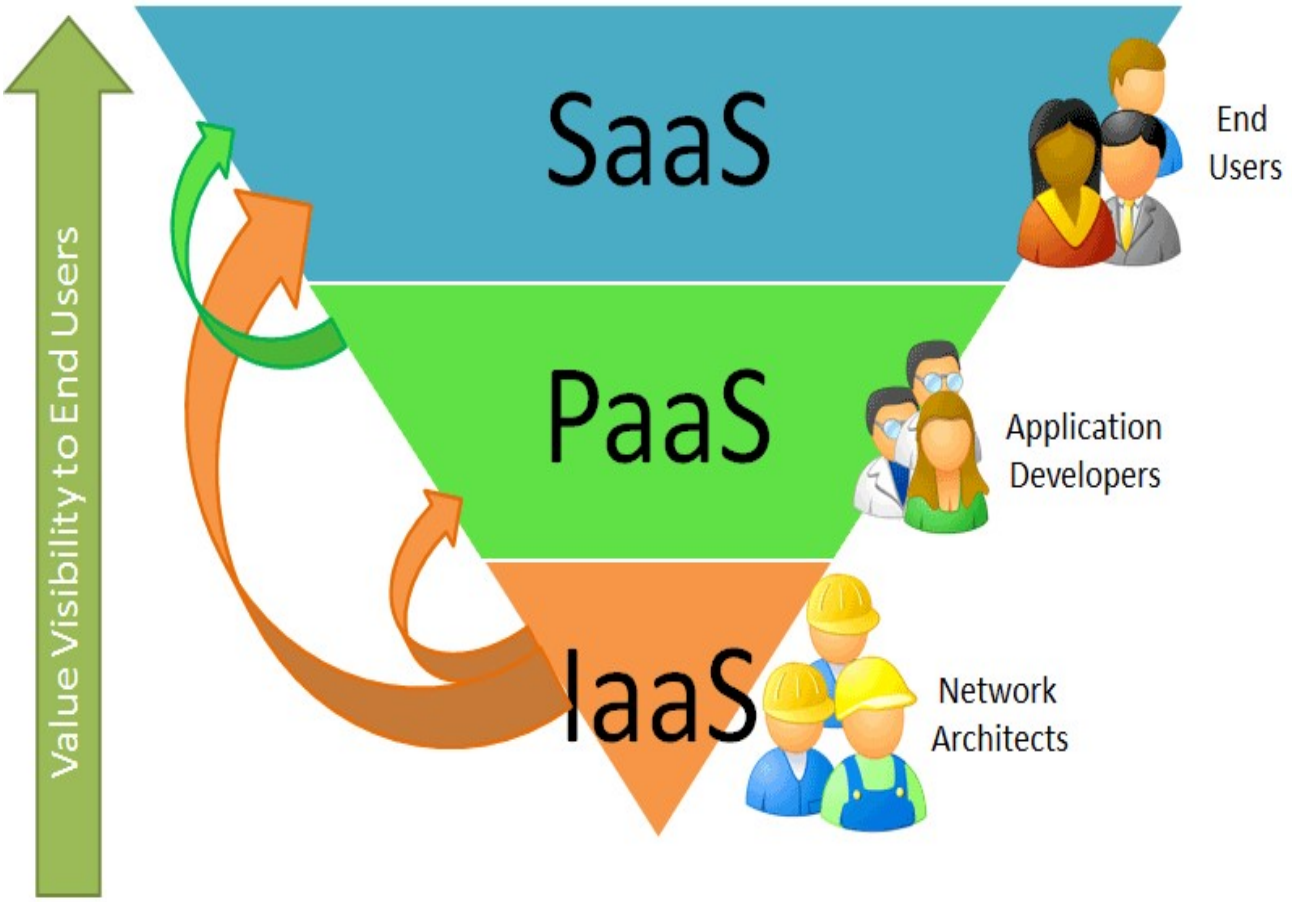

Figure 11: Les services du Cloud computing[4]

4.1. Infrastructure en tant que service (IaaS)

C'est le service de plus bas niveau du Cloud computing. Il permet de disposer d'une infrastructure à la demande, pouvant héberger et exécuter des applications, des services ou encore stocker des données. Plutôt que d'acheter des serveurs, des logiciels, de l'espace dans un centre de traitement de données, les clients n'ont plus qu'à louer ces ressources auprès des prestataires de services.

Le service est tarifié en fonction de l'utilisation et de la quantité de ressources consommées. De ce fait, le coût reflète typiquement le niveau d'activité de chaque client. C'est une évolution de l'hébergement Internet qui se différencie des anciens modes de fonctionnement.

L'IaaS offre une grande flexibilité, avec une administration à distance, et permet d'installer tous types de logiciels. En revanche, cette solution nécessite la présence d'un administrateur système au sein de l'entreprise, comme pour les solutions classiques.
Parmi les prestataires d'IaaS, on peut citer : Amazon avec EC2 ou Orange Business Services avec Flexible Computing.

4.2. Plateforme en tant que service (PaaS)

La plate-forme en tant que service, située juste au-dessus de l'IAAS, est la plate-forme d'exécution, de déploiement et de développement des applications. C'est une solution

complète sans téléchargement ou installation de logiciels destinés aux développeurs, responsables informatiques ou utilisateurs finaux disponible immédiatement via Internet.

PaaS se distingue des hébergeurs classiques par

- Une grande abstraction de la plate-forme : le système d'exploitation et les outils d'infrastructure sont sous la responsabilité du fournisseur tandis que le consommateur a le contrôle des applications et peut ajouter ses propres outils.
- Une architecture à très haute disponibilité basée sur des centres de données répartis dans le monde entier, et garantissant une grande résistance aux sinistres (inondations, incendies, etc…).

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

4.3. Application en tant que service (SaaS)

SaaS, application en tant que service, est le service du plus haut niveau du Cloud computing. C'est un modèle de déploiement d'application dans lequel un fournisseur loue une application clé en main à ses clients en tant que service à la demande au lieu de leur facturer des licences. De cette façon, l'utilisateur final n'a plus besoin d'installer tous les logiciels existants sur sa machine de travail ce qui réduit par la suite la maintenance en supprimant le besoin de mettre à jour les applications.

Ce type de modèle transforme les budgets logiciels en dépenses variables et non plus fixes et il n'est plus nécessaire d'acquérir une version du logiciel pour chaque personne au sein de l'entreprise.

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

La figure ci-dessous illustre, à titre d'exemple, certains prestataires Cloud selon les différentes catégories des services.

5. Les modèles de déploiement

Chacun de ces services peut être déployé de trois manières différentes

La figure ci-dessous décrit les trois modèles de déploiement d'un Cloud

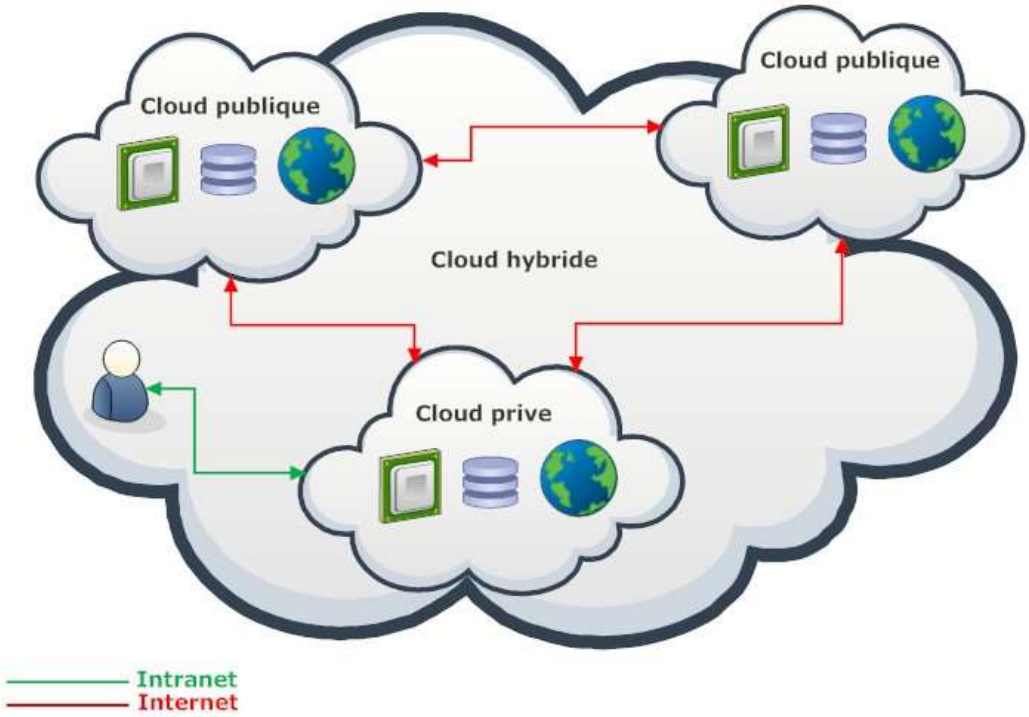

Figure 12: Prestataires de Cloud selon les catégories des services

5.1. Le Cloud public

Ce type de Cloud est créé par un organisme spécialisé qui met à disposition ses infrastructures, ses ressources pour des entreprises. Ainsi, ces entreprises consommatrices de services utilisent et payent à la demande des services dont elles ont besoin. Les fournisseurs de services Cloud Computing garantissent une disponibilité et une qualité de service à travers un contrat signé avec le consommateur du service.

5.2. Le Cloud privé

Les offres de Cloud privé, contrairement à celles du Cloud public, se distinguent par leur aspect dédié, c'est à dire destinés à satisfaire les besoins propres d'une seule entreprise. On distingue deux types de Cloud privé : interne et externe.

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

5.3. Le Cloud hybride

C'est le résultat d'une combinaison de deux ou plusieurs nuages (public ou privé). Ces nuages communiquent donc ainsi, et forment un Cloud hybride. Les différents nuages qui le composent restent des entités indépendantes à part entière, mais sont reliés par des standards ou par des technologies propriétaires qui permettent la portabilité des données et des applications déployées sur les différents nuages.

6. Les avantages et les inconvénients du Cloud

Le tableau ci-dessous résume les principaux avantages et inconvénients du Cloud Computing.

Tableau 2: Les avantages et les inconvénients du Cloud Computing

Avantages	Aucun investissement préalable. Service d'une grande disponibilité. Facturation à la consommation. Version toujours à jour et identique pour tous les utilisateurs (cas du SaaS). Architecture sur mesure adaptable selon les besoins. Accès simplifié utilisant des navigateurs web.
Inconvénients	Budget élevé : les besoins en bande passante peuvent faire exploser le budget. Cadre légal : Absence de localisation précise des données. Gouvernance et administration. Manque de confidentialité et sécurité des données.

IV. Conclusion

Au niveau de ce chapitre nous avons clarifié le concept de la virtualisation et du Cloud Computing en mettant en relief l'utilisation, les avantages et les inconvénients. Cela nous a conduits à déterminer notre besoin pour mettre en place une plateforme de virtualisation et du Cloud.

Chapitre 3 : Processus de développement et méthodologie

Chapitre 3 : Processus de développement et méthodologie

I. Introduction

Comme bonne pratique, il est fortement recommander de choisir une méthode d'organisation des phases de développement du projet pour assurer un maximum d'efficacité.

Pour ce projet, nous avons choisi de suivre le modèle sashimi qui est une évolution du modèle en cascade. Dans ce qui suit, nous allons expliquer en quoi consiste ce modèle.

II. Le modèle sashimi

Le modèle sashimi [5] (ainsi appelé car il présente des phases imbriquées, comme l'imbriquèrent du poisson dans la préparation japonaise sashimi) a été créé par Peter DeGrace, aussi appelé « le modèle en cascade avec boucle de retour» ou encore « le modèle en cascade avec évaluation ». Un avantage de ce modèle est qu'il permet de détecter rapidement les erreurs aidant ainsi à améliorer la qualité du produit final tout en réduisant la charge de travail. La phase de conception du projet permet de détecter des problèmes d'implémentation avant de commencer la phase de production. Inversement, comme la phase d'implémentation commence avant la finalisation de la conception, on peut découvrir des erreurs de conception non détectables dans la phase de conception. La méthode itérative introduite avec le modèle sashimi a été créée pour réduire les couts associés au modèle en cascade classique.

Une des plus importantes tâches pour le travail avec la méthode sashimi, est l'extraction et la définition des besoins du projet.

Selon le cadre et le type du projet, nous utiliserons les phases adéquates à utiliser. Pour notre projet, nous allons utiliser quatre phases typiques de ce modèle qui adhèrent à la nature de notre projet vu que notre architecture peut s'agrandir et évoluer avec les tests d'implémentation. Nous avons identifié quelques critères qui nous ont menés à cette méthodologie

- Le projet est voué à une utilisation massive (les employer de tout le groupe) des tests qui pourraient amener à la modification de l'architecture.
- Le projet est dans le contexte du Cloud Computing, la solution devra tirer avantage de l'environnement Cloud.
- Les scenarios de test devraient être revus à chaque fois où les phases du processus d'implémentation seront changées.

Due à ces contraintes, nous avons décidé que le modèle sashimi devrait être le meilleur pour la réalisation de notre projet.

Maintenant, nous allons décrire les phases typiques impliquées dans notre itération. Le modèle sashimi se déroule comme une cascade, il passe d'une étape à une autre avec la différence qu'il peut revenir à l'étape précédente.

Voici un aperçu des quatre étapes de notre méthode

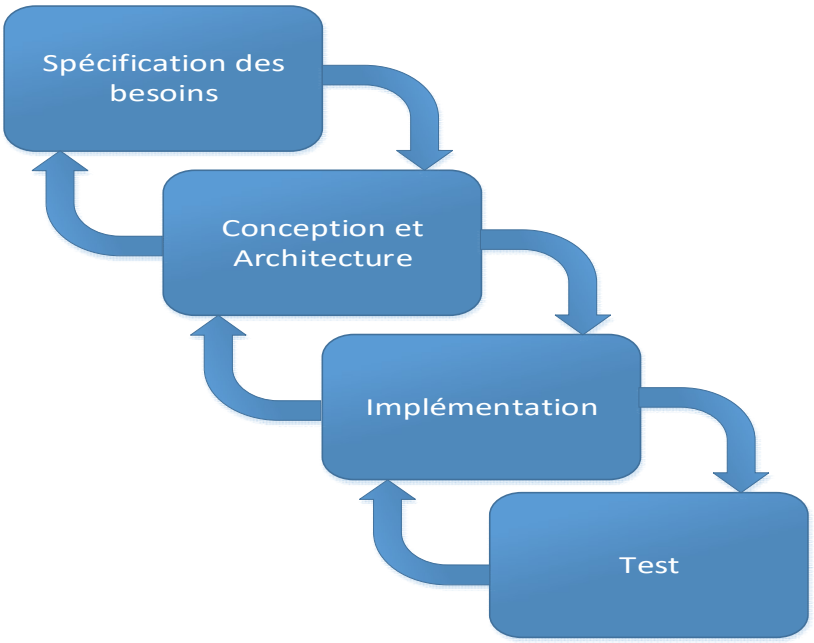

Figure 13: Modèle Sashimi

1. Description des étapes

- Spécification des besoins: l'extraction et l'analyse des attentes d'un projet est la partie la plus importante du modèle en cascade. Pour reconnaitre les buts d'un projet, il faut avoir une communication avec le client et analyser les risques qu'implique le projet
- Conception et Architecture: après avoir identifié les besoins du projet, nous devons concevoir une solution qui assure ces besoins. Ceci implique un nombre d'éléments comme la sécurité et l'administration du système.
- Implémentation: parmi les raisons pour les quelles cette phase de ce modèle est si importante, est que cette partie est celle qui consume le plus de temps et la plus couteuse dans la vie du projet. Une erreur à ce niveau va impliquer une des deux résultats suivants : retravailler cette partie ou un client mécontent. Même si elle aboutit, elle peut conduire à un retard. Comme nous utilisons le modèle sashimi, la phase d'implémentation est imbriquée avec les phases de spécification des besoins et la conception jusqu'à ce que cette phase répond aux exigences des étapes précédente.
- Test: après avoir terminé l'implémentation de la solution selon la conception, les tests sont réalisés pour vérifier que le produit ne contient pas d'erreurs. Ainsi, pour assurer que le produit ou service sont conformes aux exigences du client.

Durant le projet, le chef de projet doit planifier, suivre et contrôler les activités de chaque étape. Ceci rend cette méthode la plus systématique utilisée pour gérer un projet.

L'implémentation de ce projet a été divisée en itération qui sont appelées « boucles » chaque boucle est composé des phases qui ont été décrites dans ce qui précède.

BOUCLE 1

I. Spécification des besoins

Ce projet est le premier de son genre pour le groupe SERVICOM. Nous avons donc besoin de faire une bonne étude pour savoir quel IaaS choisir pour correspondre au besoin du groupe

Le groupe SERVICOM est en constante croissance ce qui impliquera que l'IaaS choisi devra assurer une bonne scalabilité et une évolution rapide pour suivre la croissance du groupe.

L'IaaS mise en place devra être modulable et personnalisable pour correspondre au besoin du groupe.

Dans ce qui suit nous allons procéder au choix de l'IaaS et de l'hyperviseur à utiliser.

1. Présentation IaaS

Il existe deux catégories de solutions Cloud Computing privé, les solutions propriétaires et les solutions libres et gratuites. Dans le monde du Cloud, le logiciel libre est de plus en plus sollicité par les fournisseurs pour offrir des services dédiés. Deux arguments sont souvent mis en avant pour le choix du libre, la gratuité et le développement plus rapide et agile. C'est pour cela, nous nous intéressons dans ce qui suit aux solutions open source.

1.1. Eucalyptus

Eucalyptus [6] est un outil permettant de construire des infrastructures de Cloud computing sur la base de serveurs en cluster. Il est issu d'un projet de recherche de l'université de Californie. Il permet de créer des Cloud IaaS de type privé ou hybride. Les moteurs de virtualisation supportés sont Xen, KVM pour la version open source. Il existe également une version propriétaire commercialisée par la société Eucalyptus Systems. Il apporte des fonctionnalités supplémentaires comme le support de VMware, celui des machines virtuelles Windows et l'intégration SAN.

1.2. OpenNebula

OpenNebula [7], purement open source permet de déployer des Cloud privés, hybrides et publics. Ecrite en C++, Ruby et Shell, elle supporte les plateformes de virtualisation Xen, KVM et VMware ainsi que le service "on-demand" d'Amazon EC2. Le projet est publié sous licence Apache 2.0.

Parmi ses fonctionnalités : gestion centralisée des machines virtuelles et des ressources physiques, répartition des charges, extension des capacités par ajout de serveurs physiques.

Beaucoup de ces solutions sont avant tout des solutions d'infrastructure permettant une gestion simplifiée d'architectures matérielles complexes.

1.3. CloudStack

CloudStack [8], issu du rachat de Cloud.com par Citrix, a été conçu pour permettre le déploiement et la gestion d'importants réseaux de machines virtuelles. Il supporte les principaux moteurs de virtualisation à savoir : vSphere, Oracle VM, KVM, Xen, mais aussi les services de Cloud d'Amazon. Les caractéristiques principales de CloudStack sont les suivantes:

- Réseau virtuel (support VLAN).

- Piscine de ressources (permet aux administrateurs de limiter les ressources virtuelles).
- Routeurs virtuels, un pare-feu, et un équilibreur de charge.
- Migration en direct avec le maintien d'accueil.

1.4. OpenStack

Il est né de la fusion de deux projets portés l'un par la Nasa et l'autre issu de l'offre de l'hébergeur américain Rackspace Cloud Server. L'ensemble de la plateforme est disponible sous licence Apache 2.0.

OpenStack [9] est un logiciel libre qui permet la construction de Cloud privé et public. Il est aussi une communauté et un projet en plus d'un logiciel qui a pour but d'aider les organisations à mettre en œuvre un système de serveur et de stockage virtuel.

Il s'installe sur un système d'exploitation libre comme Ubuntu ou Debian et se configure entièrement en ligne de commande. C'est un système robuste et qui a fait ses preuves auprès des professionnels du domaine.

En comparaison avec les autres produits, OpenStack semble avoir la plus grande et la plus active communauté. Les membres de la communauté sont toujours prêts à aider les autres à trouver des solutions aux problèmes qui se posent.

La technologie est populaire parmi une vaste communauté de spécialistes et est soutenue par des sociétés telles que Cisco, Dell, NASA, Intel, AMD, Citrix, Rackspace, et RightScale.

Les caractéristiques principales d'OpenStack sont les suivantes

- Capacité à gérer les ressources du serveur de produits virtualisés.
- Capacité à gérer des réseaux locaux.
- Gestion de l'image de la machine virtuelle.
- Groupes de sécurité.
- Contrôle d'accès basé sur les rôles.
- VNC proxy via un navigateur Web.

2. Etude comparative des solutions

Afin de caractériser chaque solution open source et assurer le choix adéquat pour la construction d'un Cloud, nous avons réalisé une étude comparative des quatre solutions libres en se basant sur différents critères de classification.

Le tableau ci-dessous résume les principales caractéristiques de chaque solution.

Tableau 3: comparaison IaaS

	Eucalyptus	**OpenStack**	**OpenNebula**	**CloudStack**
Date de sortie	Mai 2008	Octobre 2010	Mars 2008	Mai 2010
Licence	Propriétaire , GPL v3	Licence Apache	Licence Apache	Licence Apache

Langage de programmation	Java, C et Python	Python, JavaScript, XML	C++, C, Ruby, Java, Shell script,	Java, C
Hyperviseur	Xen, KVM	Xen, KVM, VMware, Hyper-V	Xen, KVM, VMware	KVM, Xen, ESXi, et BareMetal
Systèmes d'exploitation supportés	Linux	-Linux, Windows	Linux	Linux
Modèle de déploiement	Cloud public et privé	Cloud public et privé	Cloud privé	Cloud privé et public
Installation	Dépend de l'environnement réseau et matériel	Difficile à installer : plusieurs composants et plusieurs fichiers de configuration	Manuelle, installation facile sur les distributions supportées	Facile à installer avec les paramètres par défaut
Maturité	Aboutie, produit complet avec une interface de gestion web fonctionnelle.	Niveau de maturité avancé avec sa neuvième version	Avancée, deuxième version stable, solution supportée par Debian.	Manque de maturité
Documentation	Correcte, complète mais pas toujours à jour.	Excellente : une documentation officielle disponible et très détaillée.	Complète, documentations, références de tous les fichiers de configuration.	la documentation ne traite pas des questions complexes dans leur intégralité.

2.1. Choix du IaaS

Dans les sections précédentes, nous avons présenté une liste des solutions permettant la création d'un Cloud. Il est à présent question de faire le choix de celui qui nous convient le mieux. Notre choix s'est fixé sur la solution Openstack pour les raisons suivantes

- La solution est arrivée à un niveau de maturité suffisant : une nouvelle version tous les six mois. Actuellement, elle est à sa neuvième version.
- La conception modulaire d'Openstack permet une évolutivité souple. Il est ainsi possible d'ajouter des ressources ou des composants pour adapter la plate-forme aux évolutions des besoins en conservant une interface uniforme pour l'utilisateur final.
- C'est la solution open source la plus utilisée. La figure 4 illustre le pourcentage d'utilisation des solutions open source pour la mise en place d'un environnement Cloud et montre que la plus grande part de marché utilise la solution OpenStack. Elle est implémentée au sein de grands groupes tels que Cisco et CERN

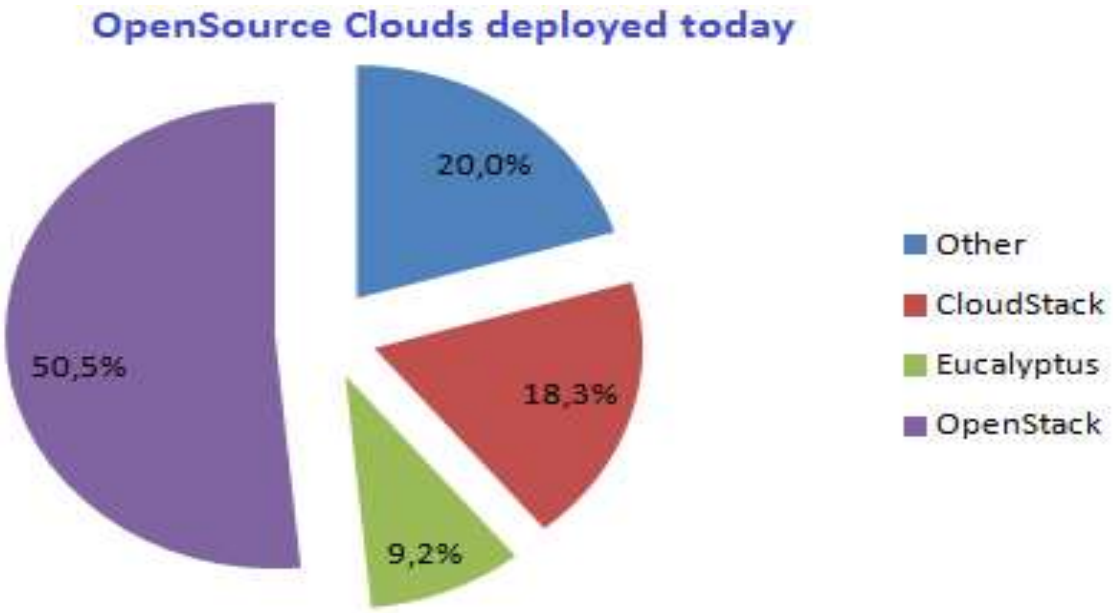

Figure 14: Utilisation des solutions open source

2.2. Présentation de l'Hyperviseur

Après avoir choisi Openstack comme IaaS, il nous reste le choix du l'hyperviseur qui se fera parmi ceux compatible à savoir Xen, KVM, vSphere, Hyper-V

2.2.1. Quelques hyperviseurs

a) Xen

Xen est un hyperviseur de type 1 (Bare Metal) qui a débuté comme un projet de recherche dans l'université du Cambridge au Royaume-Uni. La société XenSource a été créée et en a poursuivi le développement. Il est intégré partiellement dans la partie principale du noyau linux depuis la version 3.0.

b) KVM

KVM (Kernel-based Virtual Machine) est un hyperviseur libre de type 1 (Bare Metal) pour Linux développé par Red hat. KVM et est intégré dans le noyau Linux depuis la version 2.6.201.

c) vSphere/ESXI

VMware vSphere est un logiciel d'infrastructure de Cloud computing de l'éditeur VMware, c'est un hyperviseur de type 1 (Bare Metal).

d) Hyper-V

Hyper-V, également connu sous le nom de Windows Server Virtualization, est un système de virtualisation basé sur un hyperviseur 64 bits de la version de Windows Server 2008.

2.2.2. Etude comparatif des Hyperviseurs

Le tableau ci-dessous résume les principales caractéristiques de chaque hyperviseur.

Tableau 4: Comparatif des hyperviseurs

HYPERVISEUR	XEN	KVM	VSPHERE	HYPER-V
Société	XenSource	RedHat	VMware	Microsoft
Développer pour	Personnel Petite/Moyenne entreprise	Personnel Petite/Moyenne entreprise	entreprise	entreprise
Maturité	Très bonne	Très bonne	Très bonne	Bonne
Max CPU haute	160	160	480	320
Max RAM haute	1To	4To	12To	4To
Max vCPU/VM	16(win)/32(linux)	160	128	64(win)/32(linux)
Max RAM/VM	192Go	4To	4To	1TO
VM/haute	500(win)/650(linux)	Pas de limite	1000	1024

2.2.3. Choix de l'hyperviseur

Nous avons choisi d'utiliser vSphere comme hyperviseur car il est globalement plus performant que les autres hyperviseurs. En plus, SERVICOM est en cour d'établir un partenariat avec VMware ce qui a guideé notre choix vu qu'on pourra avoir une assistance technique et des conseillers si le besoin se fait ressentir.

Après avoir procéder au choix du IaaS et de l'hyperviseur à implémenter, nous allons procéder à la conception générale du projet.

3. Conception

Nous allons commencer par la conception de l'architecture réseau globale qui présente la mise en place et qui assure la communication entre les deux sites.

3.1. Conception de l'infrastructure

Ci-dessous la figure 15 décrit l'architecture globale et la connexion entre les deux sites

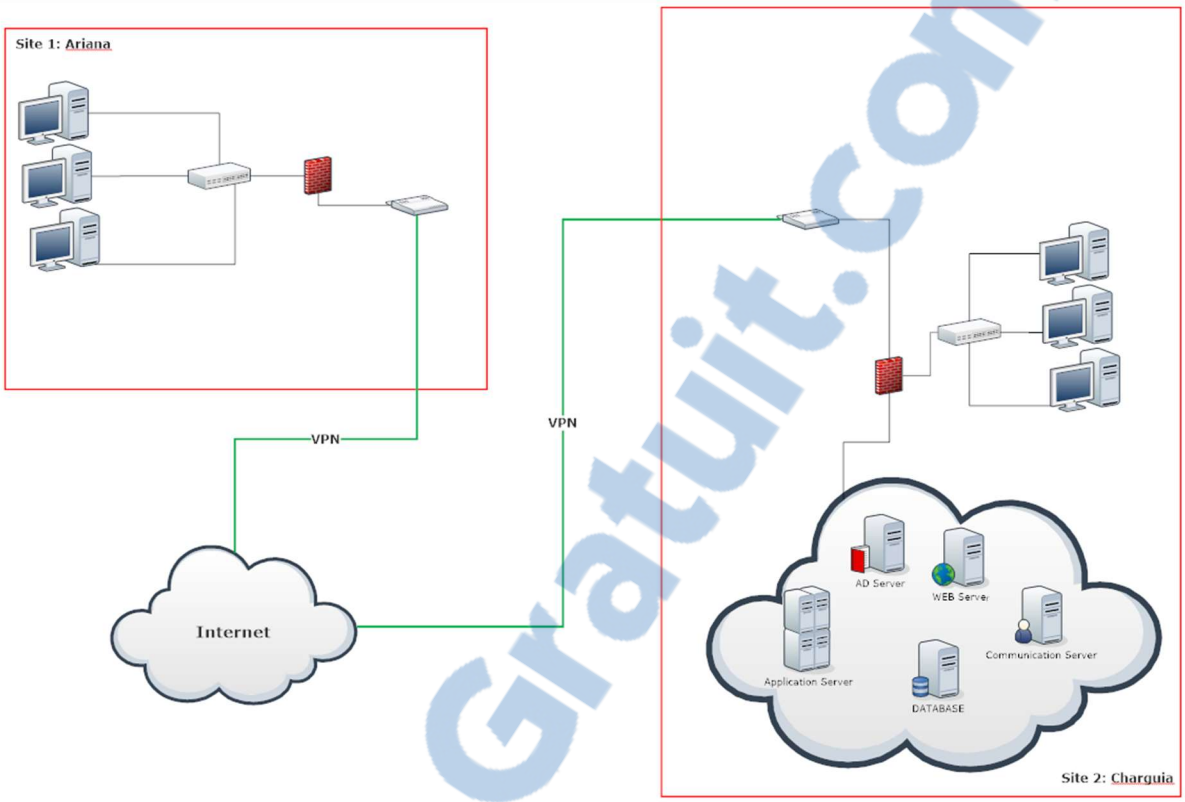

Figure 15: architecture globale

BOUCLE 2

1. Spécification du besoin

Le marché des solutions de virtualisation est un marché extrêmement concurrentiel. Parmi les principaux acteurs composant ce marché, nous détaillerons les offres des trois principaux éditeurs, à savoir VMware, Citrix et Microsoft.

VMware est le plus vieux et le plus gros éditeur sur le marché de la virtualisation, filiale du groupe EMC et fondée en 1998, cette compagnie propose une gamme de produits et services complète associée à la virtualisation.

Pour mettre en place notre plateforme de virtualisation, nous avons choisi la solution Vmware qui est le leader du marché dans le domaine.

1.1. VMware vSphere

VMware vSphere [10] gère de grandes collections d'infrastructure (par exemple des processeurs, le stockage et la gestion de réseau) sous la forme d'un environnement d'exploitation transparent et dynamique, ainsi que la complexité d'un centre de données.

La pile logicielle VMware vSphere est constituée des couches de virtualisation, de gestion et d'interfaces.

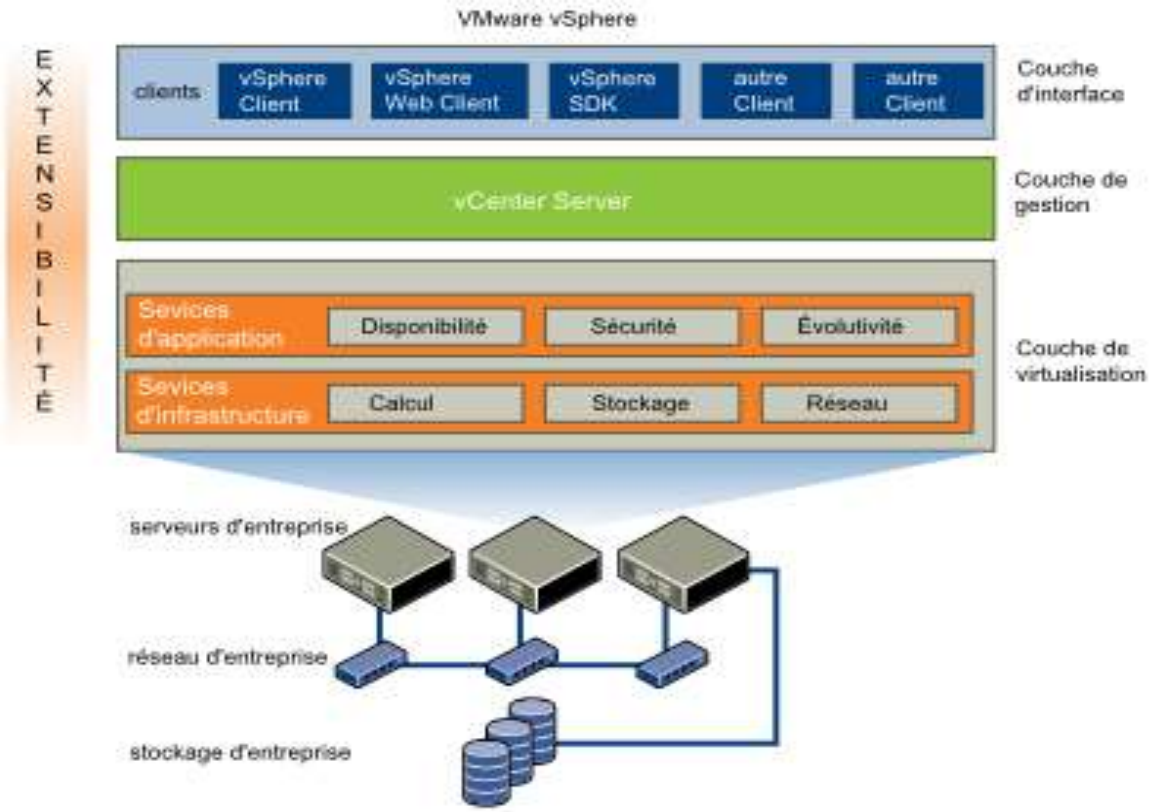

Figure 16: Relations entre les couches de composants de VMware vSphere [10]

- Couche de virtualisation : La couche de virtualisation de VMware vSphere inclut des services d'infrastructure et d'application. L'infrastructure fournit, entre autres, les services de traitement et de stockage et les services réseau extraient, agrègent et allouent les ressources matérielles ou d'infrastructure. Les services d'infrastructure comprennent les types suivants :
- Services de traitement : Inclut les fonctions VMware qui permettent de ne pas tenir compte des ressources serveur hétérogènes sous-jacentes. Les services de traitement agrègent ces ressources sur un grand nombre de serveurs discrets les affectent aux applications.
- Services de stockage : Il s'agit de l'ensemble de technologies qui permettent d'utiliser et de gérer efficacement le stockage dans les environnements virtuels.
- Services de réseau : Il s'agit de l'ensemble de technologies qui simplifient et améliorent la gestion de réseau dans les environnements virtuels.
- Couche de gestion : VMware vCenter Server est le point central de la configuration, du provisionnement et de la gestion des environnements informatiques virtualisés.
- Couche d'interfaces : Les utilisateurs peuvent accéder au centre de données VMware vSphere via des clients à interface graphique, tels que vSphere Client ou vSphere Web Client. En outre, ils peuvent accéder au centre de données via des machines clientes qui utilisent des interfaces de ligne de commande et des kits SDK pour la gestion automatique.

1.2. Composants et fonctions VMware vSphere

Une présentation des composants et fonctions de VMware vSphere explique les composants et leurs interactions.

VMware vSphere inclut les composants et fonctions suivants

- VMware ESXi [10]: Une couche de virtualisation fonctionne sur des serveurs physiques qui analysent le processeur, la mémoire, le stockage et les ressources dans les machines virtuelles multiples.

Figure 17: VMware ESXI

- VMware vSphere Client [10] : Une interface permettant aux utilisateurs de se connecter à distance vCenter Server ou ESXi depuis n'importe quel PC Windows.

Figure 18: VMware vSphere Client

- VMware vSphere Web Client [10] : Une interface Web permet aux utilisateurs de se connecter à distance à vCenter Server depuis divers navigateurs Web et systèmes d'exploitation.

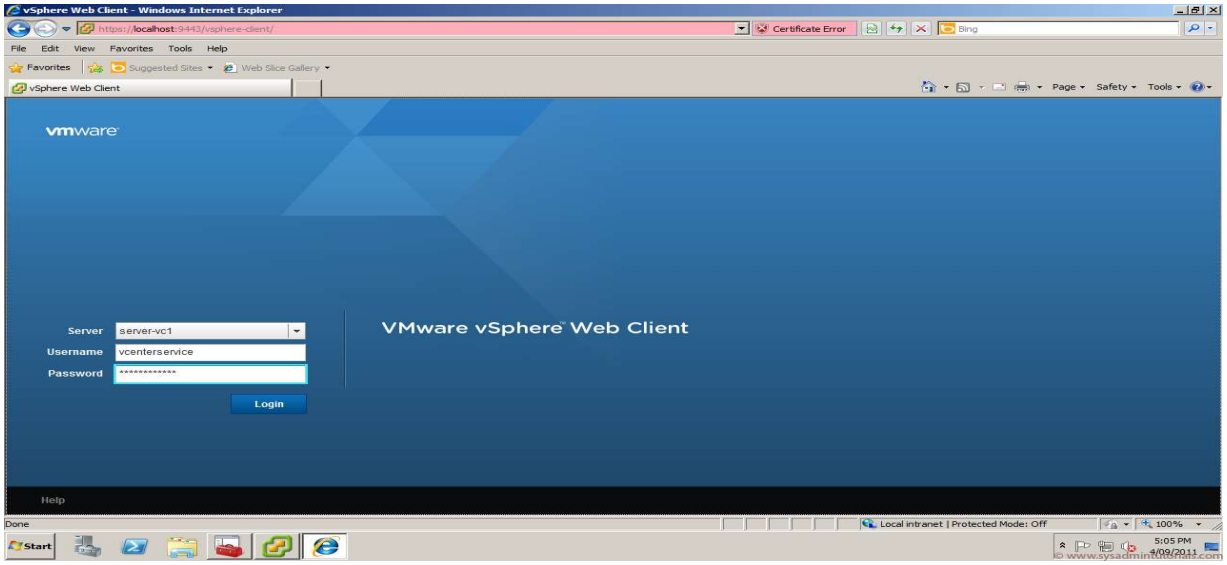

Figure 19: VMware vSphere Web Client

- VMware vCenter Server [10]: Le point central pour configurer, approvisionner et gérer des environnements informatiques virtualisés. Il fournit les services essentiels du centre de données : contrôle d'accès, surveillance des performances et gestion des alarmes.

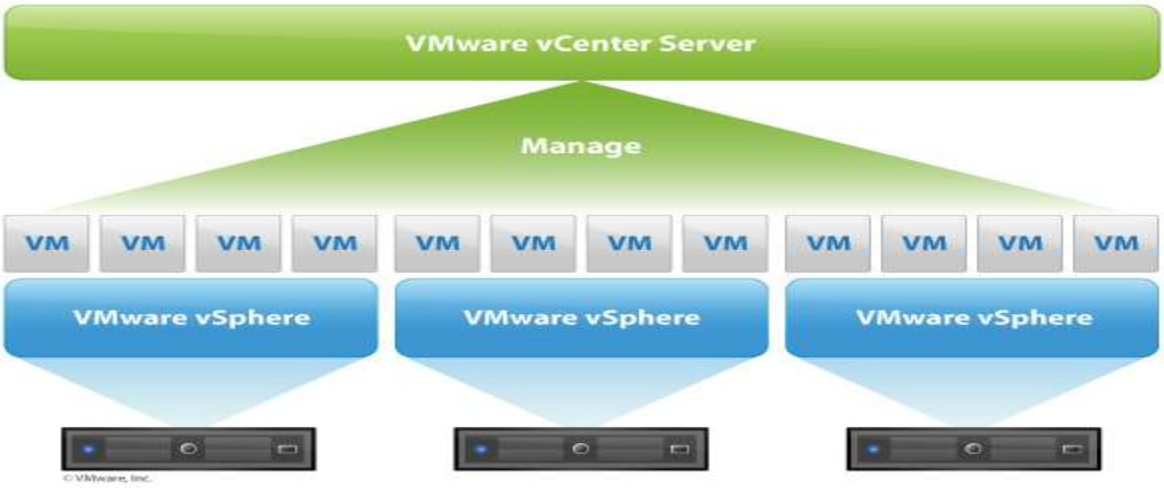

Figure 20: VMware vCenter Server

- vSphere vMotion [10] : Permet de migrer des machines virtuelles sous tension depuis un serveur physique vers un autre sans interruption avec une disponibilité de service continue et une intégrité de transaction complète.

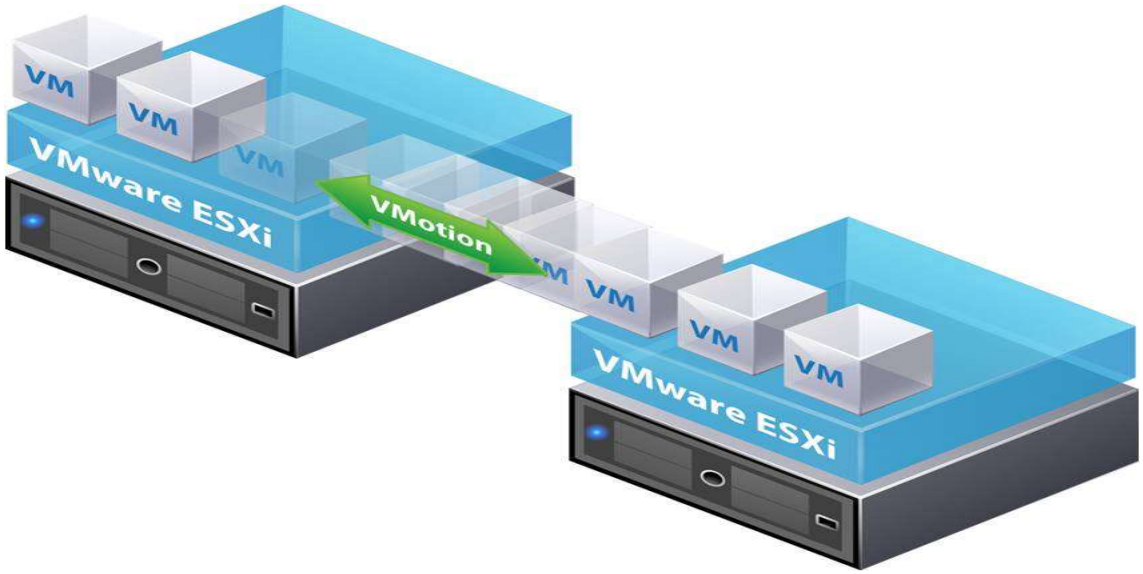

Figure 21: vSphere vMotion

- vSphere Storage vMotion [10] : Permet de migrer des fichiers de machine virtuelle d'une banque de données vers une autre sans interruption de service. Vous placez la machine virtuelle et tous ses disques dans un seul emplacement ou sélectionnez des emplacements distincts pour le fichier de configuration de la machine virtuelle et chaque disque virtuel. La machine virtuelle reste sur le même hôte pendant Storage vMotion. La migration avec Storage vMotion permet de transférer les disques virtuels ou le fichier de configuration d'une machine virtuelle vers une nouvelle banque de données alors que la machine virtuelle s'exécute. La migration avec Storage vMotion permet de transférer le stockage d'une machine virtuelle sans interrompre la disponibilité de la machine virtuelle.

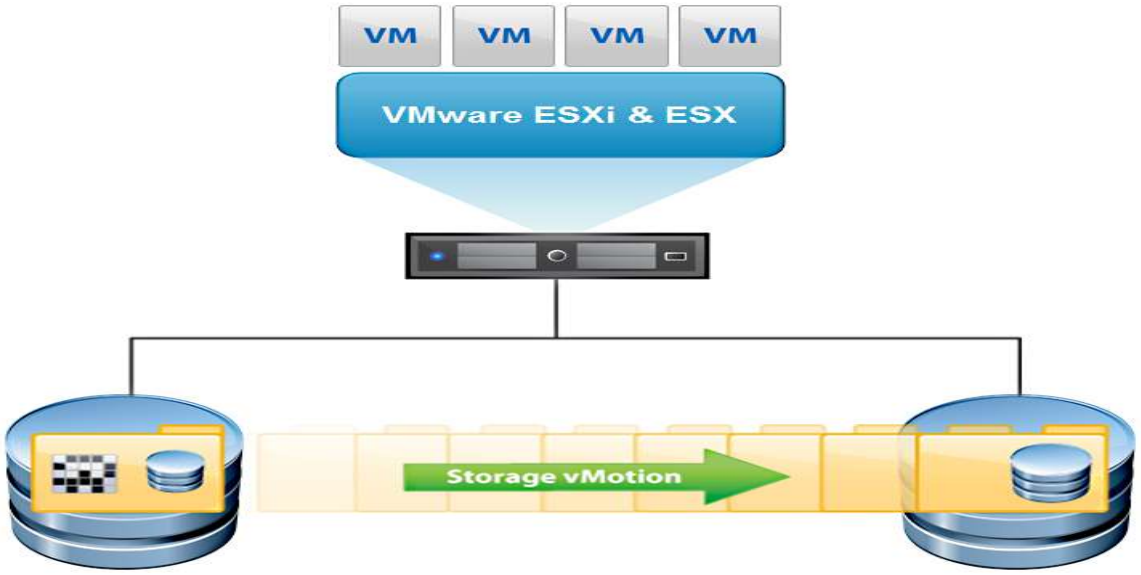

Figure 22: vSphere Storage vMotion

- vSphere High Availability (HA) [10] : Fonction qui offre une haute disponibilité pour les machines virtuelles. En cas de panne d'un serveur, les machines virtuelles affectées sont redémarrées sur d'autres serveurs disponibles ayant une capacité disponible.

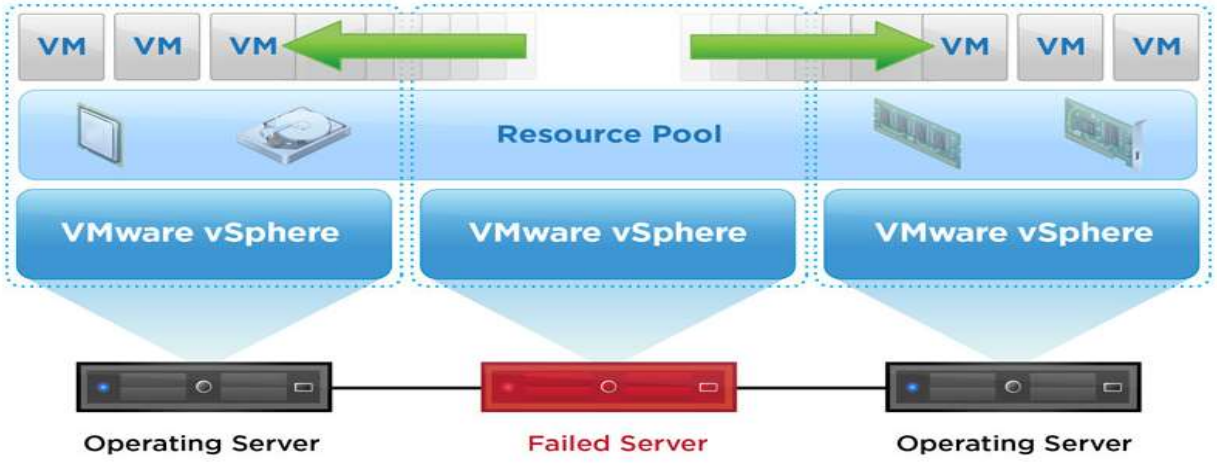

Figure 23: vSphere High Availability (HA)

- vSphere Fault Tolerance [10]: Assure la disponibilité permanente en protégeant une machine virtuelle avec une copie. Lorsque cette fonction est activée pour une machine virtuelle, une seconde copie de la machine d'origine (ou principale) est créée. Toutes les actions effectuées sur la machine virtuelle primaire sont également effectuées sur la seconde machine virtuelle. Si la machine virtuelle principale devient indisponible, la seconde machine devient active immédiatement.

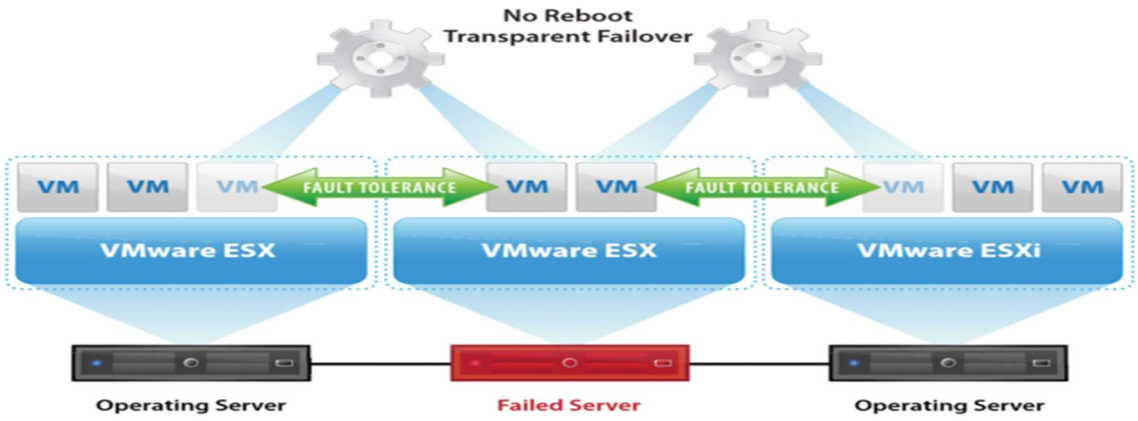

Figure 24: vSphere Fault Tolerance

2. Conception :

A ce niveau, nous traitons l'identification des cas d'utilisation et des acteurs en relation avec le fonctionnement du vSphere en tant qu'un outil qui nous permet de gérer notre infrastructure. Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Le digramme de cas d'utilisation se limite aux préoccupations réelles des utilisateurs représentés par des comptes administrateur ou simple utilisateur. Nous entamons avec un diagramme de cas d'utilisation général, où nous avons choisi pour des raisons de clarté d'omettre la représentation du cas « s'authentifier ». En fait celui-là permet d'authentifier chaque utilisateur pour qu'il puisse y accéder et exploiter ce que notre infrastructure lui offre.

2.1. Conception de l'infrastructure

La figure ci-dessous représente notre plateforme de virtualisation :

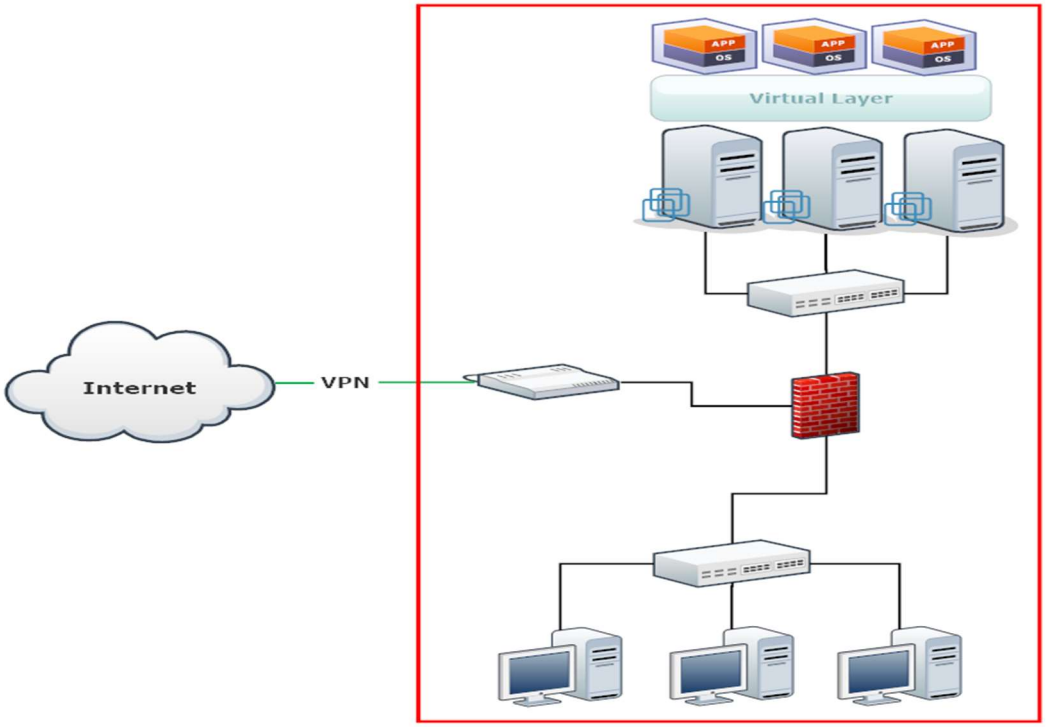

Figure 25: Architecture de la plateforme de virtualisation

2.2. Les acteurs

Dans cette section, nous présentons l'acteur qui interagit avec la plateforme de virtualisation VMware

Tableau 5: Tableau des acteurs « Vsphere »

Acteur	Type	Rôle	Système
vSphere_Admin	Principale	Gérer vSphere	vSphere
Client	Principale	Exploiter VM	vSphere

2.3. Diagramme use case

Cette section contient les interactions des acteurs avec le système

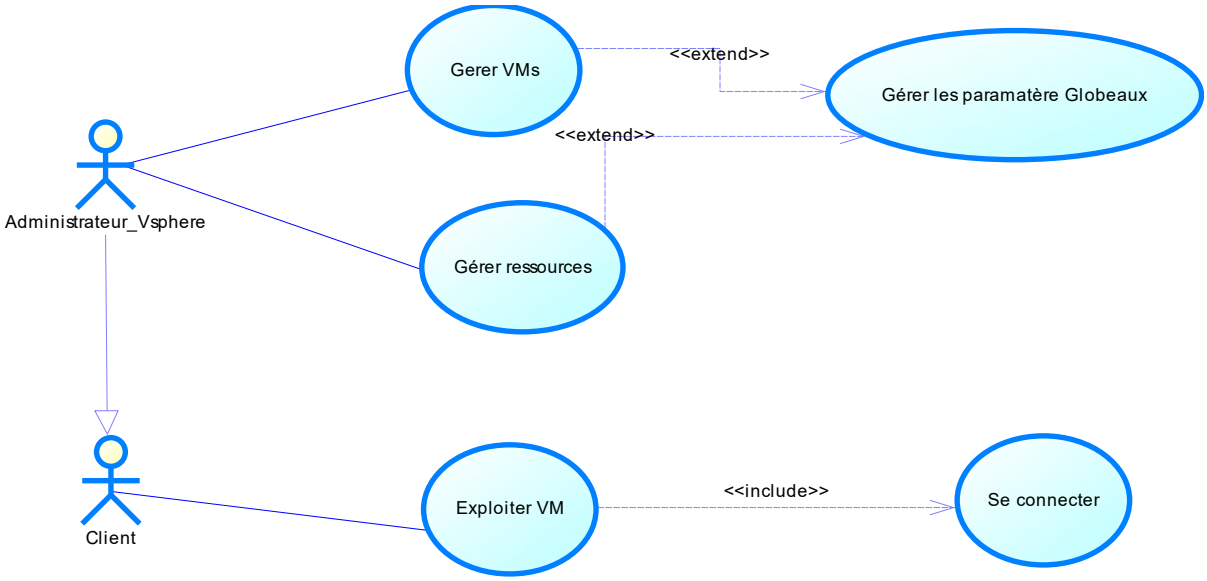

Figure 26: Use Case Générale Vsphere_Admin

Dans ce qui suit, nous nous focalisons sur chaque cas d'utilisation qui nous permet d'avoir une meilleure lecture et compréhension du fonctionnement de notre infrastructure virtuelle. Dans ce sens, une description textuelle accompagne chaque cas.

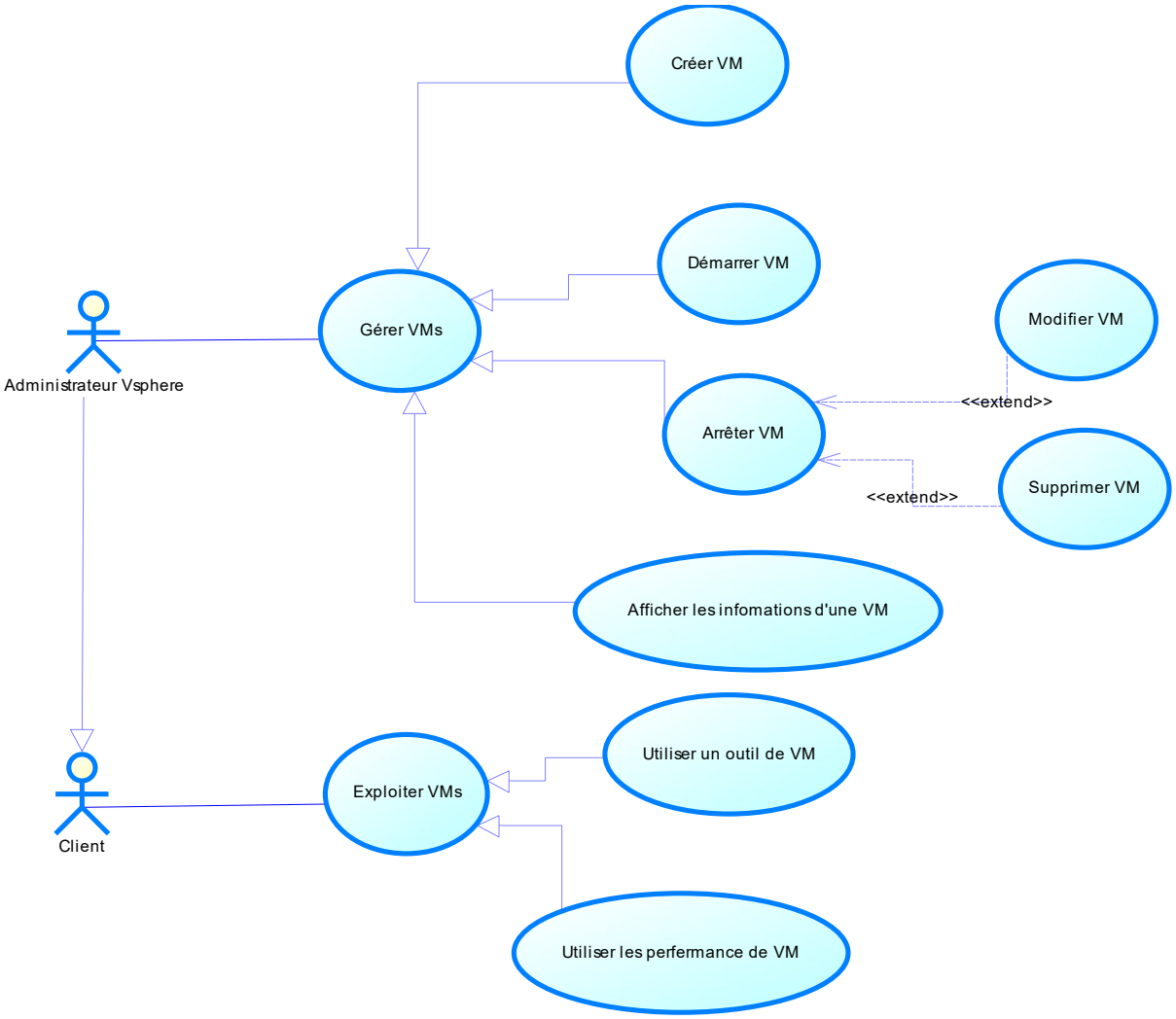

Figure 27: Use Case Gérer les VMs

Description textuelle

Sommaire d'identification	
Titre	Gérer VMs
But	Gestion des machines virtuelles
Résumé	L'administrateur vSphere gère les machines virtuelles
Acteur	Administrateur vSphere
Description des enchaînements	
Précondition	Administrateur connecté
Post condition	Les ressources ont été gérées
Description du scénario	L'administrateur accède à l'interface d'administration « vSphere client ». Créer des nouvelles machines virtuelles Démarrer des machines virtuelles Modifier ou supprimer une machine virtuelle existante

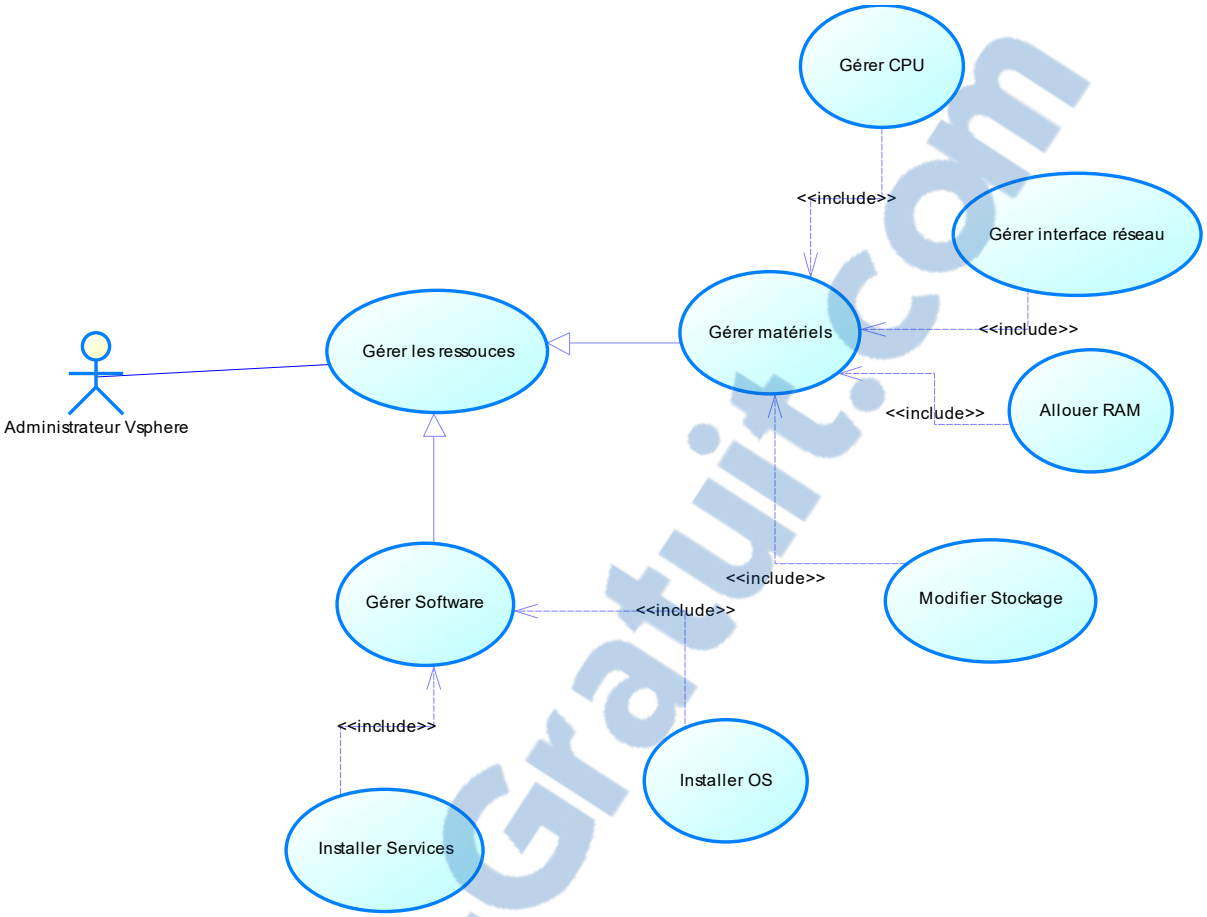

Figure 28: Use Case Gérer les ressources

Description textuelle

Sommaire d'identification	
Titre	Gérer les ressources
But	Gestion des ressources des machines virtuelles
Résumé	L'administrateur vSphere gère les ressources des machines virtuelles
Acteur	Administrateur vSphere
Description des enchaînements	
Pré condition	Administrateur connecté
Post condition	Les ressources ont été gérées
Description du scénario	L'administrateur accède à l'interface d'administration « vSphere client ». Modifier (Ajouter ou supprimer) les ressources CPU pour une machine virtuelle Gérer (Ajouter ou supprimer) une interface réseau d'une

	machine virtuelle Allouer (Ajouter ou supprimer) la capacité du RAM d'une machine virtuelle modifier (Ajouter ou supprimer) l'espace du stockage pour une machine virtuelle Installer et réinstaller des systèmes d'exploitation sur une machine virtuelle Installer et réinstaller des services sur une machine virtuelle

2.4. Diagramme de séquences

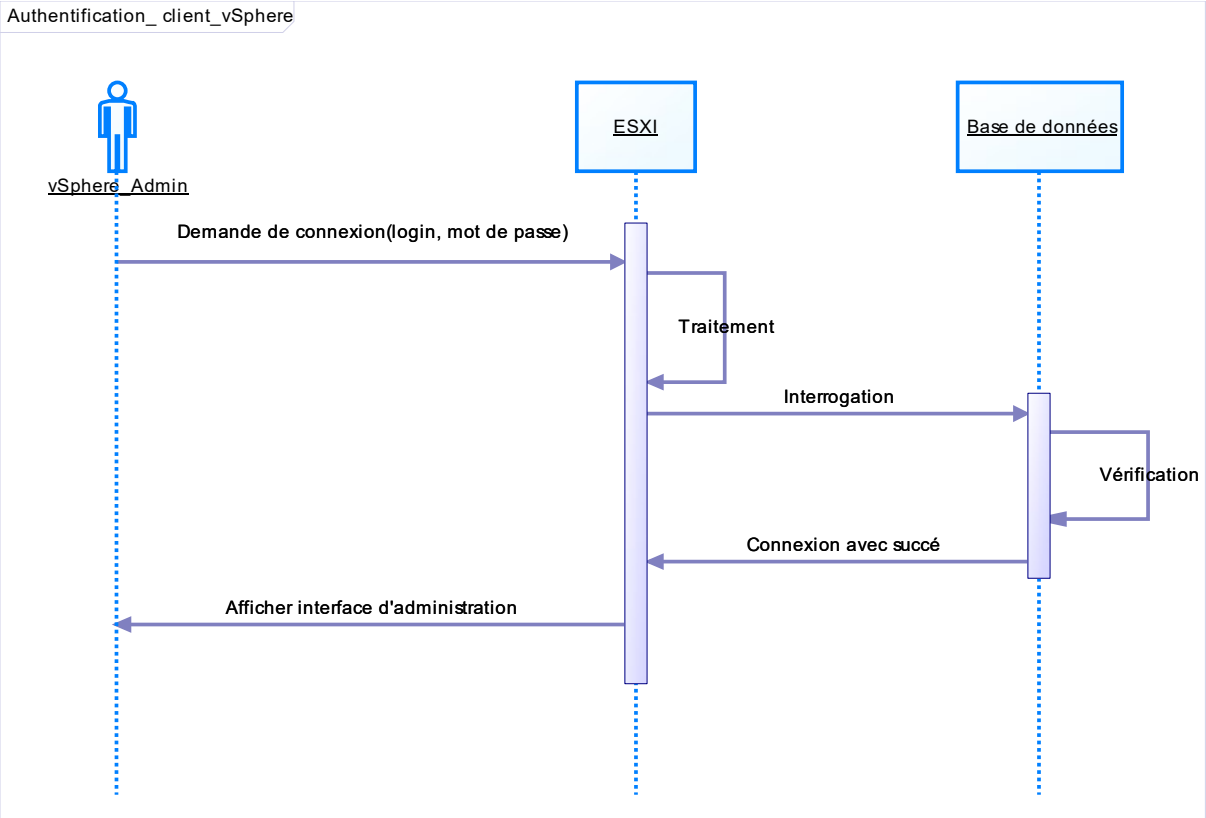

Figure 29: Digramme de séquences « Authentification vSphere Client »

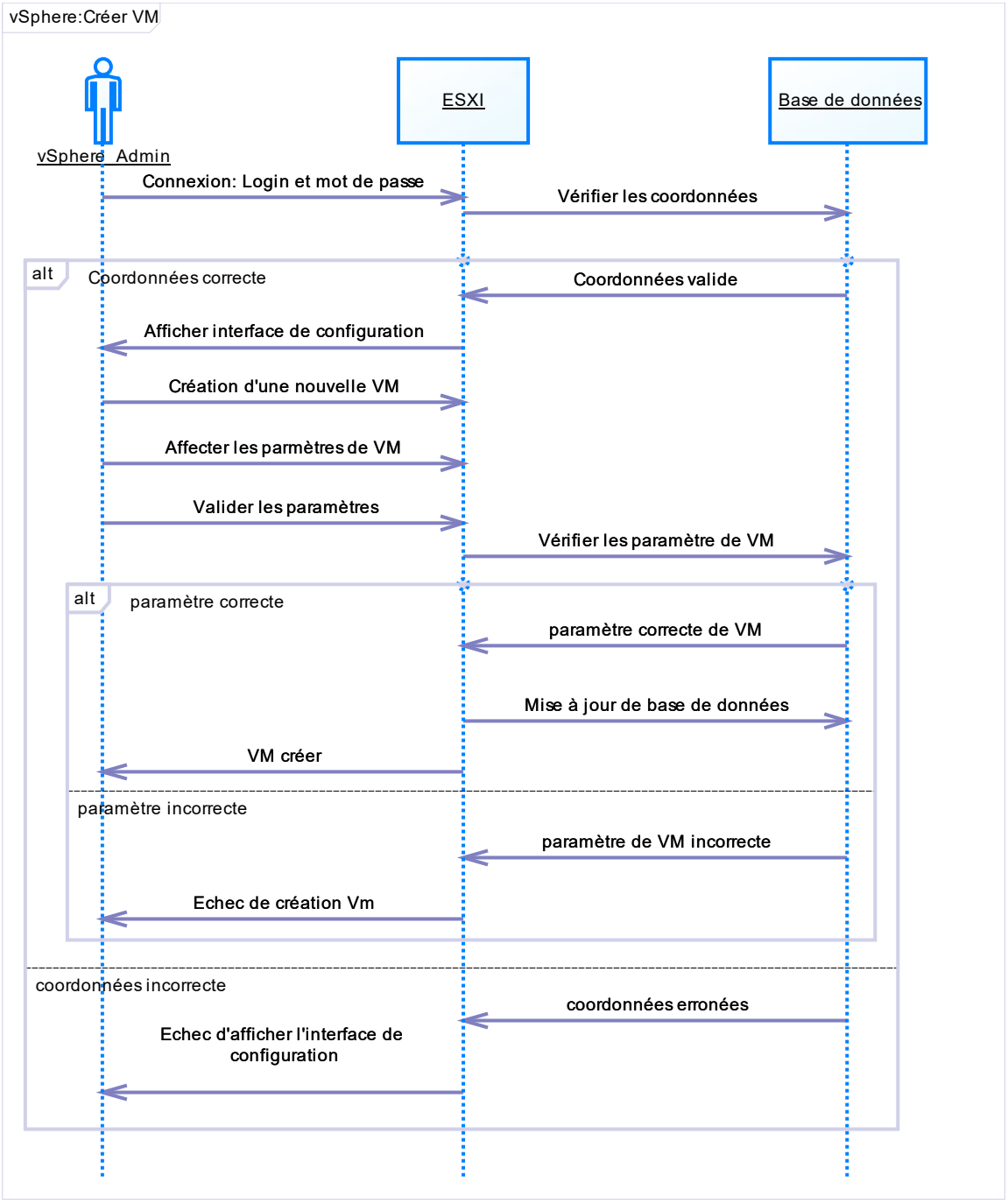

Figure 30: Digramme de séquences « Créer VMs »

3. Implémentation

Après avoir achevé l'étape de conception de notre plate-forme, nous entamons la partie implémentation. Nous allons commencer, tout d'abord, par l'installation et la configuration de notre hyperviseur, puis nous allons créer des machines virtuelles pour migrer les serveurs physiques.

3.1. Environnement matériel

Les différentes étapes d'installation et de configuration de notre hyperviseur sont réalisées sur un serveur Dell PoweEdge R210 II présentant les caractéristiques suivantes :

- Intel Xeon E3 1220 3.1Ghz

- RAM 16 GB Disque dur 1 To

3.2. Environnement logiciel

- Vmware Vsphere 6.0
- Windows Server 2008R2 pour les machines virtuelles

La figure 31 ci-dessous montre l'interface d'authentification permettant d'accéder au paramètre de « ESXI »

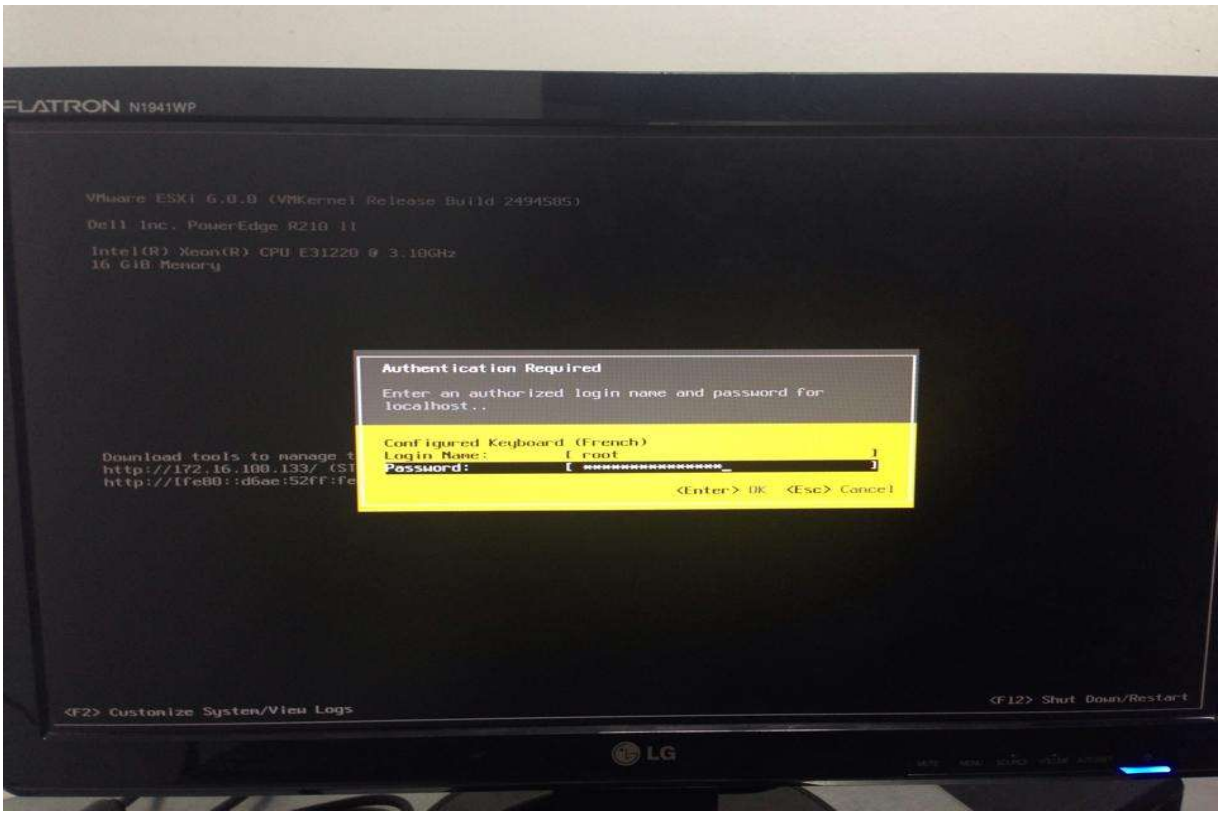

Figure 31: interface d'authentification au hyperviseur « ESXI »

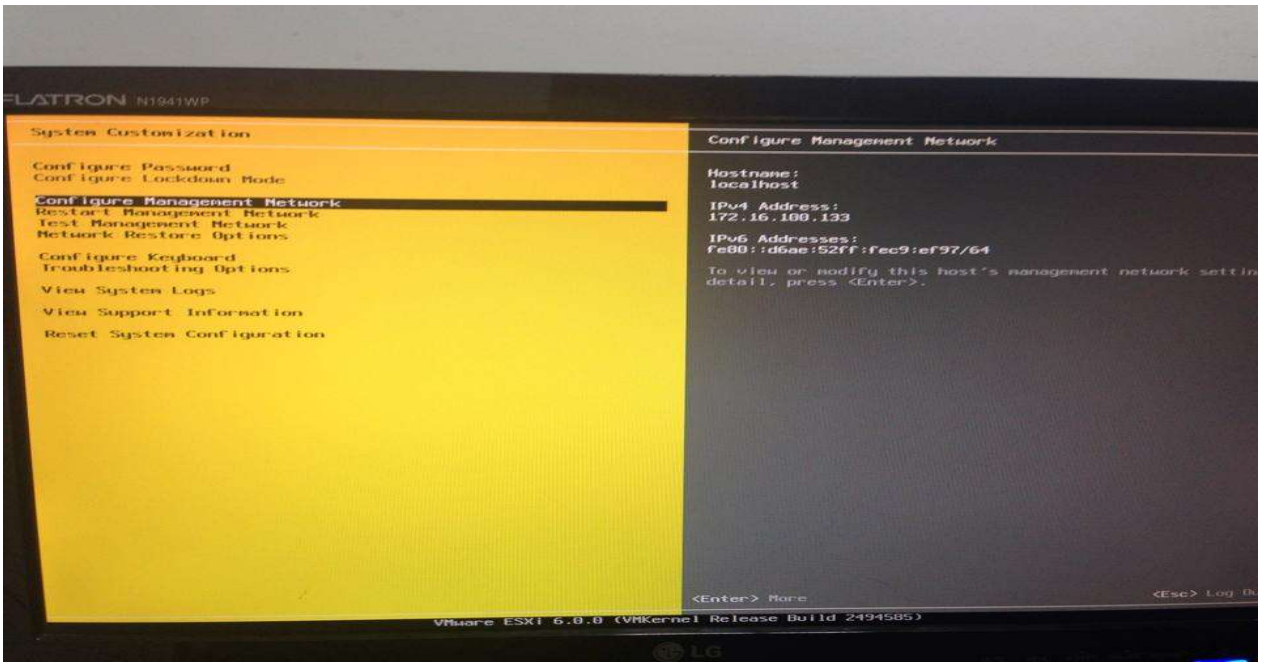

Figure 32: Interface de configuration « ESXI »

La figure 32 présente l'interface d'administration et configurations « ESXI »

Figure 33: Interface de connexion « vSphere Client »

Cette figure 33 présente l'interface de connexion « vSphere Client »

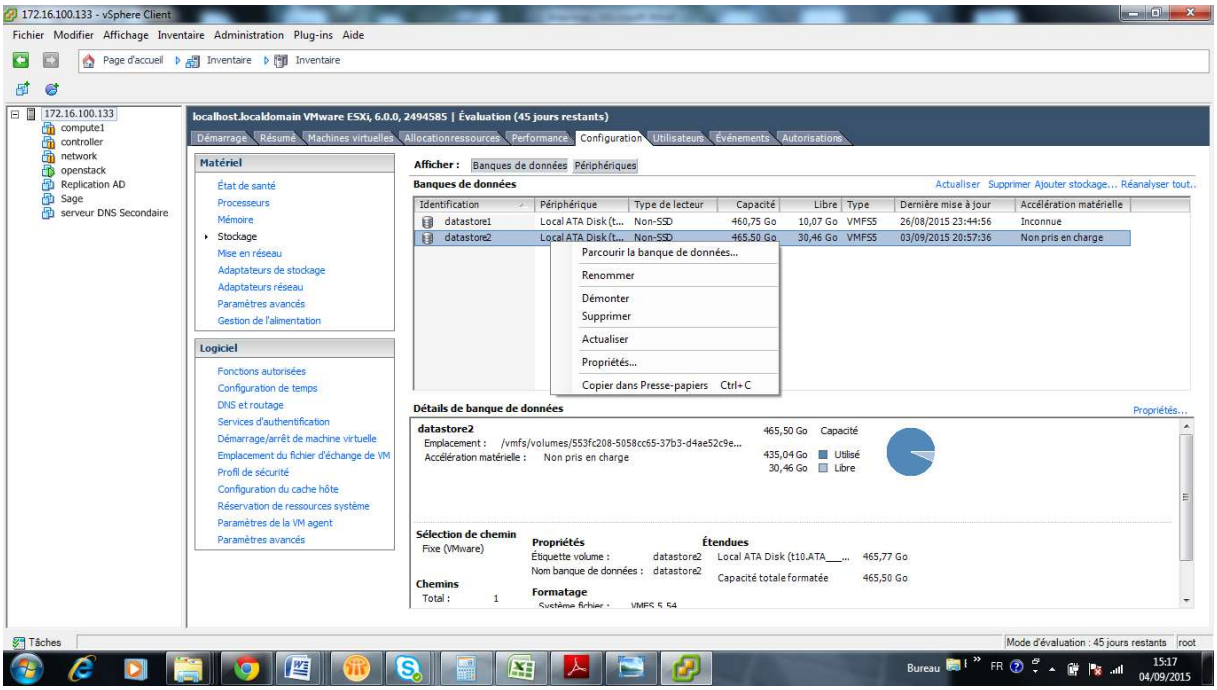

Figure 34: Configuration « DataStore »

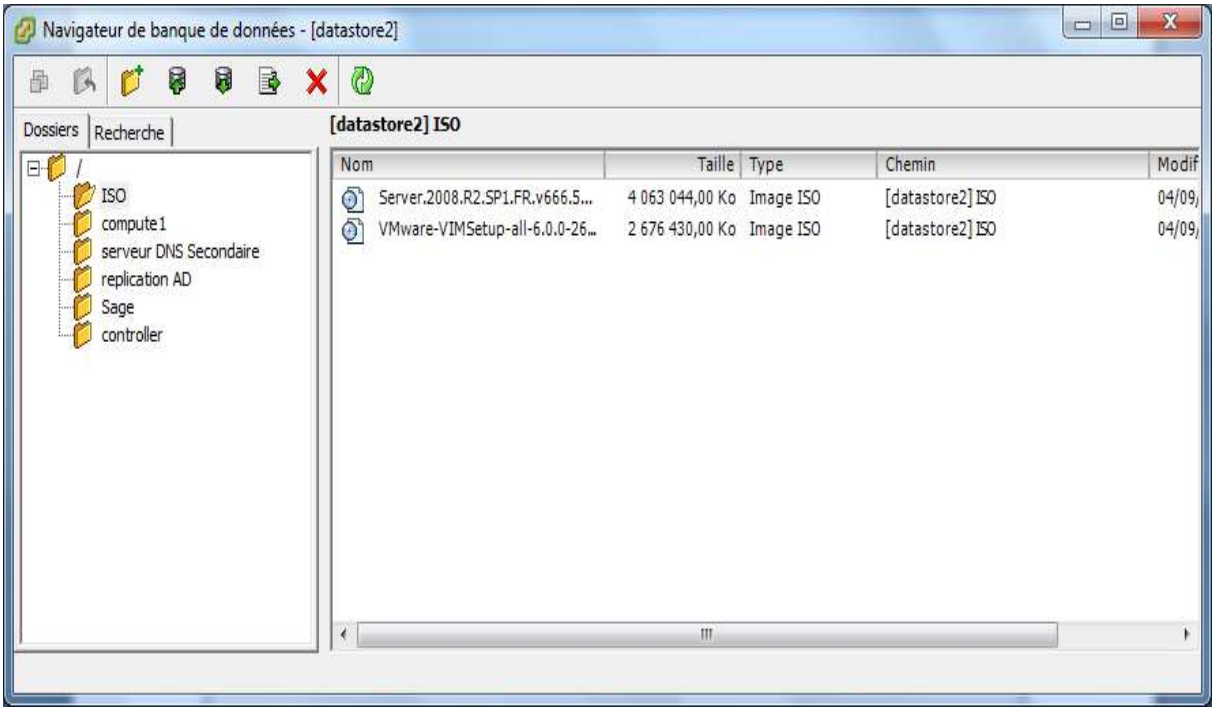

Figure 35: Ajout image ISO dans « DataStore »

Les deux figures 34 et 35 illustres la manipulation du « DataStore » (l'espace de stockage)

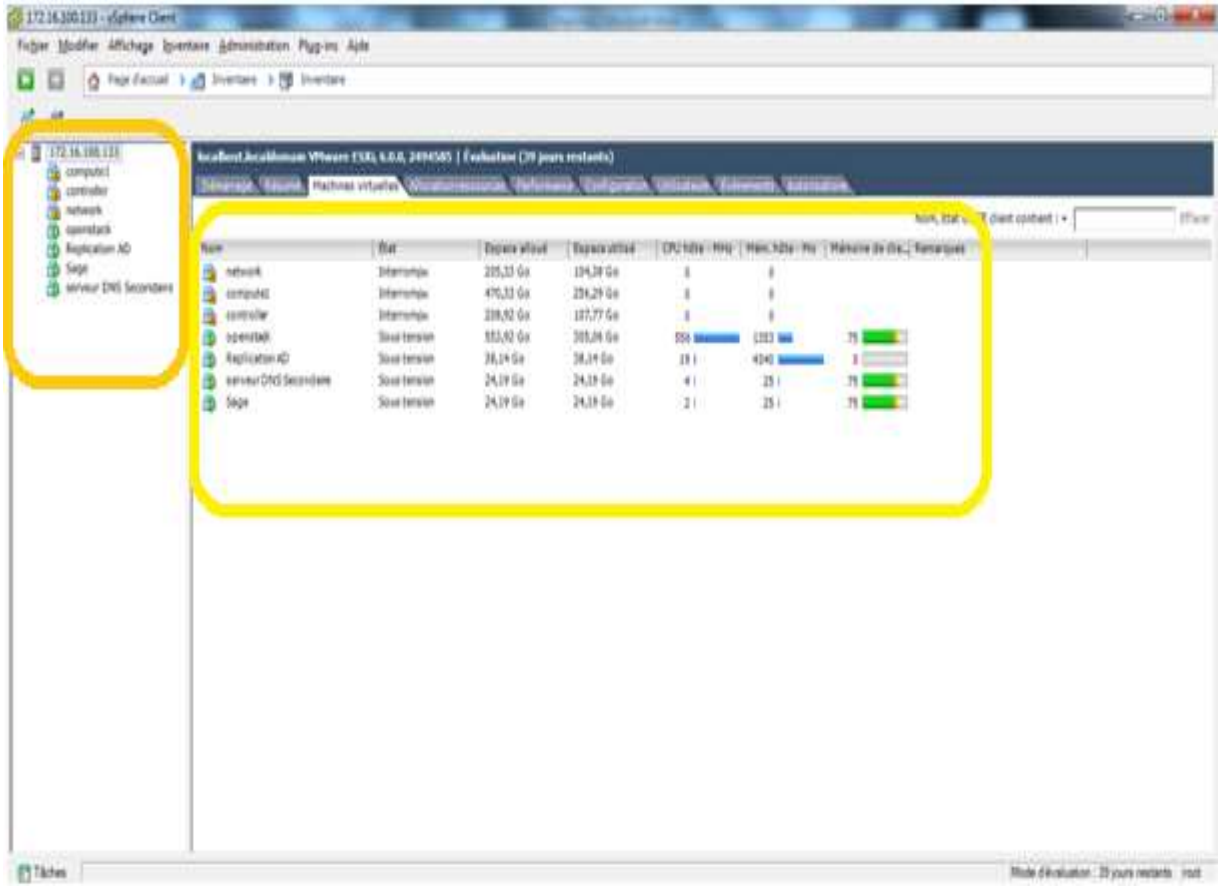

Figure 36: Etat des machines virtuelles

La figure 36 représente l'interface de monitoring des machines virtuelles

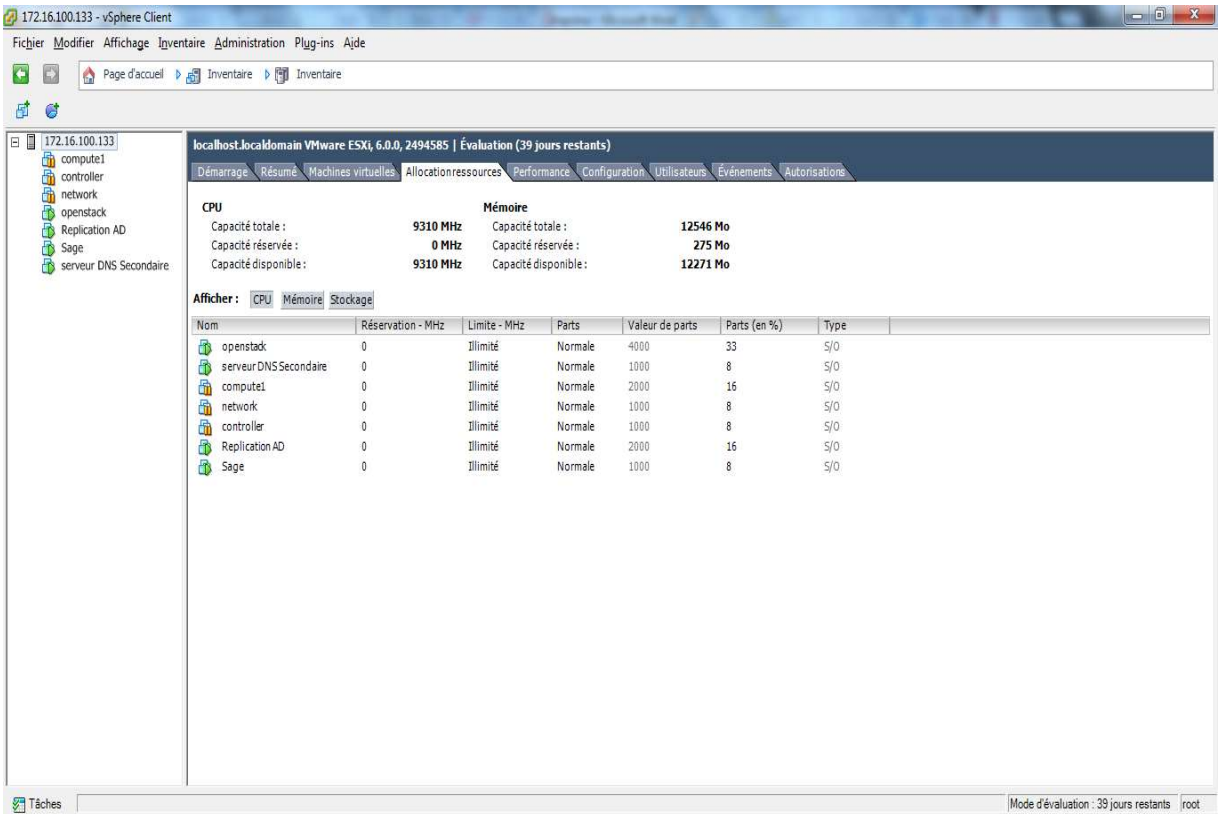

Figure 37: Vérification des ressources

La figure 37 illustre l'interface de gestion « vSphere Client » où l'administrateur peut Controller les ressources allouées par les différentes machines virtuelles

4. Test

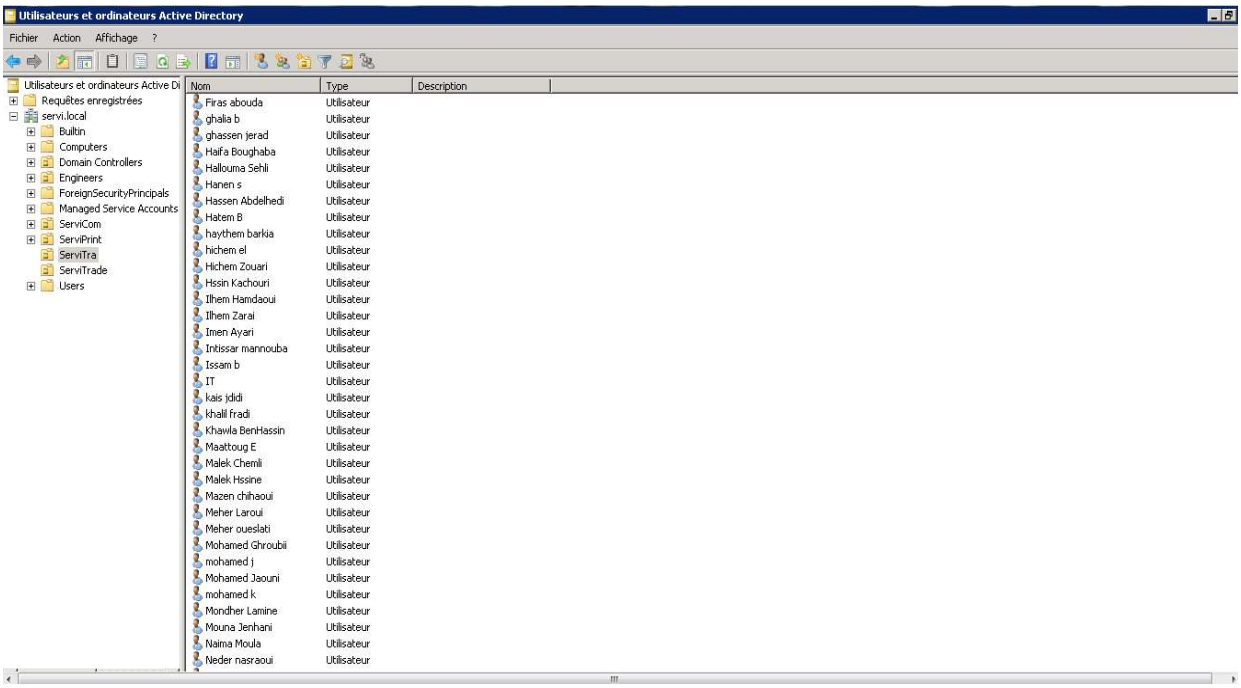

Figure 38: Interface machine virtuelle réplication Active Directory

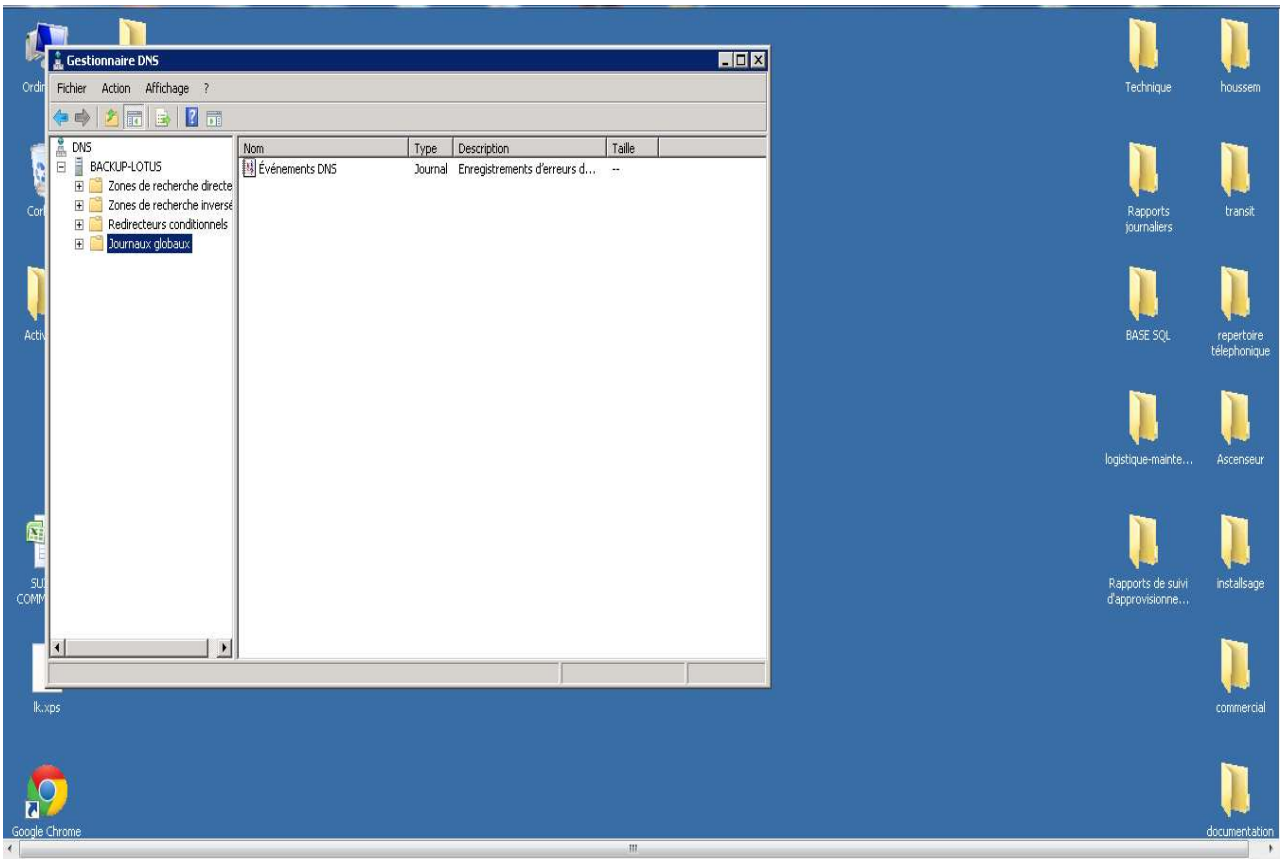

Figure 39: Interface serveur DSN Secondaire

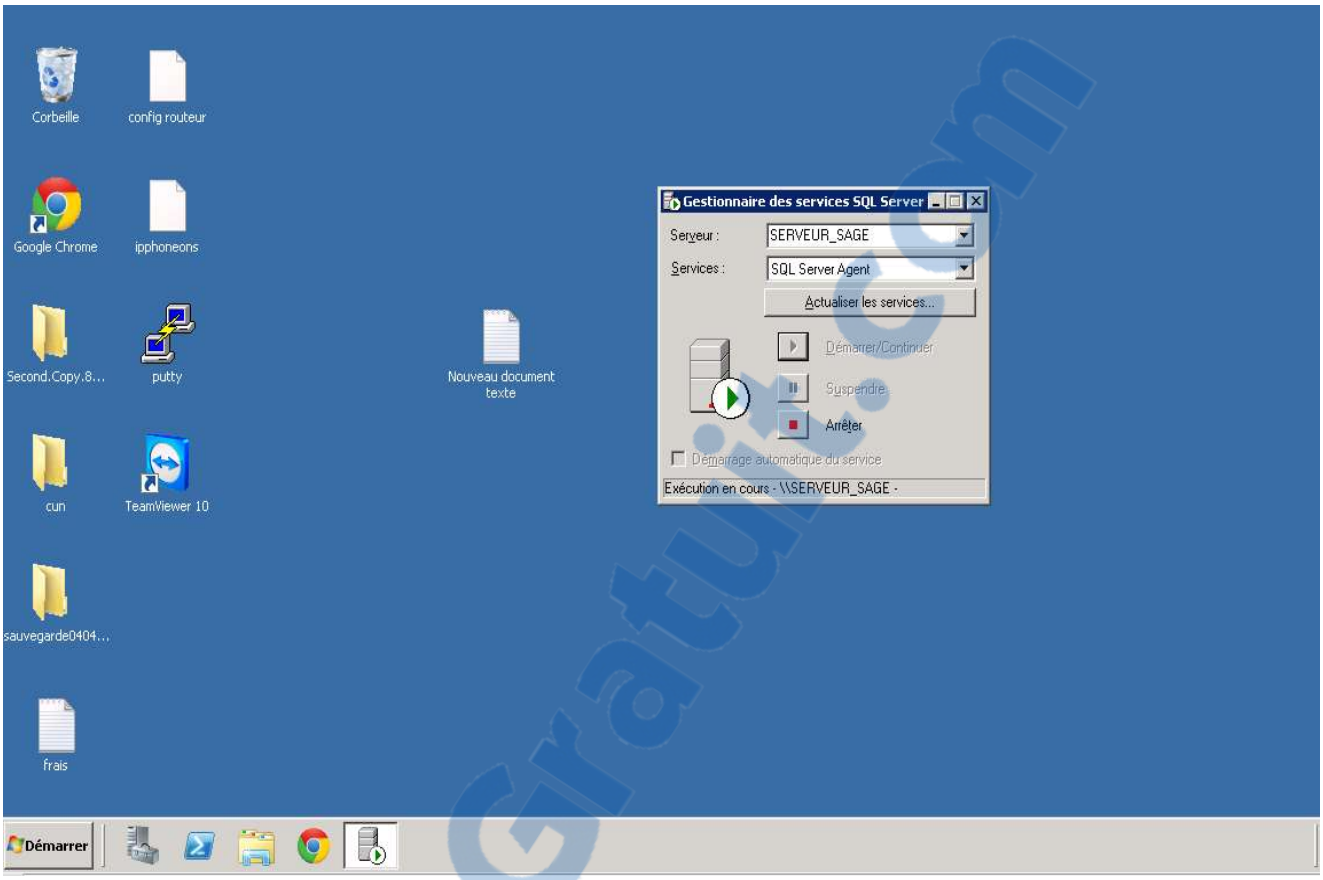

Figure 40: Interface serveur Sage

BOUCLE 3

5. Les caractéristiques de la solution OpenStack

La solution OpenStack est un projet dont l'objectif est de permettre à toute organisation de créer et d'offrir une solution Infrastructure en tant que service (IaaS) en utilisant du matériel standard à travers une variété de composants qui offrent des services complémentaires. Chaque service offre une interface de programmation d'application (API) qui facilite cette intégration.

5.1. Les versions

Le projet OpenStack a débuté en juillet 2010, il est aujourd'hui (depuis la version Essex) utilisable dans un environnement de production.

Il faut dire que le développement d'OpenStack est très soutenu avec une cadence régulière d'une nouvelle version tous les six mois.

Le tableau ci-dessous nous donne une récapitulation des neuf versions OpenStack avec les différents composants inclus dans chacune.

Tableau 6: Les différentes versions d'OpenStack

Nom	Date	Composants inclus
Austin	Octobre 2010	Nova, Swift
Bexar	Février 2011	Nova, Glance, Swift
Cactus	Avril 2011	Nova, Glance, Swift
Diablo	Septembre 2011	Nova, Glance, Swift
Essex	Avril 2012	Nova, Glance, Swift, Horizon, Keystone
Folsom	Septembre 2012	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Grizzly	Avril 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Havana	Octobre 2013	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer
Icehouse	Avril 2014	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove

5.2. Les différents composants d'OpenStack

Le projet OpenStack possède une architecture modulaire organisée sous forme de composants, chacun d'eux avec une fonction particulière offrant un service particulier.

Le lancement d'une instance (machine virtuelle) implique de nombreuses interactions entre plusieurs services.

Les composants d'OpenStack peuvent être divisés en trois grands projets selon leur degré d'importance dans l'implémentation d'un projet Cloud computing :

- Les composants communs
- Les composants de base
- Les composants optionnels

Le tableau suivant présente les différents composants de la solution OpenStack.

Tableau 7: Les différents composants OpenStack

Services	Nom du code	Description
Les composants communs		
OpenStack Identity	***Keystone***	Gestion de l'identité
OpenStack Dashboard	***Horizon***	Interface web de paramétrage et gestion
Les composants de base		
OpenStack Compute	***Nova***	Gestion des instances (serveurs)
OpenStack Network	***Neutron***	Gestion des réseaux
OpenStack Imaging Service	***Glance***	Gestion des images
Les composants optionnels		
OpenStack Storage	***Swift***	Stockage d'objet
OpenStack Block Storage	***Cinder***	Service de disques persistants pour les machines virtuelles
OpenStack Telemetry	***Ceilometer***	Service de métrologie notamment pour la facturation
OpenStack Orchestration	***Heat***	Service d'orchestration à base de template

Les composants sont tous indépendants les uns des autres sauf le module Horizon (tableau de bord) qui lui nécessite l'installation des modules Nova et Keystone. Il n'est donc pas nécessaire d'installer tous les modules pour qu'OpenStack fonctionne.

La figure suivante présente l'architecture conceptuelle d'un environnement OpenStack typique.

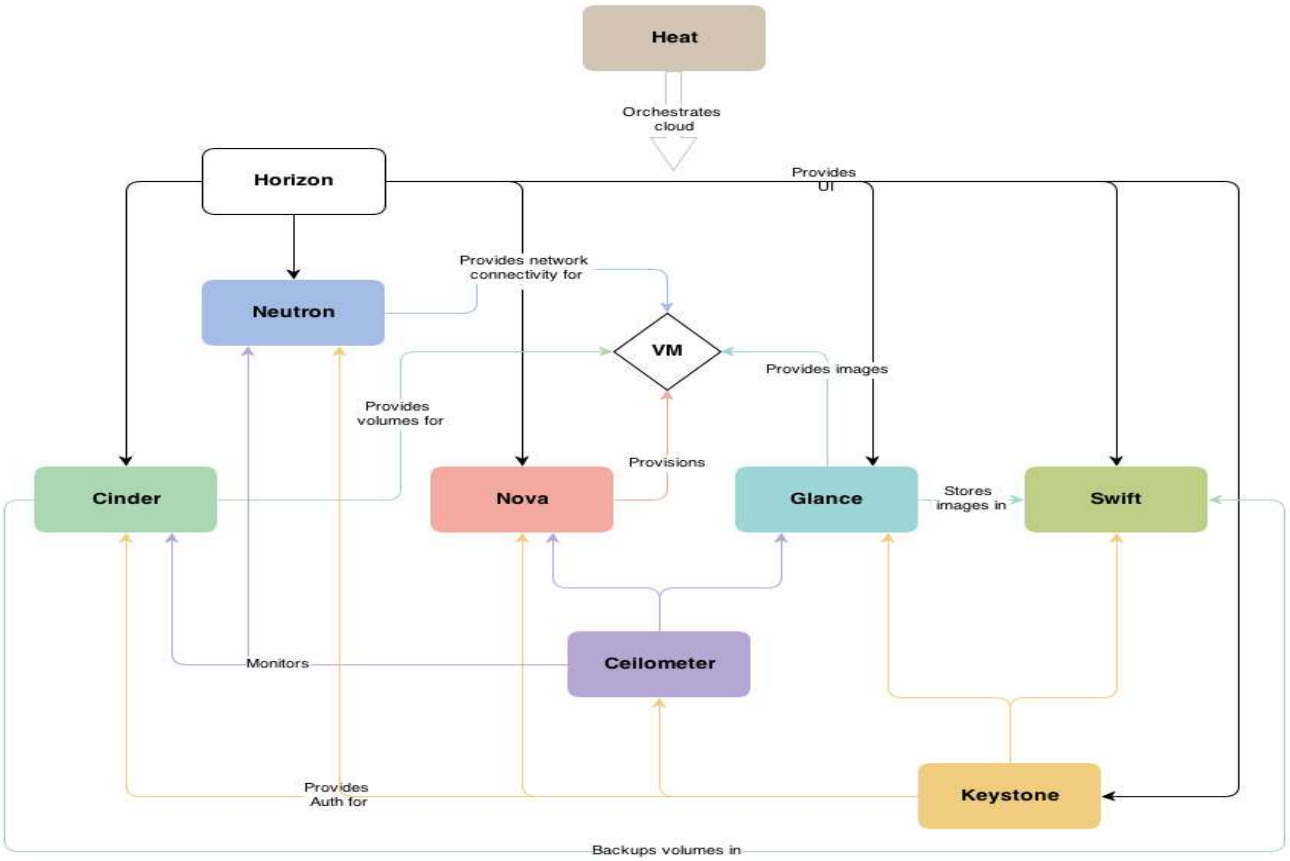

Figure 41: Architecture conceptuelle d'OpenStack

5.3. Les composants communs

5.3.1. OpenStack Identity (Keystone)

Le composant Keystone est chargé de la gestion des utilisateurs et des catalogues de services. C'est un annuaire qui centralise toutes les authentifications et autorisations nécessaires aux multiples composants d'OpenStack.

L'authentification vérifie que la demande provient en réalité de celui qu'il prétend être. L'autorisation consiste à vérifier si l'utilisateur authentifié a accès à des services auxquels il souhaite accéder.

La figure qui suit résume le fonctionnement de Keystone.

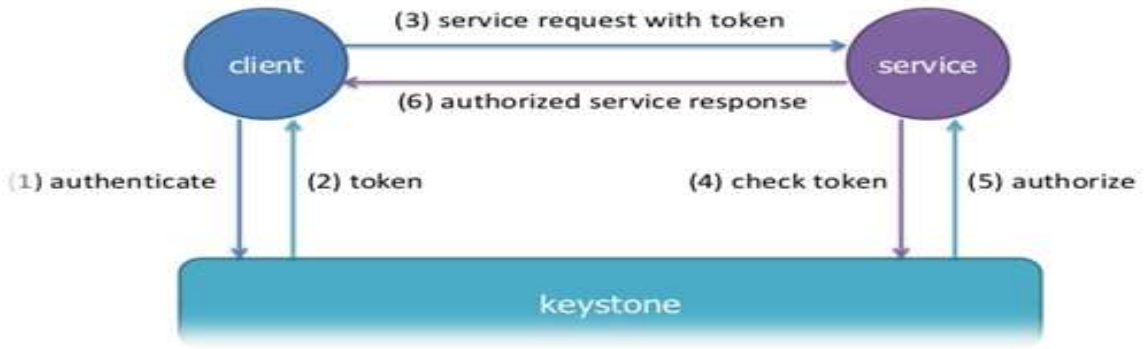

Figure 42: Fonctionnement de Keystone

- Gestion des utilisateurs

La gestion des utilisateurs s'articule autour de trois objets:

- ✓ L'objet User représentant l'utilisateur final.
- ✓ L'objet Tenant c'est un contenaire qui permet de grouper des ressources que l'on peut représenter par un projet, une organisation au sein de laquelle les instances seront regroupées et administrées.
- ✓ L'objet Role présente un ensemble de droits et de privilèges qui définit le rôle de l'utilisateur sur un tenant. Un utilisateur peut avoir un ou plusieurs rôles sur différents tenants.

5.3.2. Gestion des services et points d'accès

- ✓ Service : Openstack service comme nova, glance… fournit un ou plusieurs ''endpoints'' auquel les users peuvent y accéder et effectuer des opérations
- ✓ Endpoint : c'est un URL / adresse pour accéder à un service OpenStack.

Keystone s'appuie sur un SGBDR soit de type SQLite, MySQL ou PostgreSQL.

Keystone utilise par défaut le port 35357.

5.3.3. OpenStack Dashboard (Horizon)

Horizon est un tableau de bord permettant l'administration du Cloud via une interface web qui permet de gérer facilement la création d'instances, de réseaux virtuels, d'images, etc... Cette interface web est développée en Python, basé sur le framework Django, il utilise le module WSGI *(en)* d'Apache.

5.4. Les composants basiques

5.4.1. OpenStack Compute (Nova)

Nova est la partie principale d'un système IaaS, c'est le composant qui assure les opérations de calcul et de traitement au sein du Cloud, à savoir celui qui permet de mettre à disposition tout ce qu'il faut pour la création, le contrôle et la suppression des instances de machines virtuelles à partir de la ligne de commande ou à partir d'un tableau de bord du composant Horizon.

Pour faire un rapprochement, ce module offre les fonctionnalités proposées par EC2 d 'AWS. Il permet de configurer les groupes de sécurité pour effectuer des contrôles d'accès, définir les keypairs pour accéder aux instances et choisir les tailles d'instances à partir des ''flavors'' (m1.small, m1.large, ...) que l'administrateur peut définir en leur associant un certain nombre de VCPUs, de l'espace de stockage, une quantité de RAM.

Openstack compute prend en charge plusieurs hyperviseurs y compris KVM, QEMU, LXC, XenServer, Hyper-V et VMWare. Il repose sur le langage Python, les frameworks Tornado et Twisted ainsi que le protocole ouvert pour les systèmes de messagerie AMQP (Advanced Message Queuing Protocol).

Nova interagit avec le service d'identité keystone pour l'authentification, service de l'image glance pour récupérer et utilisé une image de disque, et le tableau de bord Horizon pour l'interface d'administration.

Nova utilise par défaut le port 8774.

Le service compute est constitué des domaines fonctionnels suivants et leurs composants sous-jacents:

- Nova-api

Accepte et répond à des appels d'API de l'utilisateur final. Aussi, il initie la plupart des activités d'orchestration, telles que l'exécution d'une instance.

- Nova-compute

Un démon de travail (worker) qui crée et met fin à des instances de machines virtuelles via les API de l'hyperviseur. Par exemple, XenAPI pour XenServer / XCP, libvirt pour KVM ou QEMU, VMwareAPI pour VMware, et ainsi de suite.

- Nova-scheduler

C'est un processus qui prend une demande de l'instance de machine virtuelle à partir de la file d'attente et détermine sur quel hôte il doit être exécuté.

- Nova-conductor (depuis Grizlly seulement)

Médie les interactions entre nova-compute et la base de données. Il vise à éliminer les accès directs à la base de données de nuage fait par nova-compute.

- Nova-novncproxy

C'est un démon qui fournit un proxy pour l'accès aux instances en cours d'exécution via une connexion VNC.

- Nova-consoleauth

C'est un démon qui gère l'autorisation des jetons pour les utilisateurs.

- Nova-cert

C'est un démon qui gère des certificats x509.

5.4.2. OpenStack Networking (Neutron)

Ce module permet la mise en place de réseaux virtuels. Historiquement connu sous le nom de Quantum il fut renommé en Neutron dans la version Havana.

Neutron est un module à part entière qui permet de définir des réseaux logiques et d'exposer certaines instances virtuelles au public ou bien au contraire d'en conserver d'autres en visibilité interne uniquement. Il s'agit de la gestion des gateways que l'on retrouve maintenant chez AWS au sein des EC2. A noter que par défaut, une instance Nova n'est pas visible de l'extérieur et ne possède qu'une adresse IP interne : il faut lui associer un réseau visible de l'extérieur via une gateway et ensuite lui attribuer une ‘‘floating IP’’ (équivalent d'une adresse IP publique) impérativement : la ‘‘floating IP’’ a une fonction similaire aux Elastic IPs d'AWS. Une fois attribuée, on peut la conserver et l'associer à telle ou telle instance pour qu'elle soit accessible de l'extérieure sur cette adresse IP.

Neutron utilise par défaut le port 9696, il se décline en plusieurs sous composants :

- Neutron-server

Il s'agit du démon qui expose aux autres composants l'API Neutron et va passer les requêtes utilisateurs vers les autres démons. Il agit comme le contrôleur du réseau.

- Neutron-plugin agent

Ce démon sera installé sur les hyperviseurs et permet de gérer les configurations des switchs virtuels. L'agent dépend du type de plugins utilisé. Il existe en effet une variété de plugins : OpenvSwitch, Cisco, Nicira NVP, etc…

Le choix du plugin dépend du budget et des besoins car des solutions comme Nicira NVP sont des solutions commerciales.

- Neutron-dhcp-agent

Ce démon propose un serveur DHCP pour chacun des tenants. Il est identique pour l'ensemble des plugins.

- Neutron-l3-agent

Ce démon se charge de la gestion du routage et du NAT afin de permettre l'accès externe aux instances de machines virtuelles pour les réseaux de chaque tenant. Il est identique pour l'ensemble des plugins.

- Neutron-metadata-agent

Ce démon permet de gérer le serveur de metadata permettant d'injecter des données lors du lancement des instances de machines virtuelles. Il est identique pour l'ensemble des plugins.

5.4.3. OpenStack Imaging Service (Glance)

Ce module fournit les services de stockages, d'enregistrements et de distributions pour les images disques des différentes instances. Il fournit également une API compatible REST permettant d'effectuer des requêtes dans une base de donnés pour récupérer des informations sur les images disques de machines virtuelles.

Il s'agit des images d'OS (Ubuntu, Cirros, Windows...) à partir desquelles on peut démarrer les instances. Il s'agit de la même brique que les AMIs d'AWS.

OpenStack Imaging Service se décline en deux sous composants:

- Glance-api

Accepte et répond aux appelles vers sa propre API. Les requêtes d'interrogation, de récupération, de téléchargement ou de suppression d'images sont reçu par glacne-api, et sont transmit par la suite à glance-registry. Par défaut, glance-api écoute sur le port 9292.

- Glance-registry

Le processus glance-registry stocke et récupère les méta-datas à partir d'une base de données. Par défaut glance-registry écoute sur le port 9191.

5.5. Les composants optionnels

5.5.1. OpenStack Block Storage (Cinder)

Ce module fournit un stockage persistant en mode block. Historiquement ce module faisait partie de Nova et se nommait nova-volume, depuis la version Folsom. Nova-volume a disparu pour laisser sa place à Cinder.

Cinder donne la possibilité d'ajouter des volumes réseau (comme les disques logiques d'un SAN) aux instances démarrées. Il s'agit d'un disque (raw device) qui est attaché à une instance et une seule à un instant donné. Il faut formater ce disque pour le monter et l'utiliser sur l'instance. Il s'agit de la même fonctionnalité que les EBS d'AWS.

OpenStack Block Storage se décline en plusieurs sous composants :

- Cinder-api

Accepte les demandes API et les achemine à cinder-volume.

- Cinder-volume

Répond aux demandes de lecture et d'écriture à la base de données de cinder, interagit avec d'autres processus (comme cinder-scheduler).

- Cinder-scheduler

C'est un démon qui détermine l'hôte optimal sur laquelle il va créer le volume.

5.5.2. OpenStack Object Storage (Swift)

Ce module permet la création d'espaces de stockage redondants et évolutives. Il s'agit d'un système de fichiers surtout conçu pour le stockage à long terme de gros volumes. Il utilise une architecture distribuée offrant plusieurs points d'accès pour éviter les SPOF (Single Point Of Failure) qui entraînent l'arrêt complet du système lors d'une panne.

5.5.3. OpenStack Telemetry (Ceilometer)

Ce module permet de mesurer la consommation d'un projet sur une plate-forme OpenStack. Ceilometer peut-être utilisé comme outil de facturation. Ceilometer sera certainement intégré officiellement à la prochaine version d'OpenStack (Icehouse).

5.5.4. OpenStack Orchestration (Heat)

Heat est une plateforme d'orchestration qui permet aux utilisateurs de provisionner plus facilement les Cloud basés sur OpenStack. Heat dispose d'un langage de template qui permet à une application de communiquer avec le Cloud afin d'accéder aux serveurs et d'accélérer le déploiement des instances virtuelles.

En utilisant Heat, les développeurs d'applications peuvent alimenter des templates directement dans le programme, l'application déployant automatiquement les ressources nécessaires Heat peut aussi lancer des applications, créer des machines virtuelles et automatiser l'ensemble du

processus. Il fournit également des compatibilités avec la plateforme d'orchestration d'Amazon Web Service nommé CloudFormation. Une information qui signifie que les templates AWS seront fonctionnels dans des environnements OpenStack.

6. Conception

6.1. Identification des Acteurs

Dans cette section nous allons présenter les acteurs qui interagissent avec le cloud OpenStack, les acteurs de notre système sont divisés en deux groupes « humain » et « non humain », nous commençons par lister :

- Les acteurs humains et leurs rôles

Administrateur Cloud « OpenStack Admin » : Il gère les différentes instances et modèles du Cloud

Client Cloud : Exploite les ressources du système d'information de la société

- Les acteurs non humains :

Serveur hyperviseur : C'est le serveur qui nous permettra de d'exploiter ces ressources de virtualisation

Serveur de stockage primaire : Il permet de stocker les instances

Serveur de stockage secondaire : Il permet de stocker les modèles et templates des instances ainsi que des images

Tableau 8: Tableau des acteurs Cloud « OpenStack »

Acteur	Type	Rôle	Système
Openstack_Admin	Principale	Gérer Openstack	Openstack
Client Cloud	Principale	Interagir avec le cloud	Openstack

6.2. Identification des messages :

Cette partie sert à identifier les différents messages entre notre système d'informations et les différents acteurs. Pour ce faire le tableau ci-dessous représente une description textuelle des messages émis et reçus pour chaque acteur vers ou depuis le système :

Tableau 9: description textuelle des messages émis et reçus pour chaque acteur

N°/Acteur	Acteur	Message émis	Message reçus
1/A	Administrateur Cloud	S'authentifier Vérifier les ressources systèmes disponibles Gérer instances Gérer les réseaux	Droit d'accès Etat des ressources disponibles Liste des instances Liste de gestion des réseaux
2/C	Client Cloud	Exploiter instance	Flux d'information suite à la consommation des

			ressources exploitées
3/S	Serveur hyperviseur	Gérer instances	Exploitation de l'hyperviseur
4/SP	Serveur stockage primaire	Gérer volume	Accès aux volumes
5/SS	Serveur de stockage secondaire	Gérer templates et ISO	Accès aux templates et ISO

6.3. Diagramme de contexte dynamique

Afin de ne pas surcharger inutilement le diagramme de contexte dynamique, nous avons omis les messages échangés et les ont remplacé par leurs numéros déjà décrit dans la partie précédente.

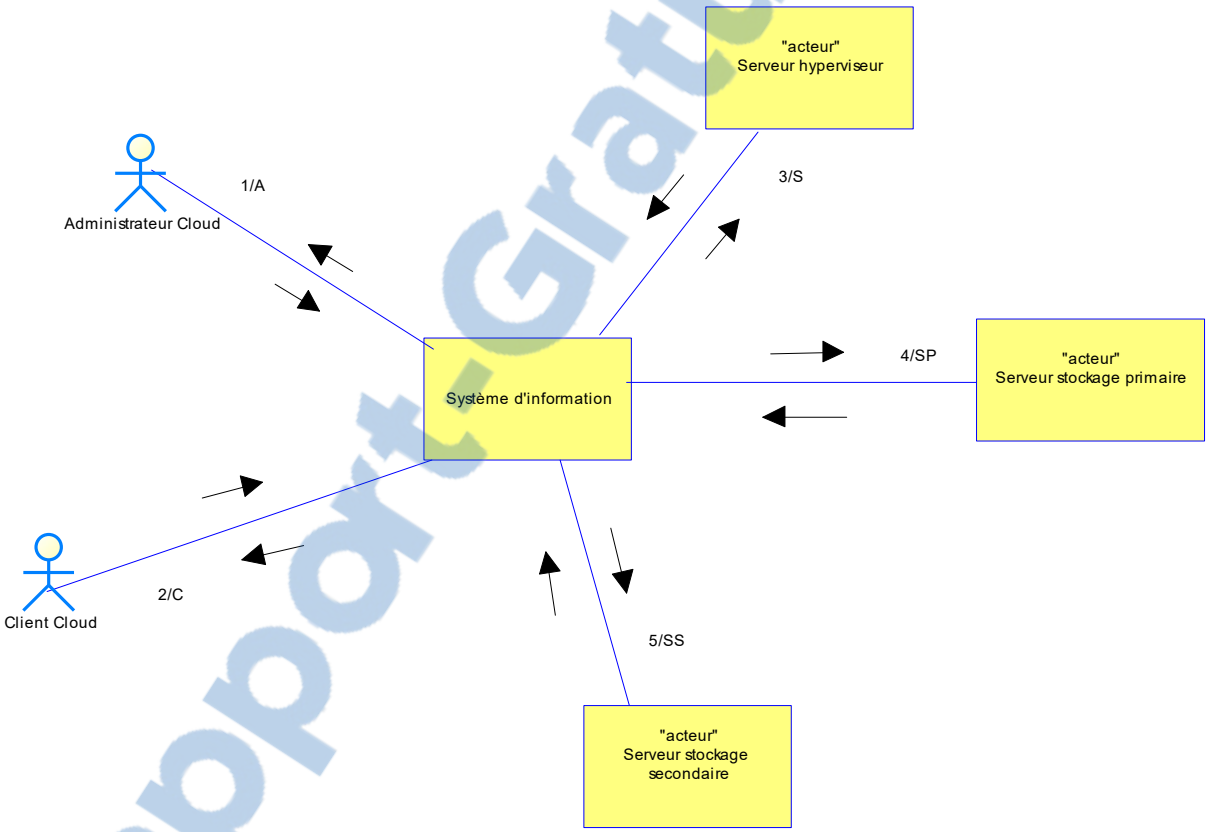

Figure 43 : Diagramme de contexte dynamique

6.4. Diagramme use case

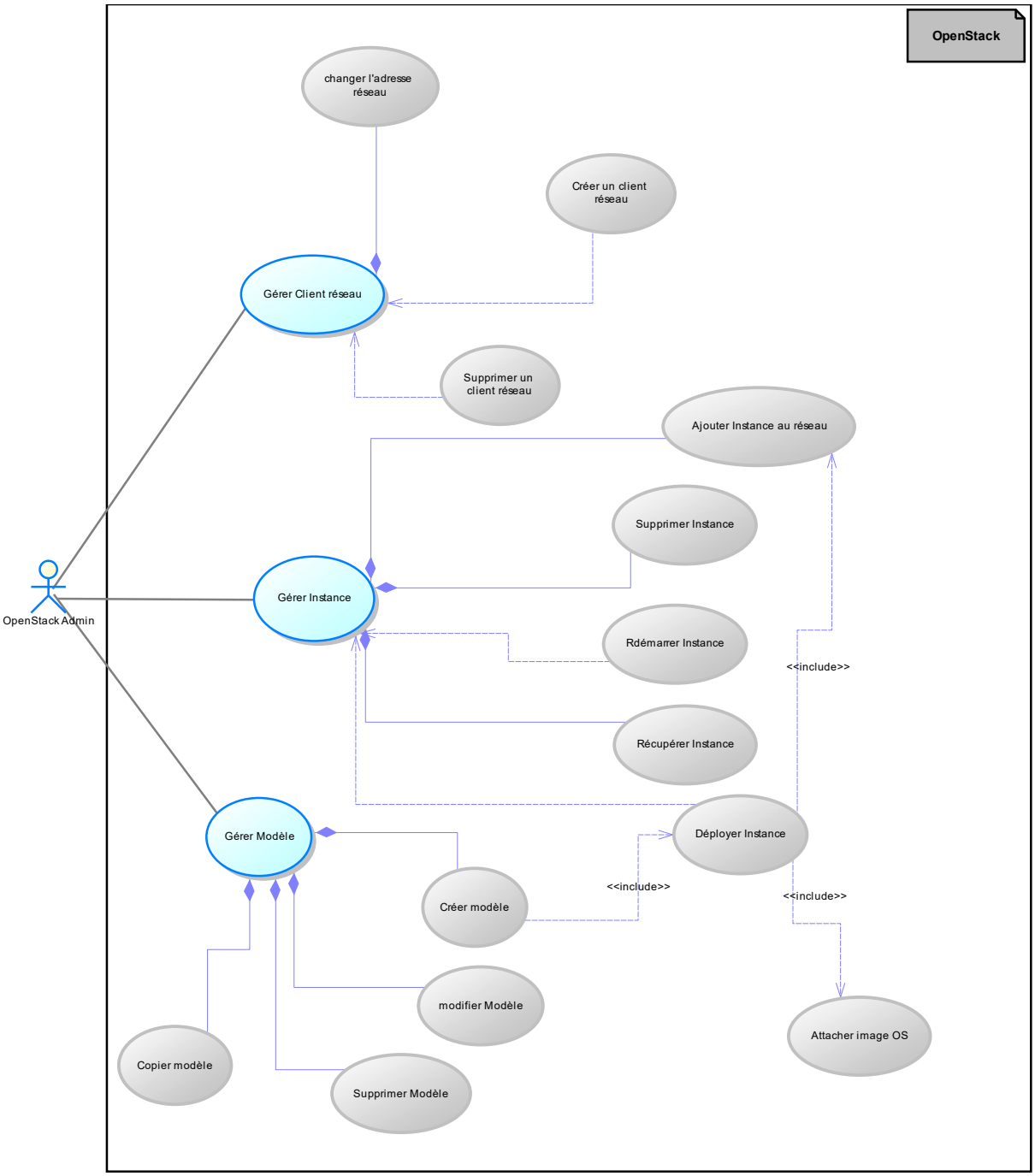

Figure 44: Use Case Générale OpenStack_Admin

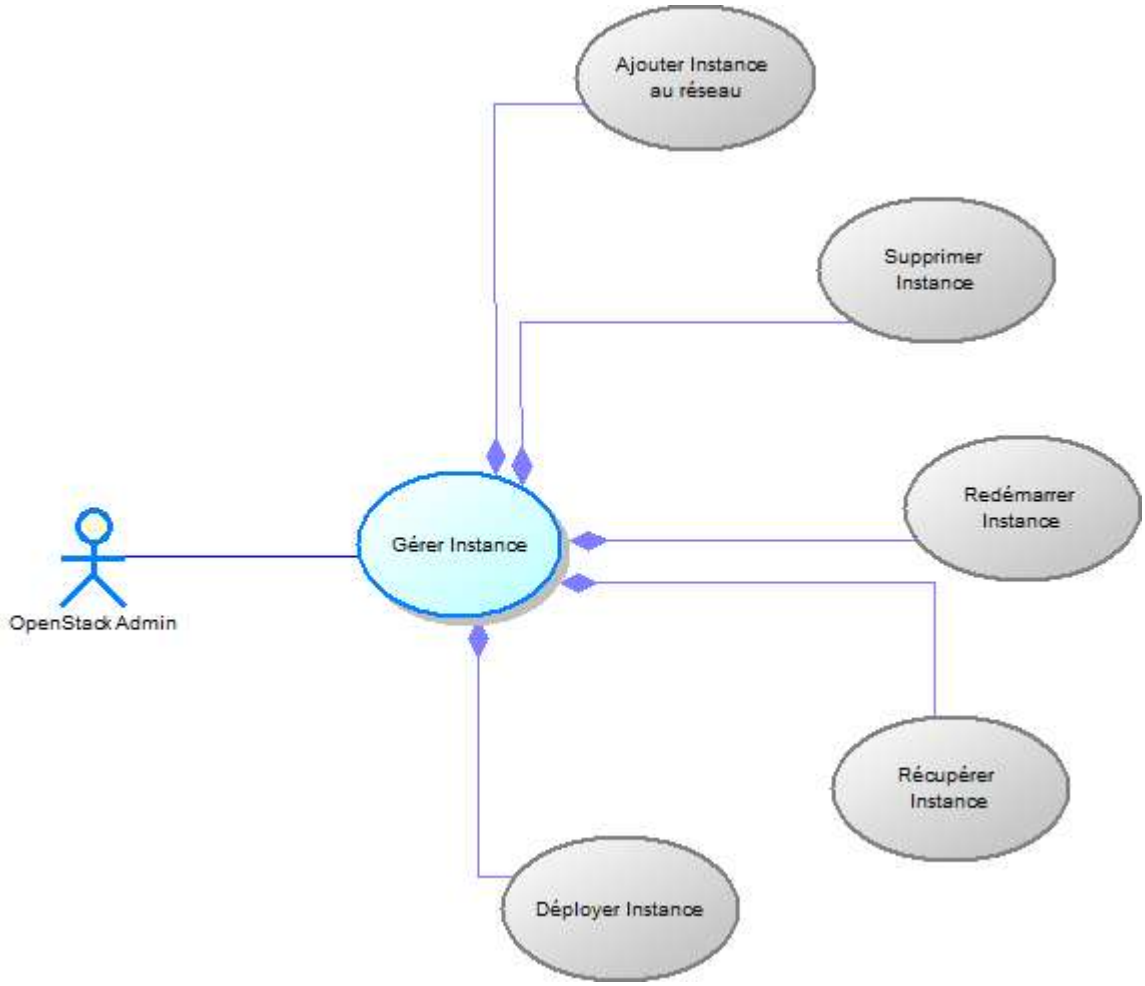

Figure 45: Use Case Gérer les instances

Description textuelle

Sommaire d'identification	
Titre	Gérer les instances
But	Gestion des instances
Résumé	L'administrateur OpenStack gère les instances
Acteur	Administrateur OpenStack
Description des enchaînements	
Pré-condition	Administrateur connecté
Post condition	Les instances ont été gérées
Description du scénario	L'administrateur accède au Cloud via le Dashboard Déployer des nouvelles instances Récupérer des instances Redémarrer des instances Supprimer des instances Ajouter des instances aux réseaux

6.5. Diagramme de séquence

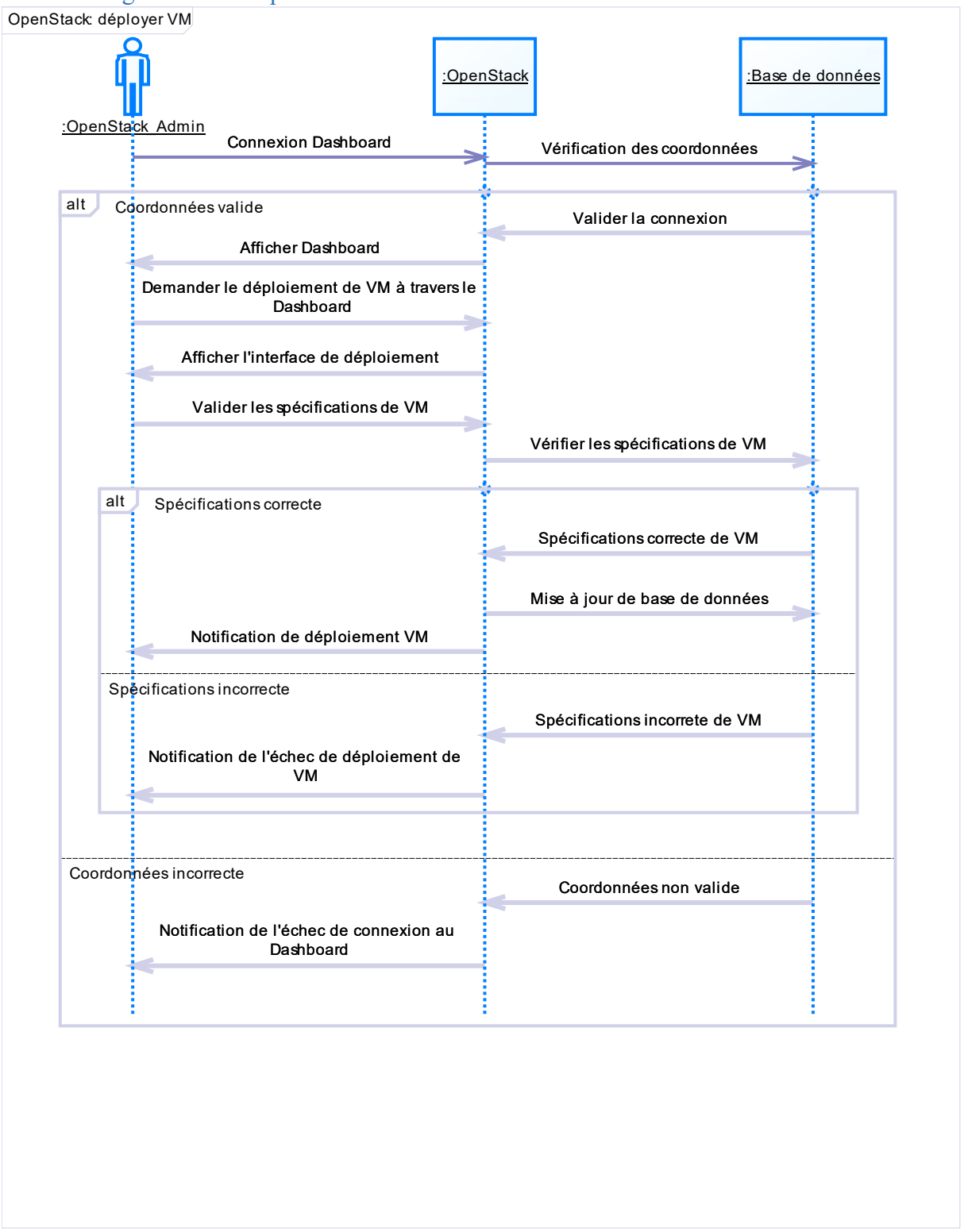

Figure 46: Diagramme de séquence « déployer VM »

7. Implémentation

Dans ce qui précède, nous avons énuméré les différents composants d'OpenStack qui peuvent être utilisés. Dans notre travail, nous avons choisi une architecture composée de cinq modules à savoir Nova, Neutron, keystone, Horizon et Glance vu que nous n'avons pas besoin d'installer tous les composants.

La figure suivante présente l'architecture que nous avons adoptée.

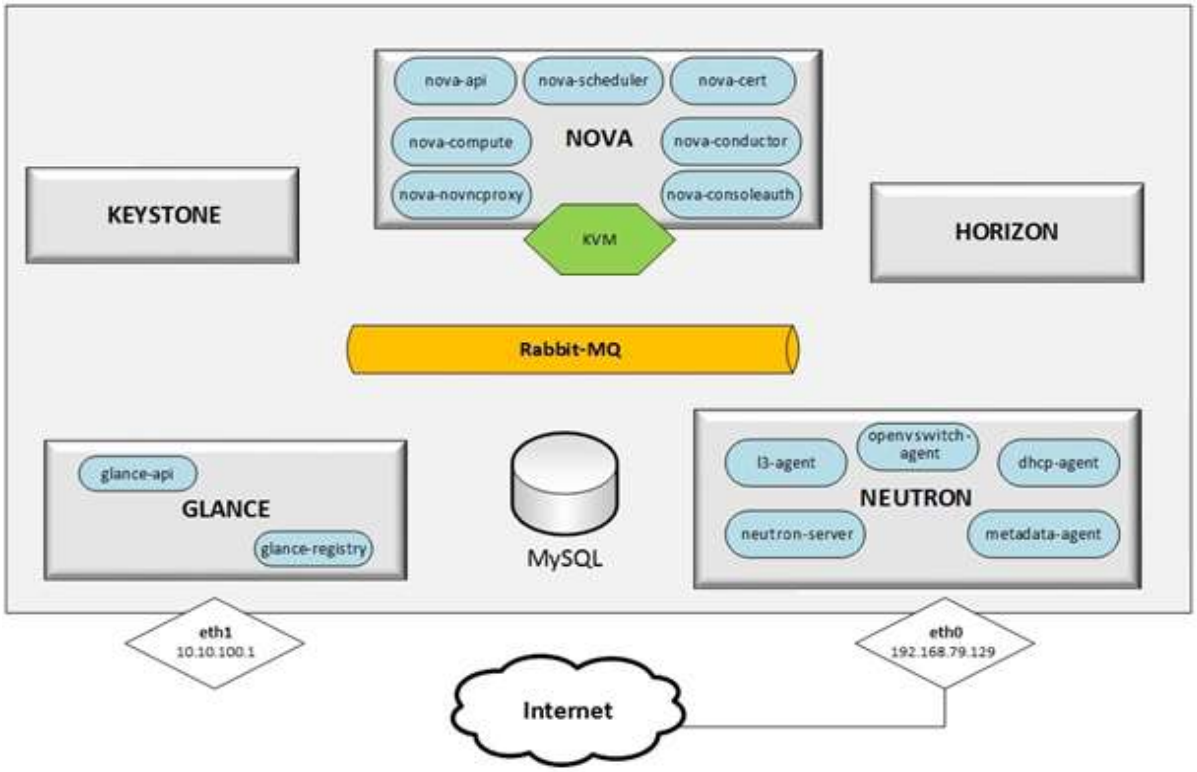

Figure 47: Architecture OpenStack utilisée

7.1. Préparation du système

7.1.1. Choix du système d'exploitation

OpenStack fonctionne actuellement sur plusieurs distributions linux telles qu'Ubuntu, redHate, Centos, Fedora. Pour des raisons de compatibilité, nous avons choisi la distribution Ubuntu (14.04 LTE server 64 bits).

Remarque : il est recommandé d'utiliser une version 64-bit pour éviter l'échec en cas de démarrage d'une instance en utilisant un système 64-bit.

La figure ci-dessous illustre le modèle de déploiement de la partie Cloud de notre projet

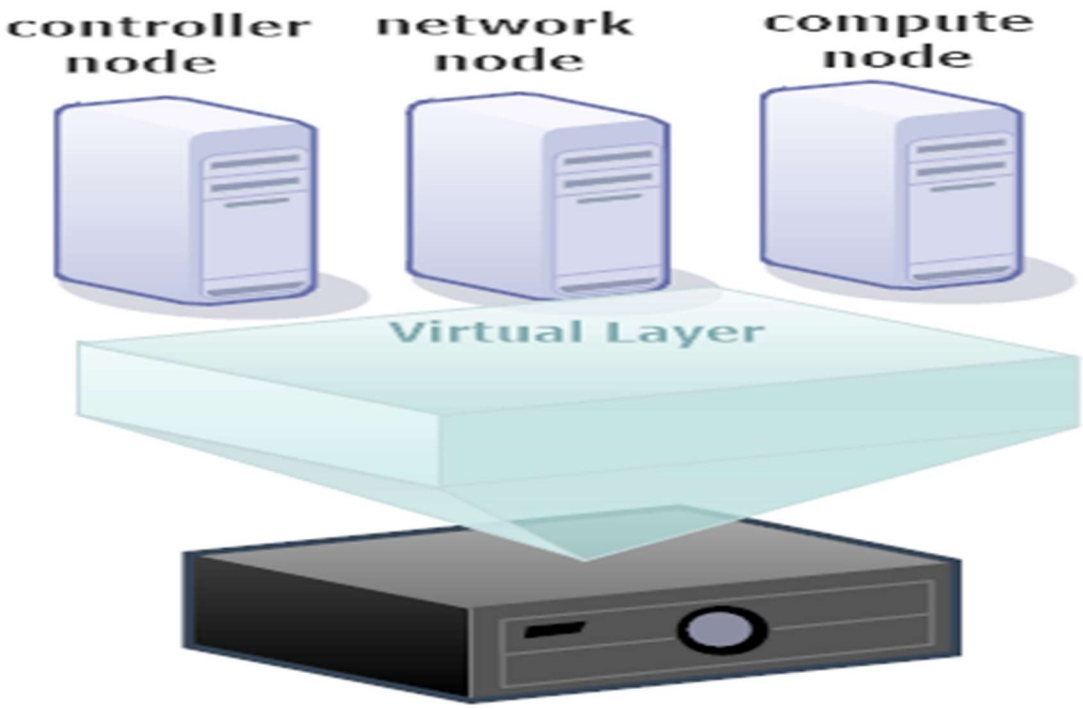

Figure 48: modèle de déploiement Openstack

Les trois nœuds ont comme système d'exploitation Ubuntu 14.04 LTE

Les connexions réseaux sont comme suit :

Un réseau de gestion reliant les trois machines virtuelles (Controller, Network et Compute)

Un réseau de tunneling acheminant la connexion externe du nœud Network au nœud Compute

- **Configuration des interfaces réseau**

Nous avons choisi d'utiliser deux réseaux distincts pour notre déploiement afin de séparer le trafic entre les services d'OpenStack et les autres communications au niveau de la machine.

- Le réseau de management d'OpenStack 10.10.100.0/24 est considéré comme le Backbone du Cloud et il fait transiter les communications entre les différents services d'OpenStack.
- Le réseau externe : 192.168.79.0/24 qui représente notre réseau local.

- **L'ajout des packages d'OpenStack**

Après l'installation d'ubuntu server, il faut tout d'abord ajouter les packages d'OpenStack et faire la mise à jour du système.

- L'ajout des packages

```
# apt-get install python-software-properties

# add-apt-repository cloud-archive:havana
```

- Mise à jour du système

```
# apt-get update && apt-get dist-upgrade
```

➢ Installation de MySQL et RabbitMQ

Mysql

La plupart des services d'OpenStack nécessitent une base de données pour stocker les informations.

- Installation de MySQL

```
# apt-get install python-mysqldb mysql-server
```

Après l'installation, il faut créer une base de données pour tous les composants d'OpenStack de notre déploiement sauf horizon qui ne nécessite pas une base de données.

RabbitMQ

RabbitMQ est un courtier de messages se basant sur le standard AMQP, c'est le service qui permet aux composants OpenStack de communiquer entre eux.

- Installation de RabbitMQ

```
# apt-get install rabbitmq-server
```

➢ Installation de KEYSTONE

- Installation

```
# apt-get install keystone
```

- Configuration

Il faut indiquer à keystone la chaine de connection à la base de données. Pour cela il faut modifier la section [sql] qui se trouve au niveau du fichier **/etc/keystone/keystone.conf.**

```
[sql]

connection = mysql://keystone:KEYSTONE_DBPASS@10.10.100.1/keystone
```

Après avoir installé le service d'identité (keystone), il faut créer des utilisateurs, des tenants et des rôles. Ceux-ci sont utilisés pour permettre l'accès aux différents services d'OpenStack.

Création des tenants

Au niveau d'Openstack un tenant représente un groupe d'utilisateurs ou un projet.

Nous allons créer deux tenants, un pour l'administrateur « ***admin*** » et un autre pour les services « ***services*** » pour regrouper tous les utilisateurs des services.

Création des utilisateurs

Nous allons créer un utilisateur administrateur qui est obligatoire, et un utilisateur pour chaque service qui va être installé afin de s'authentifier avec keystone.

Création des rôles

Pour assurer le bon fonctionnement de keystone, il est nécessaire de créer quatre rôles obligatoires à savoir «***admin***», « ***Member*** », «***KeystoneAdmin***» et «***KeystoneServiceAdmin***».

L'Ajout des rôles aux utilisateurs

Après la création des utilisateurs et des rôles, nous allons attribuer les rôles aux utilisateurs. Ces derniers se connectent toujours avec un tenant, et les rôles sont attribués aux utilisateurs dans des tenants.

- Au niveau de l'utilisateur admin

Nous allons ajouter des rôles **admin**, **keystoneAdmin**, **keystoneServiceAdmin** qui sont obligatoires à l'utilisateur **admin** lors de la connexion avec le tenant **admin**.

```
# keystone user-role-add --user=admin --tenant=admin --role=admin

# keystone user-role-add --user=admin --tenant=admin --role=KeystoneAdmin

# keystone user-role-add --user=admin --tenant=admin --role=KeystoneServiceAdmin
```

- Au niveau des autres utilisateurs

L'Ajout du rôle **admin** aux autres utilisateurs lors de la connexion avec le tenant admin.

```
# keystone user-role-add --user=glance --tenant=services --role=admin

# keystone user-role-add --user=nova --tenant=services --role=admin

# keystone user-role-add --user=neutron --tenant=services --role=admin
```

Création des services et leurs points d'accès (endpoints)

Il va falloir créer un service pour **Keystone, Glance, Nova** et **Neutron** ainsi que leurs points d'accès.

Voici un exemple de création du service Keystone ainsi que son point d'accès et cela va de même pour les autres services.

- Création du service

```
# keystone service-create --name=keystone --type=identity --description="Identity Service"
```

- Point d'accès

```
#keystone endpoint-create --service-id=1f32639b2f3541d79381c8def08c368e --region RegionOne --publicurl=http://192.168.79.129:5000 --internalurl=http://10.10.100.1:5000 --adminurl=http://10.10.100.1:35357
```

Remarque : le détail des trois fichiers de configuration de Keystone se trouve en **annexe [1].**

➢ Installation de GLANCE

- Installation

```
# apt-get install glance python-glanceclient
```

➢ Configuration

La configuration de glance se fait autour de quatre fichiers **:**

- Au niveau des deux fichiers : /etc/glance/glance-api-paste.ini et /etc/glance/glance-registry-paste.ini, sous la section [filter:authtoken], nous allons indiquer au service d'image glance d'utiliser le Service d'identité keystone pour l'authentification en précisant le port et l'adresse de keystone ainsi que le user et le password de glance.
- Au niveau des deux fichiers /etc/glance/glance-api.conf et /etc/glance/glance-registry.conf nous allons modifier sous la section [default] nous allons ajouter la chaine de connection à la base de donné glance.

Remarque : le détail des deux fichiers de configuration de glance se trouve **en annexe [2]**.

➢ Installation de NEUTRON

Avant d'installer neutron, il faut installer un plugin qui est chargé de l'exécution du service neutron. Pour cela nous avons choisit d'installer le plugin openvswitch qui est le plus utilisé.

- Installation du plugin openVSwitch

```
#apt-get install openvswitch-controller openvswitch-switch openvswitch-datapath-dkms
```

- Création des bridges

Open vSwitch Bridge est une implémentation logicielle d'un switch ethernet.

Nous allons créer deux bridges:

- ✓ br-int pour interconnecter les différentes instances et gérer le réseau interne.
- ✓ br-ex qui est utilisé par neutron l3-agent afin de gérer le trafic depuis le réseau externe.

```
#ovs-vsctl add-br br-int

#ovs-vsctl add-br br-ex
```

Nous allons ajouter une interface réseau qui est connecté au réseau externe 192.168.79.0/24 au bridge externe br-ex**.** Pour cela nous allons ajouter l'interface eth0 au bridge br-ex.

```
#ovs-vsctl add-port br-ex eth0
```

- **Installation de Neutron**

```
#apt-get install neutron-server neutron-plugin-openvswitch neutron-plugin-openvswitch-agent dnsmasq neutron-dhcp-agent neutron-l3-agent neutron-metadata-agent
```

- **Configuration**

La configuration de neutron s'articule autour de six fichiers de configuration :

- Au niveau du fichier /etc/neutron/api-paste.ini nous allons modifier la section [filter :authtoken] pour indiquer à neutron d'utiliser le service keystone pour l'authentification.
- Au niveau du fichier /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini nous allons modifier :
  - ✓ La section [**DATABASE**] pour indiquer à OVS la chaine de connection à la base de données nova.
  - ✓ La section **[ovs**] affin de presiser le type de réseau et les bridge utilisés.
  - ✓ La section [**securitygroup**] pour indiquer le driver du firewall (iptables).
- Au niveau du fichier **/etc/neutron/metadata_agent.ini,** nous allons indiquer l'url ainsi que la région d'authentification au metadata_agent ainsi que l'adresse et le port qu'il va utiliser.
- Au niveau du fichier **/etc/neutron/neutron.conf,** nous allons tous d'abord indiquer à neutron l'adresse du courtier de message RABBITMQ, par la suite nous allons modifier la section **[DATABASE]** afin d'indiquer à neutron la chaine de connection à la base de données nova.
- Au niveau de fichier **/etc/neutron/l3_agent.ini,** sous la section **[DEFAULT]** nous allons autoriser neutron l3 agent pour utiliser le concept des namespaces, nous allons indiquer le bridge externe br-ex.
- Pour la configuration dhcp_agent, nous allons modifier la section **[DEFAULT]** au niveau de fichier **/etc/neutron/dhcp_agent.ini** pour indiquer à neutron que nous allons utiliser le processus Dnsmasq comme serveur dhcp et nous allons autoriser l'utilisation des network namespaces.

➢ Installation de NOVA

Avant l'installation de nova, il faut tout d'abord installer un hyperviseur qui sera utilisé par nova pour la création des instances de machines virtuelles. Nous avons choisis KVM comme hyperviseur pour notre déploiement.

- Installation du KVM

```
#apt-get install -y kvmlibvirt-bin pm-utils
```

➢ Installation des différents composants de nova

```
# apt-get install nova-api nova-cert novnc nova-consoleauth nova-scheduler nova-novncproxy nova-doc nova-conductor nova-compute-kvm
```

➢ Configuration

La configuration de nova s'articule autour de trois fichiers de configuration :

- Au niveau du fichier **/etc/nova/api-paste.ini,** nous allons modifier la section **[filter :authtoken]** pour indiquer à nova d'utiliser le service keystone pour l'authentification.
- Au niveau du fichier **/etc/nova/nova.conf**, nous allons :
  - ✓ Indiquer la chaine de connection à la base de donné de nova.
  - ✓ Spécifier l'adresse et le port des services glance et neutron qui sont utilisé par nova.
  - ✓ Configurer nova pour utiliser RABITMQ comme courtier de messages.
  - ✓ Configurer un accès distant à la console des instances de machines virtuelles.
- Au niveau du fichier **/etc/nova/nova-compute.conf** sous la section **[DEFAULT]** nous allons indiquer à nova d'utiliser kvm comme hyperviseur, ainsi que le type des interfaces virtuels et le nom du bridge interne à utiliser (br-int).

➢ Installation d'HORIZON

```
#apt-get  install openstack-dashboard memcached
```

8. Test

Maintenant on peut administrer OpenStack à partir de l'interface web **192.168.79.129/horizon.**

Après authentification, l'utilisateur peut accéder à l'interface d'accueil illustrée ci-dessous

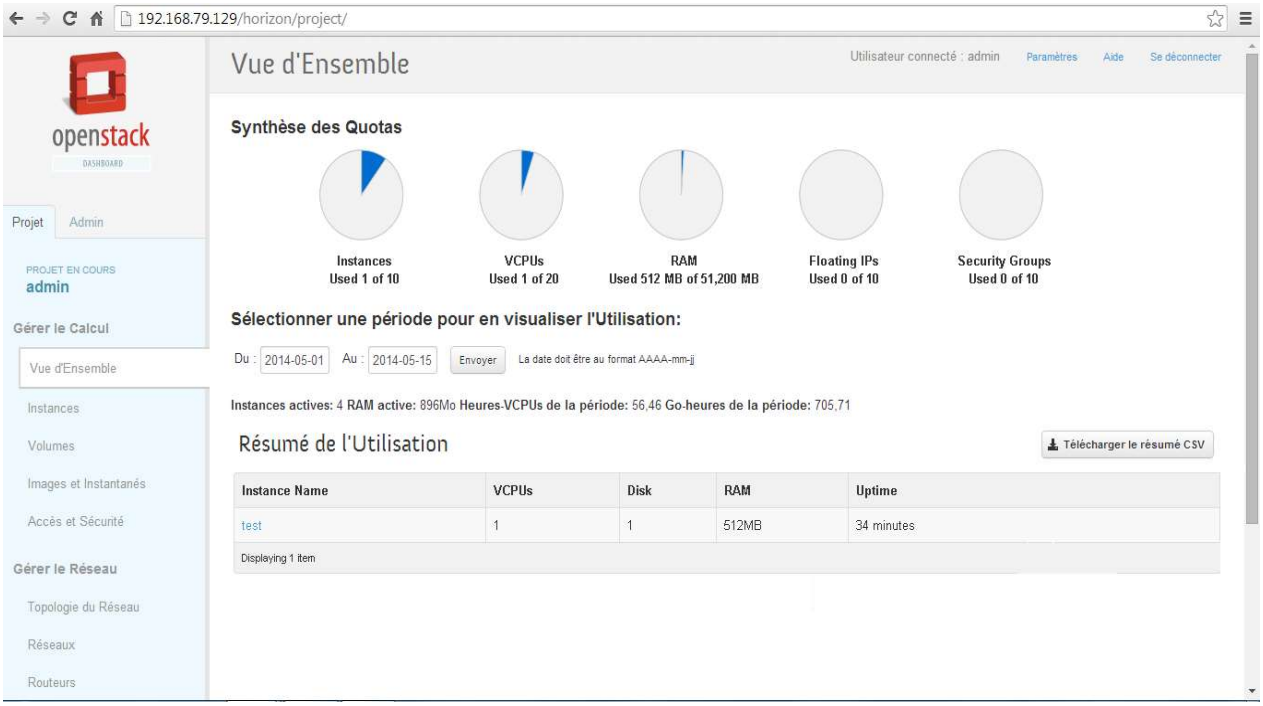

Figure 49 Interface d'accueil d'OpenStack

III. Conclusion :

Au niveau de ce chapitre nous avons décomposé notre travail en trois boucles principales selon le modèle sashimi. Dans la première boucle, nous avons identifié notre IaaS ainsi notre hyperviseur à mettre en place. Dans la deuxième, nous avons défini les caractéristiques et les composants de Vmware Vsphere et nous avons mis en place notre couche de virtualisation. Pour la dernière boucle, après avoir présenté les principales caractéristiques de notre solution OpenStack, les différentes versions et l'architecture conceptuelle avec ses différents composants, nous avons présenté les composants nécessaires pour la mis en place et l'installation de notre plateforme afin de déployer un Cloud privé.

CONCLUTION GENERALE

Durant ce projet de fin d'études nous avons tenté de mettre en place une plateforme de virtualisation VMware et un Cloud privé basé sur Openstack qui aura pu améliorer la qualité du système d'information de SERVICOM.
Ce projet s'est déroulé en quatre grandes parties :

Premièrement, nous avons réaménagé l'infrastructure réseau inter et intra sites pour assurer une haute disponibilité du réseau et remédier aux lacunes de l'installation réseau présente.

Deuxièmement, nous avons effectué une étude théorique ciblée et concise pour comprendre les notions de la virtualisation et de Cloud Computing, puis nous avons procédé à une étude comparative de différentes solutions de virtualisation et des solutions IaaS Open Source.

Troisièmement, nous avons mis en place une plateforme de virtualisation qui est basée sur des produits VMware ce qui a comme conséquence de garantir la haute disponibilité. De plus, le système assure une bonne élasticité car les augmentations des besoins en ressource seront facilement satisfaites en ajoutant le matériel nécessaire.

Quatrièmement, nous avons déployé un PaaS (Platfom as a Service)basé sur l'IaaS Openstack qui devait contenir les outils et service dont la société a besoin pour avoir un rendement optimal, être compétitif sur le marché et assurer la bonne coordination entre les différents filiales du groupe SERVICOM.

Cette expérience au sein de SERVICOM, dans le cadre de mon projet de fin d'études, m'a permis d'acquérir de précieuses connaissances techniques (la virtualisation et le Cloud Computing) et de me familiariser avec un environnement de travail professionnel.

LIMITIONS ET PERSPECTIVES

Limitations

Notre projet nécessite plus de ressources matérielles (CPU, RAM, Stockage,…) pour pouvoir faire tourner les outils et services dont la société a besoin dans notre plateforme Cloud.
Les ressources demandées n'ont pas été fournies par la société ce qui nous a empêché de déployer les outils et les services exigés par la société.

Perspectives

Notre solution pourra être améliorée en y ajoutant des composants et services fournis par VMware et OpenStack.
Le projet qui a été mis en place est un environnement de test et démonstration qui sera par la suite redéployé en un environnement de production suite au investissement qui auront lieu dans un avenir proche et suite à la finalisation du partenariat avec VMware.

BIBLIOGRAPHIE :

Livres de référence :

Nicolas GREVET, Le Cloud Computing : Evolution ou Révolution ?

Pascal Sauliere, Cloud Computing et sécurité, Microsoft France

Sites web de références :

[1] http://reseau-informatique.prestataires.com/conseils/virtualisation

[2] http://fr.wikipedia.org/wiki/Cloud_computing

[3] http://www.saizenitc.com/training/big-data-cloud-computing/

[4] http://icp.ge.ch/sem/cms-spip/spip.php?article962

[5] https://en.wikipedia.org/wiki/Waterfall_model#Model

[6] http://open.eucalyptus.com/

[7] http://opennebula.org/

[8] http://www. cloudstack.apache.org/

[9] http://www.openstack.org/

[10] https:// http://www.vmware.com/files/fr/pdf/support/VMware-Introduction-to-vSphere-PG-FR.pdf

[13] http://docs.openstack.org/user-guide/user-guide.pdf

[14] http://www.mirantis.com/blog/openstack-networking-single-hostflatdhcpmanager/

[15] http://www.ubuntu.com

ANNEXE :

Annexe 1 : **Ajout du serveur ESXI au domaine locale de SERVICOM**

La figure 51 ci-dessous illustre la configuration et l'ajout au domaine « servi.local » du notre serveur « ESXI »

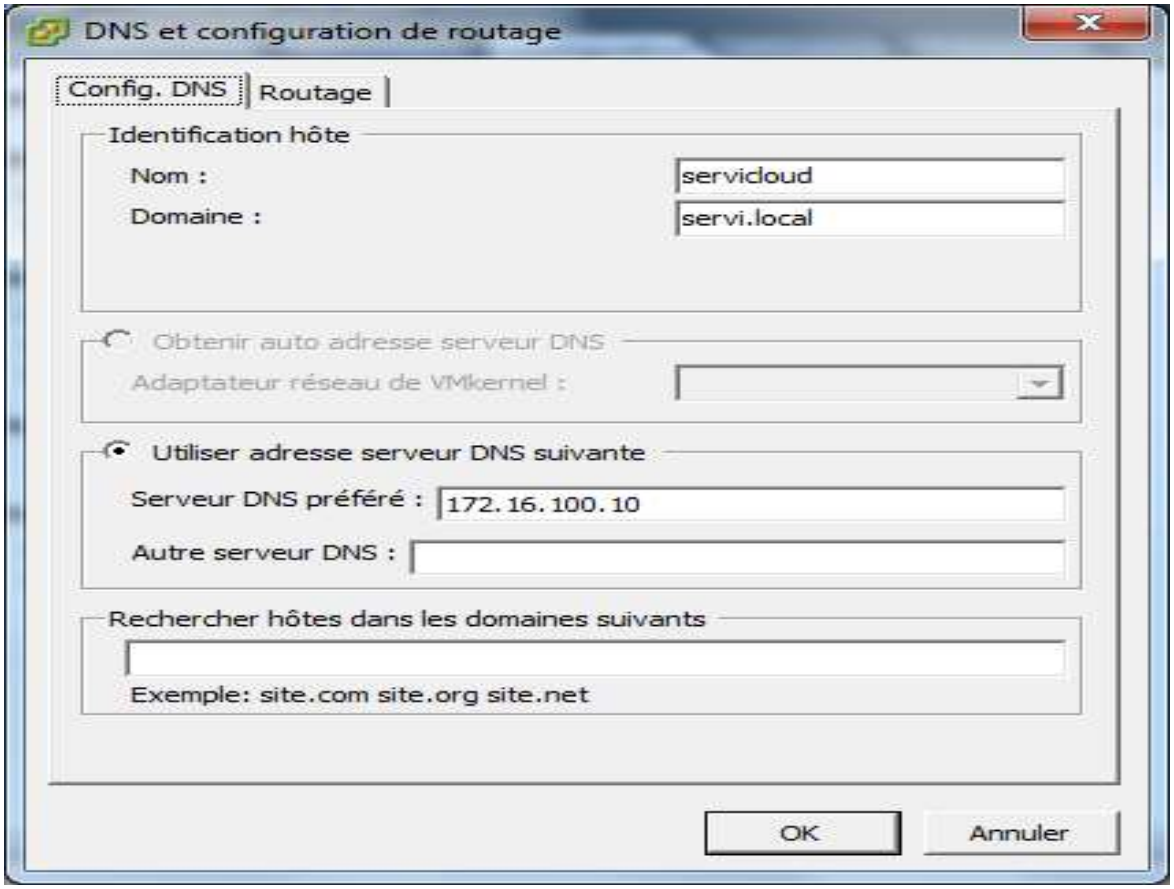

Figure 50: Ajout domaine servi.local au serveur « ESXI »

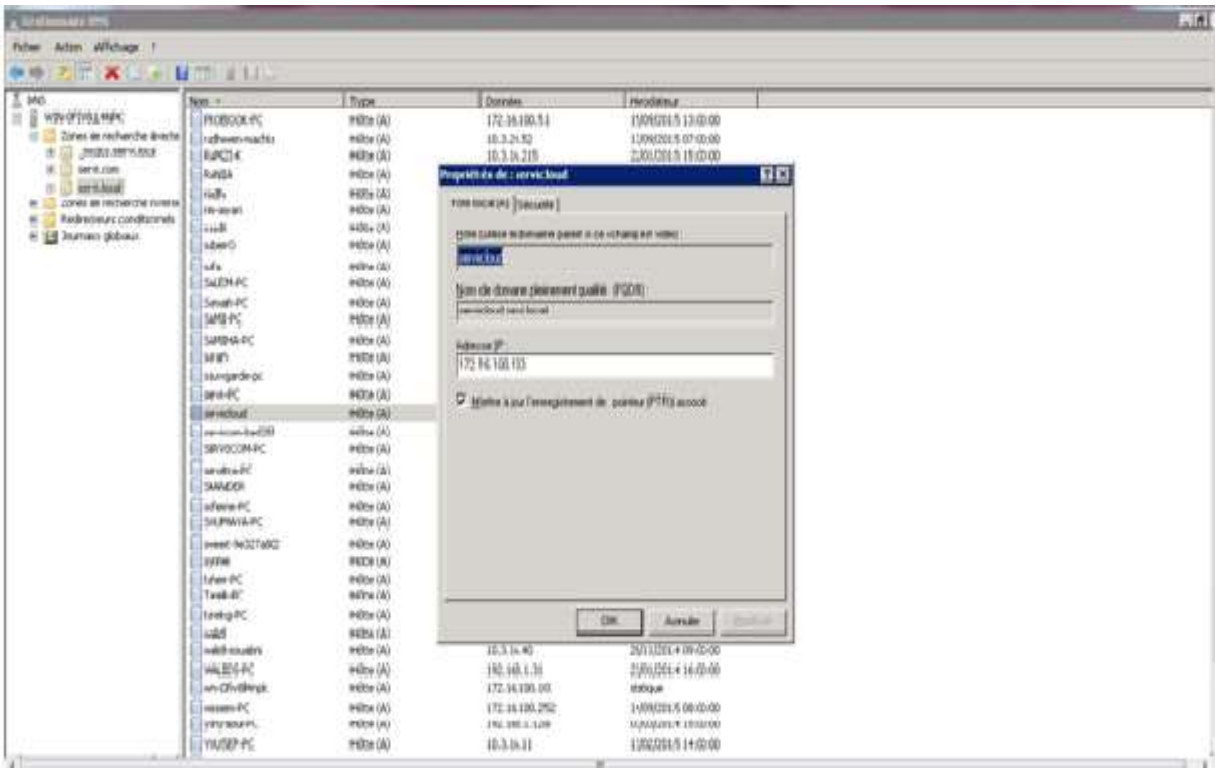

Figure 51: Ajout de l'hôte « ESXI » au niveau du serveur DNS local

La figure 52 décrit l'association de l'adresse IP 172.16.100.133 du serveur « ESXI » à notre domaine local « servi.local »

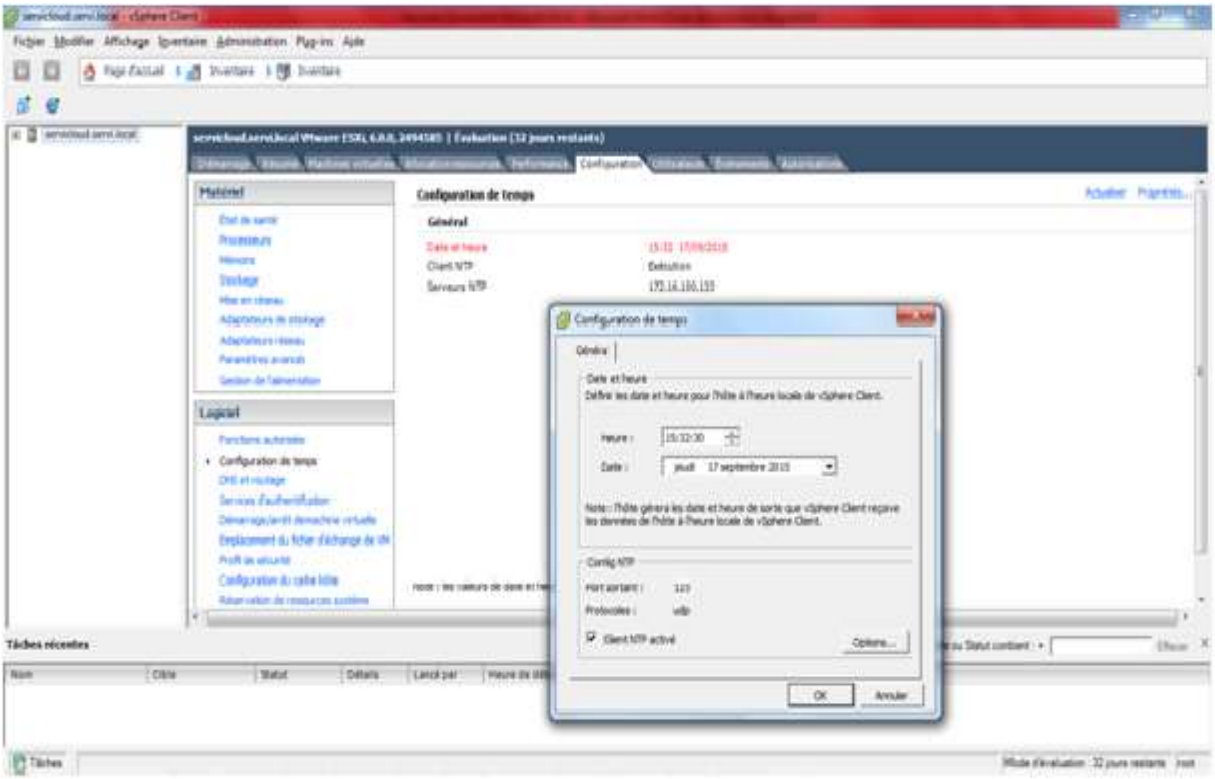

Figure 52: Réglage de l'heur du serveur DNS au niveau de l'hôte « ESXI »

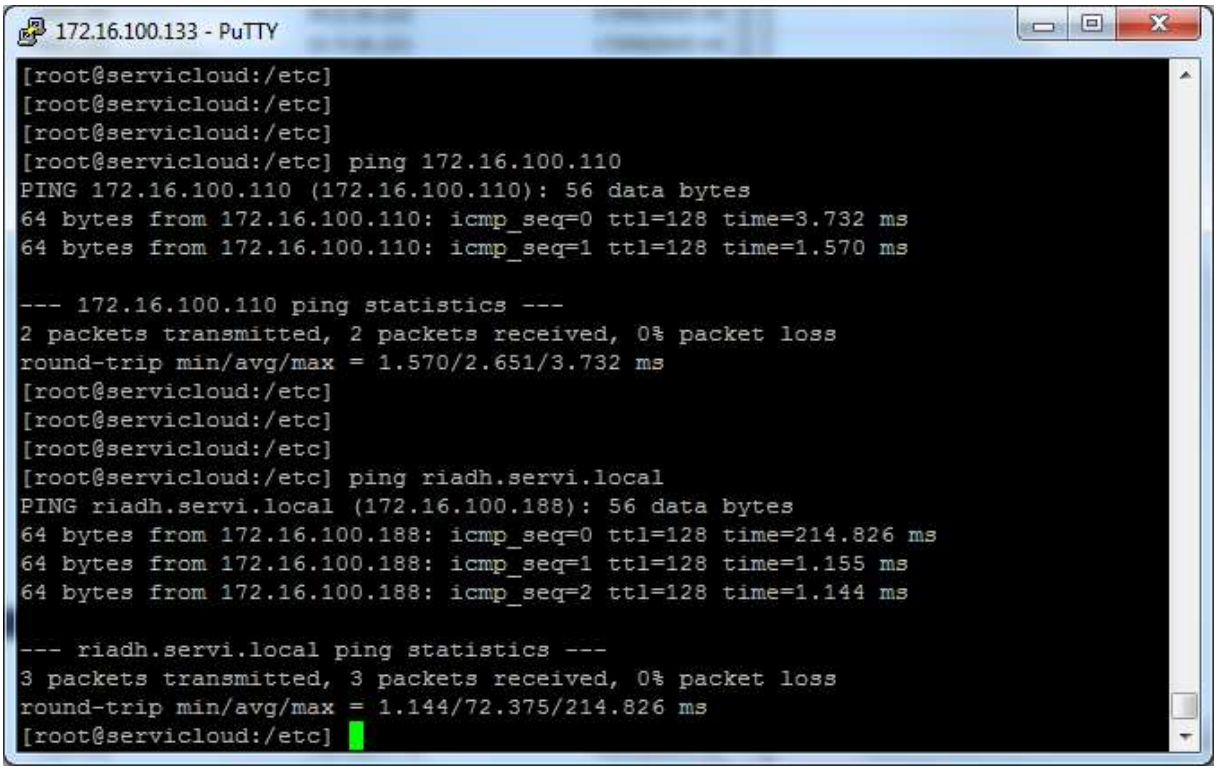

Figure 53: Test de fonctionnement DNS

Dans la figure 54 nous avons testé le bon fonctionnement de l'ajout de notre hote « ESXI » au domaine local « servi.local »

Annexe 2 : **Configuration Keystone**

I-)Editer le fichier :« /etc/keystone/keystone.conf » et modifier les sections suivantes :

1-) Dans la section [DEFAULT] définir la valeur initiale du token :

```
[DEFAULT]
...
admin_token = ADMIN_TOKEN
```

Remplace ADMIN_TOKEN par une valeur aléatoire

2-) Dans la section [database] section, configurer l'accès à la base de données :

```
[database]
...
connection = mysql://keystone:KEYSTONE_DBPASS@controller/keystone
```

Replace KEYSTONE_DBPASS with the password you chose for the database.

3-) Dans la section [memcache] configurer le service Memcache :

```
[memcache]
...
servers = localhost:11211
```

4-) Dans la section [token] configurer l'UUID token provider et Memcached driver:

```
[token]
...
provider = keystone.token.providers.uuid.Provider
driver = keystone.token.persistence.backends.memcache.Token
```

5-) Dans la section [revoke] configure la révocation du SQL driver:

```
[revoke]
...
driver = keystone.contrib.revoke.backends.sql.Revoke
```

Pour configurer le serveur Apache HTTP Server

II-) Editer le fichier « /etc/apache2/apache2.conf » et configuer l'option ServerName pour référencer le controller node:

```
ServerName controller
```

Créer le fichier « /etc/apache2/sites-available/wsgi-keystone.conf » avec le contenu suivant

```
Listen 5000
Listen 35357
<VirtualHost *:5000>
WSGIDaemonProcess keystone-public processes=5 threads=1 user=keystone
```

```
display-name=%{GROUP}
WSGIProcessGroup keystone-public
WSGIScriptAlias / /var/www/cgi-bin/keystone/main
WSGIApplicationGroup %{GLOBAL}
WSGIPassAuthorization On
<IfVersion >= 2.4>
ErrorLogFormat "%{cu}t %M"
</IfVersion>
LogLevel info
ErrorLog /var/log/apache2/keystone-error.log
CustomLog /var/log/apache2/keystone-access.log combined
</VirtualHost>
<VirtualHost *:35357>
WSGIDaemonProcess keystone-admin processes=5 threads=1 user=keystone
display-name=%{GROUP}
WSGIProcessGroup keystone-admin
WSGIScriptAlias / /var/www/cgi-bin/keystone/admin
WSGIApplicationGroup %{GLOBAL}
WSGIPassAuthorization On
<IfVersion >= 2.4>
ErrorLogFormat "%{cu}t %M"
</IfVersion>
LogLevel info
ErrorLog /var/log/apache2/keystone-error.log
CustomLog /var/log/apache2/keystone-access.log combined
</VirtualHost>
```

Annexe3: Configuration Glance

1-) Editer le fichier « /etc/glance/glance-api.conf » et effectuer les modifications suivantes:

Dans la section [database] configurer l'accés à la base de données:

```
[database]
...
connection = mysql://glance:GLANCE_DBPASS@controller/glance
```

Remplacer GLANCE_DBPASS par un mot de passe convenable

Dans la section [keystone_authtoken] et [paste_deploy] configurer l'accès au service d'identité :

```
[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
username = glance
password = GLANCE_PASS
[paste_deploy]
...
flavor = keystone
```

Remplacer GLANCE_PASS par un mot de passe convenable

Dans la section [glance_store] configurer l'emplacement du fichier système local et l'emplacement du fichier image

[glance_store]

```
...
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
```

Dans la section [DEFAULT] configurer le « notification_driver » de noop pour désactiver les notifications

```
[DEFAULT]
...
notification_driver = noop
```

2-) Editer le fichier « /etc/glance/glance-registry.conf » et apporter les modifications suivantes :

Dans la section [database] configurer l'accès à la base de données :

```
[database]
...
connection = mysql://glance:GLANCE_DBPASS@controller/glance
```

Remplacer GLANCE_DBPASS par un mot de passe convenable

Dans les sections [keystone_authtoken] et [paste_deploy] configurer l'accès au service d'identité :

```
[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
auth_plugin = password
project_domain_id = default
user_domain_id = default
project_name = service
```

```
username = glance

password = GLANCE_PASS

[paste_deploy]

...

flavor = keystone
```

Dans la section [DEFAULT] configure le notification driver de « noop » pour désactiver les nofifications

```
[DEFAULT]

...

notification_driver = noop
```

RESUME

Ce projet de fin d'études, proposé par la société SERVICOM a pour but de mettre en place une plateforme de virtualisation et le déploiement d'une solution Cloud OpenSource. Notre plateforme est composé d'une couche de virtualtisation VMware dans laquelle les différentes machines virtuelles créées et déployées pour garantir la fiabilité et l'efficacité du système d'information et d'une plateforme du Cloud privé IaaS basé sur OpenStack.
Ce rapport regroupe tout le processus de réalisation en particulier conception, implémentation et test.

Mots clés

Cloud, VMware, IaaS, OpenStack

Abstract

This graduation project proposed by SERVICOM aims to implement a new virtualization platform and deploy a Cloud Open source solution. Our proposed platform consists of a VMware virtualization layer. Several virtual machines are created and deployed via this layer in order to insure reliability and information system efficiency. Also, a private Cloud platform IaaS based on OpenStack is a part of the virtualization layer.

This report contains the realization process, in particular, the conception, the implementation and tests.

Keywords

Cloud, VMware, IaaS, OpenStack

14, Rue Yahia Ibn Omar - 1082 Mutuelleville - Tunis
Tél. : +216 71 28 99 81 / +216 71 89 17 31 - Fax : +216 71 89 26 25