

DÉDICACE

*A mes parents*

## REMERCIEMENTS

Ce présent travail est l'aboutissement de la coopération et le soutien de plusieurs personnes. Nous tenons à remercier :

- **Pr JAZET Michel**, notre superviseur, qui a su nous orienter tout au long de cette fin d'étude avec beaucoup de pédagogie ;
- **M. YONOU Fred**, notre encadreur académique, qui nous a soutenu pendant cette expérience avec beaucoup de patience et de pédagogie ;
- **Mme NOUBANKA Manuella**, Directrice de 3IAC, pour son engagement au suivi de notre cursus ;
- **M. GUIMEZAP Paul**, Président Fondateur de l'IUC pour son engagement sans relâche à faire de ses étudiants des professionnels répondants aux attentes du marché de l'emploi ;
- **M. ASSONTSA Robert Charly**, Directeur de ARC Corporate et notre encadreur professionnel, qui nous a accompagné tout au long de cette expérience professionnelle avec beaucoup de professionnalisme et de rigueur ;
- **L'ensemble du personnel et enseignants de 3IAC** pour leur esprit de collaboration et leur gentillesse qui a facilité notre travail ;
- **Les membres du jury** qui nous font l'honneur d'évaluer notre travail ;
- **Tous nos camarades** pour leur solidarité à notre égard ;

Je ne peux terminer sans mentionner mes remerciements, les plus dévoués, à tous ceux ou toutes celles qui, de près ou de loin m'ont également aidé dans ce travail.

## SOMMAIRE

<b>Dédicace</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Sommaire</b>	<b>v</b>
<b>Glossaire</b>	<b>vi</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Résumé</b>	<b>xii</b>
<b>Abstract</b>	<b>xiii</b>
<b>Introduction Générale</b>	<b>1</b>
 <b>Partie I ETAT DE L'ART</b>	 <b>3</b>
 <b>Chapitre 1 GESTION DES DONNÉES DANS UNE ENTREPRISE</b>	 <b>4</b>
1.1 La donnée dans un monde numérique. [19] . . . . .	4
1.2 Étude de l'existant . . . . .	8
1.3 Critique de l'existant / Problématique . . . . .	9
 <b>Chapitre 2 LE CLOUD COMPUTING - GESTION DE PROJET</b>	 <b>11</b>

2.1	Généralités sur le cloud computing. . . . .	11
2.2	Les aspects de sécurité du cloud. . . . .	25
2.3	Gestion d'un projet de cloud computing. . . . .	31
2.4	Solutions du Cloud existante. . . . .	37
2.5	Critique générale. . . . .	40
2.6	Les scanners des vulnérabilités.[23] . . . . .	42
2.7	Choix des outils. . . . .	44

## **Partie II ANALYSE MODELISATION ET MISE EN PLACE DU PRO- JET** **52**

### **Chapitre 3 PRÉSENTATION DE LA SOLUTION ET ANALYSE** **53**

3.1	Cahier de charges . . . . .	53
3.2	Branche Fonctionnelle : Analyse et Spécification des Besoins . . . . .	55
3.3	Branche Techniques : Environnement Matériel et Logiciel . . . . .	69
3.4	Architecture Réseau.[18] . . . . .	72
3.5	Diagramme de déploiement du système. . . . .	73
3.6	Environnement Logiciel. [15] . . . . .	74
3.7	Architecture de Solution Openstack.[13] . . . . .	76
3.8	Diagramme de Déploiement . . . . .	78

### **Chapitre 4 IMPLEMENTATION – TEST - BILAN** **87**

4.1	Implémentation. . . . .	87
4.2	Tests . . . . .	91
4.3	Bilan . . . . .	96

### **Conclusion Générale** **97**

### **Bibliographie** **A**

### **Chapitre A** **AA**

### **Chapitre B** **BA**

B.1	Les scanners des vulnérabilités . . . . .	BA
-----	---	----

**Table des matières**

**BB**

## LISTE DES SIGLES ET ABRÉVIATIONS.

- **API** Applications Programming Interface
- **CPU** Central Processing Unit
- **DAS** Direct Attached Storage
- **DELL** Development of Early Language Learning
- **ESXI** Elastic Sky X Integrated
- **FC** Fiber Channel
- **HP** Hewlett Packard
- **IT** Information Technologies
- **IP** Internet Protocol
- **ISCSI** Internet Small Computer System Interface
- **IaaS** Infrastructure as a Service
- **KVM** Kernel-based Virtual Machine
- **MYSQL** Système de gestion de base de données
- **NAS** Network Attached Storage
- **NFS** Network File System
- **NIST** National Institute of Standards and Technology

- **TPE** Très Petite Entreprises
- **PaaS** Plateform as a Service
- **PME** Petites et Moyennes Entreprises
- **RAM** Random Access Memory
- **RAID** Redundant Array of Independent Disks
- **SaaS** Software as a Service
- **SAN** Storage Area Network
- **SI** : Système d'Information
- **vCPU** virtual Central Processing Unit
- **VM** Virtual Machine
- **VNC** Virtual Network Computing
- **VPN** Virtual Private Network
- **UML** : Unified Model Language
- **XML** : Extensible Markup Language

## LISTE DES TABLEAUX

2.1	Avantages et inconvénients des services . . . . .	19
2.2	Caractéristiques de la solution voulue . . . . .	31
2.3	Comparaison entre les solutions Cloud . . . . .	40
2.5	Comparaison entre les méthodes de modélisation . . . . .	42
2.6	Les composants de Nova . . . . .	47
3.1	Phase du Projet . . . . .	54
3.2	Estimation financière des ressources matériels . . . . .	55
3.3	Configuration logiciel . . . . .	75



## LISTE DES FIGURES

1.1	La pyramide des savoirs et la gestion de la connaissance. . . . .	5
1.2	Panorama de données structurées et non-structurées. . . . .	6
2.1	Les différents niveaux des services du Cloud Computing. . . . .	17
2.2	Comparatif des technologies impliquées selon le modèle de service souscrit. . . . .	17
2.3	Les modèles de déploiement de cloud. . . . .	21
2.4	Répartition du marché des applications cloud dans le monde en 2015 et en 2020. . . . .	22
2.5	Architecture de cloud brokering. . . . .	23
2.6	Revenu de l'année 2017 en millions de dollars US des principaux fournisseurs cloud mondiaux. . . . .	23
2.7	Cycle de vie de la sécurité des données dans le cloud. . . . .	30
2.8	Modèle du cycle de vie en cascade . . . . .	33
2.9	Modèle de cycle de vie en V . . . . .	34
2.10	Modèle de cycle de vie en spirale . . . . .	35
2.11	Méthode agile . . . . .	36
2.12	Méthodologie 2TUP . . . . .	37
2.13	Pourcentage d'utilisation d'OpenStack . . . . .	45
2.14	Le rôle d'OpenStack . . . . .	46
2.15	Ecosystème d'images d'OpenStack . . . . .	46
2.16	L'architecture Nova. . . . .	47
2.17	L'Architecture de Swift. . . . .	49
2.18	L'Architecture d'OpenStack. . . . .	50

3.1	Diagramme de Gantt . . . . .	55
3.2	Diagramme des cas d'utilisation Générale . . . . .	59
3.3	Cas d'utilisation « Consulter l'état du nuage » . . . . .	60
3.4	Cas d'utilisation « Gérer les instances» . . . . .	61
3.5	Cas d'utilisation « Gérer les services » . . . . .	62
3.6	Cas d'utilisation « Gérer les Flavours » . . . . .	63
3.7	Cas d'utilisation « Gérer les Images » . . . . .	64
3.8	Cas d'utilisation « Gérer les Projets » . . . . .	65
3.9	Cas d'utilisation « Gérer les Utilisateurs » . . . . .	65
3.10	Cas d'utilisation « Membre d'un projet » . . . . .	66
3.11	Diagramme d'activité globale . . . . .	68
3.12	Diagramme d'activité « créer une instance » . . . . .	69
3.13	Architecture-SAN "Storage Area Network" . . . . .	70
3.14	Exemple d'un châssis . . . . .	71
3.15	Représentation d'un serveur blade . . . . .	72
3.16	Architecture réseau . . . . .	73
3.17	Diagramme de déploiement du système . . . . .	74
3.18	Architecture Générale OpenStack . . . . .	77
3.19	Diagramme de déploiement OpenStack . . . . .	78
3.20	Diagramme de séquences globale . . . . .	79
3.21	Scénario d'authentification . . . . .	80
3.22	Diagramme de séquences « créer un projet » . . . . .	81
3.23	Diagramme de séquences « Créer une instance » . . . . .	83
3.24	Diagramme de séquences d'entités globales . . . . .	84
3.25	Diagramme de séquences « authentification » . . . . .	84
3.26	Diagramme de séquences entité « Créer un projet » . . . . .	85
3.27	Diagramme de séquences entité « créer une Instance » . . . . .	86
4.1	Liste des Composants d'OpenStack . . . . .	88
4.2	Dashboard d'authentification au nuage . . . . .	92
4.3	Vue d'ensemble de Nuage . . . . .	92
4.4	Création d'un Projet « Tenant » . . . . .	93
4.5	Création d'un utilisateur . . . . .	93

4.6	Création d'un Image . . . . .	94
4.7	Création d'un groupe de sécurité . . . . .	95
4.8	Création d'une instance « machine virtuelle » . . . . .	95
4.9	La rédaction des règles d'un groupe . . . . .	96
4.10	L'interface de création le paire de clés. . . . .	96

## RÉSUMÉ

Le présent document présente le processus de gestion de la disponibilité et de la sécurisation des données dans un système Cloud Computing afin d'offrir des services IAAS dans une architecture distribuée et supervisée. Cette solution est conçue pour l'entreprise ARC Corporate afin d'améliorer le rendement, les performances, la disponibilité et la sécurité des serveurs sur les quels ses systèmes et applications sont déployés en entreprise . Car en effet l'achat d'ordinateurs puissants et le temps d'installation et de maintenance de l'environnement de travail coûtent énormément d'argent. Cette plateforme nous permettra donc de servir des machines virtuelles personnalisées et robustes, cela fera office de serveurs où seront déployées différentes applications de l'entreprise.

Nous avons fait une étude sur différentes solutions open source et propriétaires du Cloud Computing tout en précisant les techniques de virtualisation utilisées pour chacune d'entre elles. Ceci nous a permis d'avoir une idée riche sur les techniques de virtualisation ainsi que les différentes solutions disponibles de Cloud Computing et de maîtriser son concept. Suite à notre étude, OpenStack a été la solution adéquate du fait de sa sécurité, flexibilité, modularité et extensibilité. Cette solution sera implémentée dans un environnement virtuel afin de procéder aux différents tests à court terme. En se basant sur l'étude réalisée, nous pourrons mettre en place un environnement Cloud si les contraintes matérielles sont relaxées.

## ABSTRACT

This document describes the process of managing data availability and security in a cloud computing system to deliver IAAS services in a distributed and supervised architecture. This solution is designed for ARC Corporate to improve the performance, performance, availability and security of servers on which its systems and applications are deployed in the enterprise. Indeed, the purchase of powerful computers and the time required to install and maintain the working environment cost a lot of money. This platform will allow us to serve customized and robust virtual machines, it will act as servers where different applications of the company will be deployed.

We did a study on different open source and proprietary cloud computing solutions while specifying the virtualization techniques used for each of them. This allowed us to have a rich idea on virtualization techniques as well as the different cloud computing solutions available and to master its concept. As a result of our study, OpenStack was the right solution because of its security, flexibility, modularity and scalability. This solution will be implemented in a virtual environment in order to carry out the various short-term tests. Based on the study, we will be able to set up a Cloud environment if hardware constraints are relaxed.

## INTRODUCTION GÉNÉRALE

Face à l'augmentation continue des coûts de mise en place et de maintenance des systèmes d'informations, les entreprises externalisent de plus en plus leurs services informatiques en les confiant à des entreprises spécialisées comme les fournisseurs de Cloud. L'intérêt principal de cette stratégie pour les entreprises réside dans le fait qu'elles ne paient que pour les services effectivement consommés.

Le Cloud Computing est aujourd'hui le sujet phare dans le domaine des systèmes d'information et de communication. Après la virtualisation, il paraît être la révélation qui va permettre aux entreprises d'être plus performantes et de gérer le coût des systèmes d'informations plus sereinement. Mais suite à cette entrée fracassante nous pouvons tout de même nous demander comment sécuriser et rendre disponibles les informations dans un système de Cloud Computing ? C'est pour cela que ce travail de fin d'études d'ingénieur s'intéresse à ce domaine tout nouveau, du moins pour nous.

Le terme Cloud Computing, ou informatique dans les nuages, est un nouveau modèle informatique qui consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui. Cette nouvelle technologie permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosses quantités d'informations.

L'objectif de ce projet est de garantir une exploitation du système d'information plus souple, flexible, disponible et sécurisé en accord avec les besoins métiers à tout instant. Et justement d'approfondir et d'expérimenter nos connaissances sur le Cloud Computing et ses aspects de sécurité, puis de faire son état de l'art, en vue de choisir la meilleure solution disponible à l'heure actuelle, de la déployer et l'évaluer. Pour ce faire nous avons déployé un Cloud privée de type infrastructure en tant que service.

Ainsi, le présent manuscrit s'articule autour de quatre chapitres :

- Le premier chapitre nous présentons la gestion de données dans une entreprise, faire l'étude de l'existant puis ressortir la problématique ;
- Le deuxième chapitre nous donnons quelques définitions et généralités sur le Cloud, ses aspects de sécurités, la gestion d'un projet de Cloud Computing, la description des différentes solutions existantes, la présentation de la solution et enfin ;
- Le troisième chapitre nous ressortons le cahier de charge, les spécifications fonctionnelles et techniques puis les différents diagrammes ;
- Le quatrième chapitre détaille les différentes phases d'implémentation, de test de déploiement et de bilan de la solution.

# **Première partie**

## **ETAT DE L'ART**



# CHAPITRE 1

## GESTION DES DONNÉES DANS UNE ENTREPRISE

Ce chapitre permet de mettre ce projet dans son cadre général. Il comporte deux parties : la première porte sur l'entreprise d'accueil alors que la deuxième décrit la problématique.

### 1.1 La donnée dans un monde numérique. [19]

Selon l'encyclopédie en ligne Wikipédia, une donnée [1] est « une description élémentaire d'une réalité ». Par exemple, un nom de famille, une date de naissance, un poids, une taille, une nationalité sont des données.

De façon générale, une donnée peut être caractérisée de la manière suivante :

- **La donnée quantitative** « qui peut être mesurée ou repérée » (Définition du livre d'Albert Monjallon, Introduction à la méthode statistique, Librairie Vuibert Paris 1963). Il s'agit d'une donnée qui se réfère aux chiffres ;
- **La donnée qualitative** auquel « *on ne peut pas attribuer une valeur ou une caractéristique* » (Définition provenant de l'encyclopédie en ligne Wikipédia). Par exemple, il peut s'agir de la description d'un objet (texture, aspect).

En rassemblant des données qui ont un sens, cela permet d'en déduire une information. Et le résultat de l'analyse après collecte des informations permet de constituer une connaissance. L'obtention de savoirs s'avère primordiale pour une entreprise ou une organisation, afin que les dirigeants puissent prendre les meilleures décisions possibles. Ainsi, la connaissance devenant intelligence.



FIGURE 1.1 – La pyramide des savoirs et la gestion de la connaissance.

De nos jours, la collecte et le traitement de ces données s’effectuent numériquement aux moyens d’applications informatiques. Au sens IT, une donnée digitale est un élément compréhensible et traitable par des équipements informatiques. Trois formes de données peuvent être définissables :

- **La donnée structurée** : il s’agit d’une donnée qui est organisée suivant une structure. Elle peut être générée par un humain ou une machine, cependant sa qualité première est qu’elle puisse être facilement classée, retrouvée et extraite ;
- **La donnée non-structurée** : il s’agit d’une donnée qui est organisée sans la moindre structure. Elle peut être également générée par un humain ou une machine ;
- **La donnée semi-structurée** : il s’agit d’une donnée dont certains éléments de celle-ci sont organisés suivant une structure.

Pour donner quelques exemples, une base de données relationnelles est une donnée structurée, alors qu’un document Word, un fichier PDF, un fichier audio, un fichier image au format JPEG ou encore une vidéo sont des données non-structurées. Les fichiers XML et JSON, souvent utilisés par des applications informatiques sont qualifiés d’informations semi structurées.

Cependant, il convient de souligner qu’en réalité, les limites entre ces trois catégories sont floues : en effet, un document Word sera considéré comme un ensemble de données non structurées, mais qui peut comporter des méta-données, qui sont des données semi-structurées.



FIGURE 1.2 – Panorama de données structurées et non-structurées.

### 1.1.1 Le stockage des données numériques.

#### 1.1.1.1 Les technologies.

Les technologies de stockage des données ont été scindées en deux grandes parties : le stockage de masse, dont l'objectif est de stocker une grande quantité d'information à long terme, et le stockage à accès rapide, souvent utilisé pour le traitement interne des informations dans les ordinateurs. Concernant les solutions de stockage de masse, elles ont été regroupées en cinq catégories :

- **Les supports physiques** (cartes et rubans perforées) ;
- **Les supports magnétiques** (bandes, disquettes, disques durs, cassettes DLT) ;
- **Les supports optiques** (CD, DVD, Blue-Ray) ;
- **Les supports à semi-conducteur** (clés USB, cartes SD et micro SD, disques SSD) ;
- **Les supports en ligne** (cloud).

Concernant les solutions de stockage à accès rapide, elles peuvent être scindées en deux grandes familles :

- **Les mémoires vives**, stockant les données de façon temporaire (RAM, SDRAM) ;
- **Les mémoires mortes**, qui peuvent conserver les informations de façon permanente (ROM, PROM, EPROM, EEPROM).

Pour la suite de cette section, nous présenterons les moyens actuels pour la conservation des données numériques.

### 1.1.1.2 Les principaux modes de stockage.

**Le disque dur magnétique (Hard Disk Drive – HDD) :** est encore utilisé de nos jours. Il peut être embarqué directement au sein d'un ordinateur (disque dur interne), ou bien être intégré dans un boîtier qui se relie à l'ordinateur via un câble USB (disque dur externe). Les capacités de stockage d'un disque dur peuvent aller à plus de 1 To dans certains cas.

**Le disque dur Solid State Drive (SSD) :** Il commence à être proposé de base sur les ordinateurs actuels, notamment pour ses qualités de rapidité d'accès à la donnée et mais aussi sur le fait qu'il offre des capacités de stockage équivalentes à celle du disque dur magnétique. Il est également disponible en version disque dur externe.

**La clé USB et les cartes mémoires :** un volume de données non négligeable (de 128 Mo à plusieurs Go d'enregistrement). Leurs avantages sont le branchement à chaud sur son équipement informatique ou encore l'aisance pour transporter ces supports.

**Le serveur de stockage réseau Network Attached Storage (NAS) :** la forme d'un boîtier qui fonctionnait initialement avec des outils logiciels (système d'exploitation de stockage) adaptés avec des outils matériels (disque dur). L'intérêt de ce support étant le stockage centralisé et le partage de données au travers d'un réseau, le plus souvent en local (Local Area Network – LAN), en s'appuyant sur les protocoles standards tels qu'Ethernet et Internet Protocol (IP).

Les fruits du travail sur la virtualisation ainsi que le stockage NAS ont permis de faire émerger le concept de réseau de stockage Storage Area Network (SAN), où les informations sont stockées dans des baies de stockage et partagées au travers d'un réseau par le biais de protocoles dédiés aux systèmes de stockage (Fibre Channel ou iSCSI). À l'instar du NAS, le SAN fonctionnait initialement avec des outils logiciels adaptés avec des outils matériels.

Les réflexions autour du stockage à définition logicielle Software Defined Storage (SDS) ont permis notamment de proposer du stockage NAS ou SAN avec des dispositifs de stockage non couplés au système d'exploitation de stockage (par exemple, l'outil logiciel FreeNAS de IxSystems). Le SAN et le NAS sont les principaux modes de stockage des données utilisés chez les fournisseurs cloud.

## 1.1.2 Les principaux défis pour la gestion des données numériques.

Un premier challenge en termes de gestion des données numériques consistera en premier lieu de disposer de moyens et d'outils permettant l'exploitation de qualité d'énormes volumes de données générées, provenant de sources diverses (ordinateurs, smartphones, tablettes, objets connectés, services

cloud, etc.). Ces moyens s'appuieront principalement sur des techniques provenant de l'Intelligence Artificielle et/ou de Big Data.

Un deuxième axe majeur concerne la sécurité des données, et notamment celles non-structurées. En effet, l'évolution des infrastructures réseaux des organisations vers des solutions de type « Borderless Networks » (réseaux sans frontières) a contribué à rendre ces informations de plus en plus difficilement localisables, et a compliqué aussi de fait de connaître les personnes qui en ont l'accès et l'usage.

En effet, sans une visibilité et un contrôle de ce qui est fait, de comment cela est fait et surtout par qui cela est fait, il devient délicat d'assurer une sécurité des données efficiente. Ainsi, les entreprises et les organisations se doivent de trouver des réponses à ces questions (non exhaustives) :

- Comment les personnes de l'organisation peuvent accéder aux données de l'organisation présentes dans le cloud, sans compromettre la sécurité de celle-ci ?
- Comment rendre les données de l'organisation incompréhensibles aux personnes qui n'y sont pas autorisées, notamment si elles sont situées dans le cloud ?
- Comment faire en sorte qu'une organisation autorisant ses employés à utiliser leur propre matériel informatique (concept du BYOD) puisse accéder aux données et applications de celle-ci sans réduire la sécurité ?
- Quels peuvent être les moyens qu'une organisation peut mettre en œuvre pour limiter le risque de fuite des données (notion du Data Leak Prevention – DLP) ?

Dans tous les cas, avant de mettre en place des solutions pour la protection des données, l'organisation devra mener au préalable une analyse des risques auprès des métiers, afin de savoir quelles sont les données à protéger (classification de l'information) et qui peuvent être utilisées et/ou stockées dans le cloud.

Mais avant de présenter ces solutions, nous allons nous intéresser au concept du cloud-computing.

## **1.2 Étude de l'existant**

ARC Corporate a depuis longtemps opter pour le déploiement de ses systèmes en local. En effet, elle dispose d'un réseau d'entreprises chez une grande partie du territoire. Tous ces sites ne sont pas reliés, qui constitue un manque de centralisations des données. En effet ARC Corporate est en cour

d'étudier un projet qui sert centraliser et à dupliquer toutes les données critique vers un autre site distant grâce à un services de cloud pour assurer la continuité de service en cas d'un désastre, car la perte d'un serveur signifie la perte de toutes les données. De ce fait, la majeure partie d'administration et de sécurisation du réseau se trouvera au niveau du site à déployer. Pour des raisons de confidentialité, nous ne pouvons détailler l'architecture de ce réseau et plus précisément du DataCenter. Nous notons juste qu'il sera constitué d'un grand nombre d'équipements hétérogènes tels que :

- Des switch et des routeurs.
- Une solution antivirus Symantec End Point.
- Des serveurs Web Apache et Microsoft IIS et des serveurs des bases de données ORACLE et MS-SQL Server.
- Des serveurs applicatifs Windows 2012 R2, Windows 2008, Ubuntu 16,...
- Des solutions de virtualisation classique tel que VMware et Oracle VM Server.
- Plusieurs Consoles d'administrations.

### **1.3 Critique de l'existant / Problématique**

D'après ce qu'on a précédemment, ARC Corporate dispose d'une multitude de serveurs qui facilite la gestion des services au sein de l'entreprise en local. Tous ces serveurs installés en local n'ont pas une administration centralisée. Bien qu'elle soit une société de services, ARC Corporate dispose tout aussi de nombreux services qui n'ont pas trait aux études.

- Inexistence des plateformes de travail collaboratif.
- Une perte du temps et augmentation de cout de maintenance des outils ;
- Perte de l'espace chaque serveur travail avec 30 pourcent de sa capacité ;
- Perte d'électricité et énorme environnement de travail non utilisé ;
- Gaspillage de l'espace de Stockage et manque des statistiques à jour ;
- Moyen de monitoring faible ;
- Le nombre d'effectifs actuel ne peut plus gérer les ressources infrastructures ;

Sans oublier la lourdeur des procédures administratives, qui en piètre très souvent sur la qualité du service rendu à la fin.

L'objectif sera de constituer une mémoire organisationnelle de l'entreprise qui prend en compte les acteurs et les contextes en procédant à la disponibilité et la sécurisation des données des serveurs (système de gestion, pharmacie, téléphonie, ...)

En somme, l'entreprise nous a fait comprendre qu'elle avait commencé à travailler dans ce sens et réfléchissait sur la technologie à déployer, qui pourrait être fiable, sécuriser et moins coûteuse, permettant d'assurer la pérennité, la fiabilité, la sécurité, la disponibilité et la traçabilité des données.

Il revenait donc à nous d'analyser le projet et lui donner une orientation plus appropriée à leurs besoins.

## CHAPITRE 2

# LE CLOUD COMPUTING - GESTION DE PROJET

Ce chapitre est celui dans lequel nous allons présenter les notions fondamentales du Cloud Computing, ses enjeux, ses évolutions et son utilité ainsi que la technologie qui la constitue et les différents acteurs du secteur. puis une critique générale et enfin nous présentons de manière synthétique les outils technologiques et les méthodes qui ont contribué à produire ce travail.

### 2.1 Généralités sur le cloud computing.

#### 2.1.1 Les origines.[5]

Les premières réflexions ayant permis l'émergence du cloud computing tel que nous le connaissons aujourd'hui dataient de 1961, quand le Professeur étasunien John McCarthy, qui était également en charge des travaux sur le développement d'un système informatique pouvant être utilisé par plusieurs utilisateurs en même temps (Compatible Time-Sharing System), a imaginé le concept de l'informatique à la demande (computing utility), lors d'un discours public pour la célébration du centenaire de l'Institut de recherche du Massachusetts Institute of Technology (MIT), dans lequel « l'informatique [sera] organisée comme un service public, tout comme le téléphone est un service public »

Les années 1990 ont vu le développement de l'Internet et des technologies du Web, ce qui a permis l'arrivée de premières solutions s'appuyant sur le concept du computing utility, appelées Application Service Provider (ASP) pouvant être considérées comme une première forme de services



cloud. Il s'agissait de proposer, au grand public et/ou aux entreprises, des applications fonctionnant sur des serveurs du fournisseur de services et qui sont accessibles par le réseau Internet via un navigateur web. Nous pouvons citer par exemple Yahoo et son moteur de recherche (1995), Hotmail et ses services de messagerie électronique (1996, racheté en 1997 par Microsoft) ou encore Salesforce, un éditeur de logiciel faisant partie des pionniers dans la fourniture de services en ligne pour les entreprises (1999).

Mais le concept du cloud computing a véritablement vu le jour au début des années 2000, sous l'impulsion du groupe américain Amazon, l'un des leaders dans les secteurs du commerce électronique.

En d'autres termes, le cloud computing est l'utilisation, sans posséder quoi que ce soit d'autre qu'une connectivité à Internet, de services proposés ou de moyens partagés permettant d'effectuer des traitements informatiques divers en fonction de ses besoins, de façon rapide et le plus automatisé possible.

À noter que les notions de sécurité, de disponibilité et de « quelque part » utilisées dans les définitions précédentes seront évoquées dans le cadre de ce mémoire, principalement en matière juridique, de protection des données ou encore en termes de confiance dans le fournisseur cloud.

## **2.1.2 Bénéfices du cloud Computing.[4]**

Les retombées des principes du cloud sont bénéfiques à la fois pour son fournisseur, les entreprises délocalisant leurs infrastructures. Généralement, ils assurent aux deux premiers une meilleure rentabilité. De plus, ils permettent à l'entreprise de se concentrer sur les tâches de production autres que la maintenance de systèmes informatiques.

### **2.1.2.1 Pour le fournisseur**

Les bénéfices du fournisseur sont uniquement dus au fait de la mutualisation des ressources. En effet, après son investissement dans la mise en place des infrastructures pour le cloud, il fait payer aux entreprises la marge nécessaire pour sa rentabilisation. Comme pour une entreprise disposant d'une plateforme interne, il paie pour les frais d'administration de l'ensemble. Cette dépense peut être amortie par facturation aux entreprises. En plus de cette marge, il bénéficie des coûts de réutilisation des ressources. En effet, compte tenu de la non appartenance des ressources aux entreprises, elles (les ressources) leurs sont facturées à chaque usage. La même ressource peut ainsi faire l'objet de plusieurs facturations

### 2.1.2.2 Pour l'entreprise

C'est elle la première gagnante de cette technologie. Elle réalise des bénéfices en argent et en flexibilité dans sa capacité à s'agrandir.

1. **La réduction des coûts :** Le recours au cloud permet à l'entreprise d'être facturée à l'usage, en fonction de ses besoins. Pour avoir une idée du gain réalisé, reprenons cette observation de Michael Crandell du groupe RightScale à propos du cloud d'Amazon « Le cout à pleine charge d'un serveur sur Amazon se situe entre 70USD et 150USD par mois alors qu'il s'élève à 400USD en moyenne par mois s'il était hébergé par l'entreprise en interne ». Plusieurs raisons expliquent cette différence de cout. En effet, une gestion interne de l'infrastructure implique l'achat des matériels, l'affectation du personnel (et donc du cout salarial qu'il induit) pour la gestion de l'infrastructure et divers moyens de production mis en place pour le fonctionnement de l'ensemble (électricité, locaux, ...etc.). Le partage de ressources tel que pratiqué dans le cloud permet au fournisseur de répartir ces couts entre plusieurs entreprises.
2. **La réduction des gaspillages :** Les infrastructures gérées en interne sont souvent sous-utilisées, alors que l'infrastructure d'un cloud mutualise l'ensemble de ressources pour un grand nombre d'entreprises. La mutualisation consiste à mettre à la disposition de plusieurs utilisateurs une base commune de ressources. Elle permet ainsi d'augmenter le taux d'utilisation de ces ressources. En effet, les ressources n'étant pas dédiées à un seul utilisateur, elles pourront servir à d'autres en cas de besoin.
3. **La flexibilité et accès aux ressources à larges échelle :** L'entreprise peut augmenter la capacité de son infrastructure sans investissement majeur. En effet, grâce à l'allocation dynamique (à la demande) des ressources qu'offre le cloud, il suffit de souscrire à des nouvelles ressources et celles-ci sont directement allouées.  
De plus, l'entreprise est libre de ses allées et venues car les contrats d'utilisation sont limités dans le temps (autour de l'heure).  
Ainsi, l'entreprise peut augmenter ou réduire son infrastructure à sa guise à moindre cout et dans un délai réduit (il faut mettre en avant le critère de rapidité qui est un grand avantage) .  
Rappelons que le cloud offre ainsi à l'entreprise une possibilité d'accéder à une quantité de ressources dont elle ne pourrait se l'offrir en interne. Elle peut dorénavant envisager des applications large échelle sans se soucier de l'obtention des équipements.

### 2.1.3 Les principales caractéristiques.[4]

Le NIST a défini cinq caractéristiques majeures pour le cloud computing :

- **Libre-service et à la demande (On-demand self-service)** : les ressources du fournisseur cloud peuvent être demandées par un client à tout moment et en fonction de ses besoins et sans aucune intervention de la part du fournisseur ;
- **Accès réseau universel (Broad network access)** : l'accès par un client aux ressources du fournisseur cloud s'effectuent par le biais des protocoles réseaux standards (par exemple TCP/IP pour Internet) et à partir de divers matériels informatiques (ordinateurs, tablettes, téléphones mobiles) ;
- **Mutualisation de ressources et multi-location (Resource pooling and multi-tenancy)** : les ressources du fournisseur cloud sont mises en commun avec l'ensemble des clients qui y ont accès ;
- **Élasticité rapide (Rapid elasticity)** : les ressources du fournisseur cloud sont allouées dynamiquement en fonction de la demande (et sont libérées lorsque les besoins des clients sont moindres) ;
- **Service mesurable (Measured service)** : la consommation de chaque ressource cloud est contrôlée et affichée en temps réel.

### 2.1.4 Éléments constitutifs du Cloud Computing.[6]

#### 2.1.4.1 La virtualisation

La virtualisation se définit comme l'ensemble des techniques matérielles et/ou logiciels qui permettent de faire fonctionner sur une seule machine, plusieurs systèmes d'exploitation (appelées machines virtuelles (VM), ou encore OS invitée).

La virtualisation des serveurs permet une plus grande modularité dans la répartition des charges et la reconfiguration des serveurs en cas d'évolution ou de défaillance momentanée.

Les intérêts de la virtualisation sont multiples, on peut citer :

- L'utilisation optimale des ressources d'un parc de machines (répartition des machines virtuelles sur les machines physiques en fonction des charges respectives).

- L'économie sur le matériel (consommation électrique, entretien physique, surveillance).
- L'installation, tests, développements sans endommager le système hôte.

#### **2.1.4.2 Le Datacenter**

Un centre de traitement de données (data center en anglais) est un site physique sur lequel se trouvent regroupés des équipements constituant le système d'information de l'entreprise (mainframes, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.). Il peut être interne et/ou externe à l'entreprise, exploité ou non avec le soutien des prestataires. Il comprend en général un contrôle sur l'environnement (climatisation, système de prévention contre l'incendie, etc.), une alimentation d'urgence et redondante, ainsi qu'une sécurité physique élevée.

Cette infrastructure peut être propre à une entreprise et utilisée par elle seule ou à des fins commerciales. Ainsi, des particuliers ou des entreprises peuvent venir y stocker leurs données suivant des modalités bien définies.

#### **2.1.4.3 La Plateforme collaborative**

Une plate-forme de travail collaboratif est un espace de travail virtuel. C'est un site qui centralise tous les outils liés à la conduite d'un projet et les met à disposition des acteurs. L'objectif du travail collaboratif est de faciliter et d'optimiser la communication entre les individus dans le cadre du travail ou d'une tâche. Les plates-formes collaboratives intègrent généralement les éléments suivants :

- Des outils informatiques.
- Des guides ou méthodes de travail en groupe, pour améliorer la communication, la production, la coordination.
- Un service de messagerie.
- Un système de partage des ressources et des fichiers.
- Des outils de type forum, pages de discussions
- Un trombinoscope, ou annuaire des profils des utilisateurs.
- Des groupes, par projet ou par thématique.
- Un calendrier.

## 2.1.5 Les modèles de service.[5]

### 2.1.5.1 IaaS : Infrastructure as a Service

**L'Infrastructure en tant que Service** mise à disposition par le fournisseur cloud d'une infrastructure avec des capacités de calcul, de serveurs, du stockage et d'une bande passante suffisante. L'avantage de ce modèle pour le client est que cela lui permet de ne pas se préoccuper de l'achat et de la gestion du matériel. Mais il devra gérer toutes ses ressources virtuelles, notamment au niveau de redondance et de sécurité.

En résumé : l'IaaS est le monde des administrateurs informatiques réseau et système ;

**Avantage** : Grande flexibilité, contrôle total des systèmes, qui permet d'installer tout type de logiciel métier.

**Inconvénient** : Besoin d'administrateurs système comme pour les solutions de serveurs classiques sur site.

### 2.1.5.2 PaaS : Platform as a Service.

La Plateforme en tant que Service mise à disposition par le fournisseur cloud d'une plateforme déjà configurée pour permettre au client de déployer les applicatifs métiers souhaités. L'avantage étant que le client n'a pas à se soucier du matériel, ni de la maintenance ou de mise à jour des serveurs virtuels. En revanche, il devra s'assurer totalement de la configuration, de la maintenabilité ou de la montée en charge de ses applications.

En résumé : le PaaS est le monde des développeurs et architectes logiciels, des web designers ;

**Avantage** : Le déploiement est automatisé, pas de logiciel supplémentaire à acheter ou à installer.

**Inconvénient** : Limitation à une ou deux technologies (ex. : Python ou Java pour Google AppEngine, .NET pour Microsoft Azure, propriétaire pour force.com). Pas de contrôle des machines virtuelles sous-jacentes. Convient uniquement aux applications Web.

### 2.1.5.3 SaaS : Software as a Service.

L'Application en tant que Service : mise à disposition par le fournisseur cloud d'une application accessible au client via le réseau Internet. Ainsi, le déploiement, la maintenance, le bon fonctionnement ou encore la gestion des données de l'application sont du ressort du fournisseur. Le service étant disponible en ligne, le client n'a rien à installer sur ses propres équipements informatiques.

En résumé : le SaaS est la partie du cloud qui est visible et accessible par le grand public.

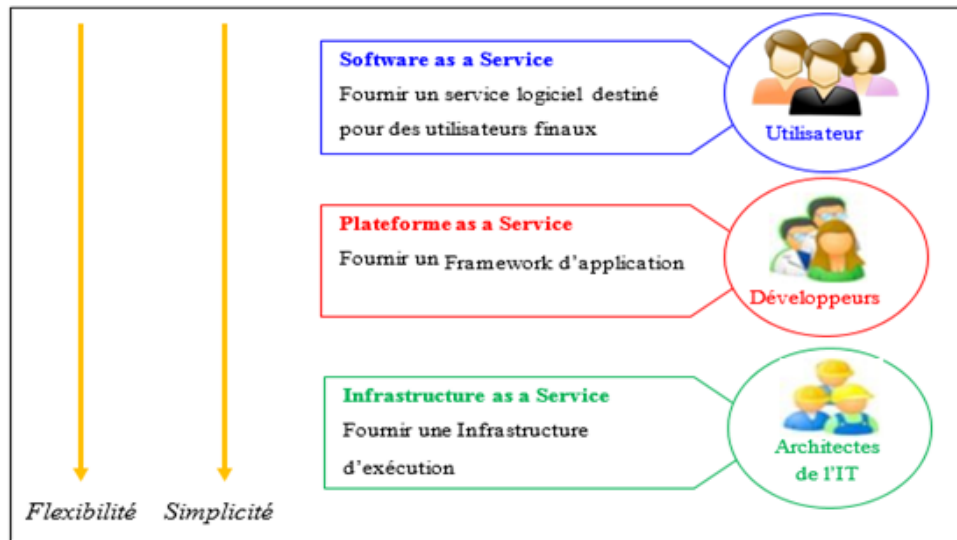


FIGURE 2.1 – Les différents niveaux des services du Cloud Computing.

Par la suite nous avons les différentes technologies qui interviennent dans les différents services.

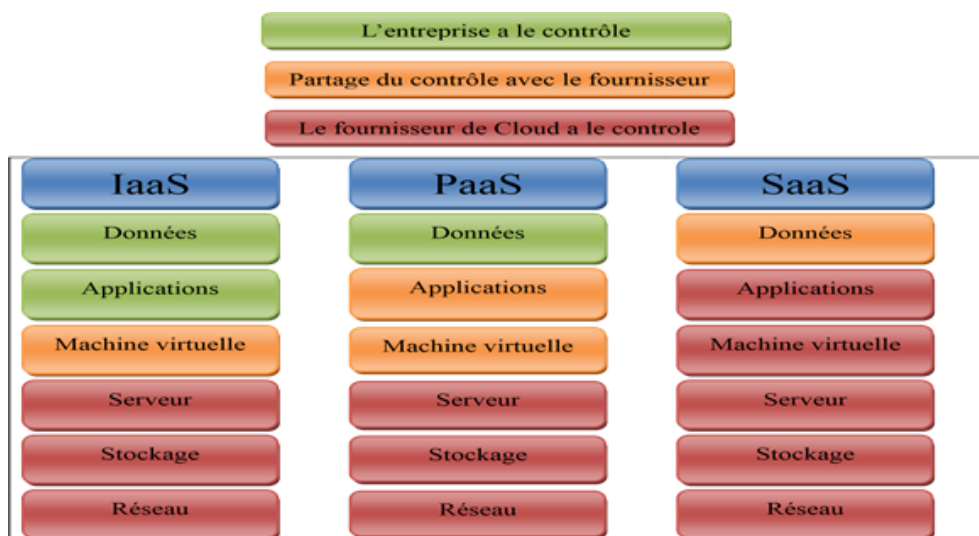


FIGURE 2.2 – Comparatif des technologies impliquées selon le modèle de service souscrit.

D'autres modèles de service existent, cependant ce ne sont que des sous-jacents aux trois principaux modèles, généralement utilisés à des fins marketings ou pour mettre en avant un type particulier de service :

- **Le Tout en tant que Service** (Everything as a Service – XaaS) : il s'agit de l'expression désignant l'ensemble des modèles de service ;

- **La Plateforme d'intégration en tant que Service** (integration Platform as a Service – iPaaS) : service permettant de favoriser les échanges de données et l'interopérabilité entre applications clouds hétérogènes ;
- **Le Bureau en tant que Service** (Desktop as a Service – DaaS) : service de fourniture de postes de travail accessible à distance. À noter que l'administration de ces postes virtuels est à la charge du fournisseur cloud, contrairement à une solution Virtual Desktop Infrastructure (VDI) ;
- **Les Données en tant que Service** (Data as a Service – DaaS) : service d'accès à des sources de données dites « de qualité ». Les traitements spécifiques demandés par les entreprises clientes (en vue de leur business) seront facturés par le fournisseur DaaS ;
- **Les Bases de Données en tant que Service** (DataBase as a Service – DBaaS) : service de fourniture d'un SGBD ;
- **Les Sauvegardes en tant que Service** (Backup as a Service – BaaS) : service permettant la sauvegarde externalisée des données numériques d'une entreprise ou d'une organisation ;
- **Le Plan de Reprise d'Activités en tant que Service** (PRAaaS ou Disaster Recovery as a Service – DRaaS) : service proposant aux entreprises une infrastructure de secours à distance appartenant au fournisseur, en cas de risque majeur de leur site informatique principal ;
- **La Vidéo en tant que Service** (Video as a Service – VaaS) : service délivrant des solutions de visioconférence ;
- **Le Backend Mobile en tant que Service** (Back-end Mobile as a Service – BMaaS) : service offrant aux programmeurs d'applications mobiles des éléments d'infrastructure serveur, afin qu'ils puissent se consacrer pleinement aux développements de leurs applications ;
- **La Gestion des Processus Métiers en tant que Service** (Business Process as a Service – BPaaS) : service proposant entre autres des applications en ligne de gestion des ressources humaines, de la paie, de logistique, de vente ;
- **Le Stockage en tant que Service** (STorage as a Service – STaaS) : service proposant de stocker les données ;
- **La Sécurité en tant que Service** (Security as a Service – SECaaS) : service proposant une ou plusieurs solutions de sécurité à distance. Les produits proposés peuvent comporter du

PRAaaS, du DLP, de la protection des courriels, des outils de chiffrement, des outils de monitoring (SIEM).

#### 2.1.5.4 Avantages et inconvénients des services. [10]

Du point de vue économique, le Cloud Computing est essentiellement une offre commerciale d'abonnement économique à des services externes. Selon le National Institute of Standards and Technology, il existe trois catégories de services qui peuvent être offerts en Cloud Computing : IaaS, PaaS et SaaS.

Les avantages et les inconvénients de ces services se résume dans le tableau ci-dessous.

Tableau 2.1 – Avantages et inconvénients des services

Services	Avantages	Inconvénients
<b>SaaS</b>	<ul style="list-style-type: none"> <li>● Pas d'installation</li> <li>● Plus de licence</li> <li>● Migration</li> <li>● Accessible via un abonnement</li> </ul>	<ul style="list-style-type: none"> <li>● Logiciel limité</li> <li>● Sécurité</li> <li>● Dépendance des prestataires</li> </ul>
<b>PaaS</b>	<ul style="list-style-type: none"> <li>● Pas d'infrastructure nécessaire</li> <li>● Pas d'installation</li> <li>● Environnement hétérogène</li> </ul>	<ul style="list-style-type: none"> <li>● Limitation des langages</li> <li>● Pas de personnalisation dans la configuration des machines virtuelles</li> </ul>
<b>IaaS</b>	<ul style="list-style-type: none"> <li>● Administration</li> <li>● Personnalisation</li> <li>● Flexibilité d'utilisation</li> <li>● Capacité de stockage infinie</li> </ul>	<ul style="list-style-type: none"> <li>● Sécurité</li> <li>● Besoin d'un administrateur système</li> <li>● Demande pour les acteurs du Cloud des investissements très élevés</li> </ul>



### 2.1.6 Les modèles de déploiement.[4]

Le NIST a défini quatre modèles de déploiement :

- **Cloud public (Public cloud)** : les ressources et services cloud sont mises à disposition du grand public par un ou plusieurs fournisseurs (cela peut être une entreprise, un établissement d'enseignement, une organisation gouvernementale ou une combinaison des trois). L'administration de cette infrastructure partagée entre plusieurs clients est entièrement gérée par le(s) fournisseur(s). Cet environnement est hébergé dans les locaux des fournisseurs (ou sous leurs contrôles). À noter que le cloud public a été le premier modèle proposé au début de l'informatique en nuage, et c'est probablement celui qui a le plus d'avenir ;
- **Cloud privé (Private cloud)** : les ressources et services cloud sont fournis à l'usage exclusif d'une seule entreprise ou organisation. À noter que la gestion de cet environnement peut s'effectuer soit par l'organisation elle-même (cloud privé interne) ou soit par un prestataire qu'elle aura choisi (cloud privé externe). De plus, l'infrastructure d'un cloud privé ne sera pas obligatoirement hébergée dans les locaux de l'entreprise ;
- **Cloud communautaire (Community cloud)** : les ressources et services cloud sont fournis à l'usage exclusif d'un groupe d'entreprises ou d'un groupe d'organismes ayant des intérêts communs. À noter que l'administration de cette infrastructure peut s'effectuer soit par l'un ou plusieurs des membres du groupe (cloud communautaire interne) ou soit par un prestataire qu'ils auront choisi (cloud communautaire externe). De plus, l'hébergement d'un cloud communautaire ne sera pas obligatoirement localisé dans les locaux de ses membres ;
- **Cloud hybride (Hybrid cloud)** : les ressources et services cloud sont présents dans un ensemble composé de modèles distincts (en général, il s'agit de la combinaison entre un cloud privé ou communautaire et un cloud public). Le choix de l'emplacement des ressources et services cloud sera à l'appréciation du client, en fonction de ses besoins et des contraintes

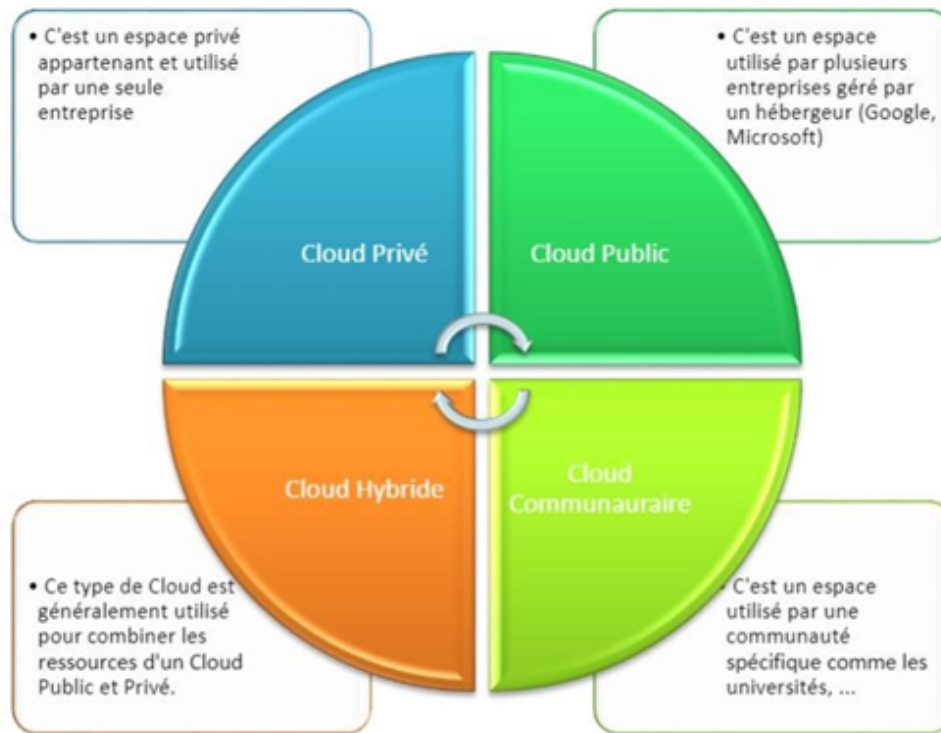


FIGURE 2.3 – Les modèles de déploiement de cloud.

### 2.1.7 Les principales applications.

Voici une liste exhaustive des applications proposées, la plupart étant en mode SaaS :

- **Les outils de gestion de la relation client (CRM)** ;
- Les applications de ressources humaines (gestion du recrutement, de la paie, etc.) ;
- **Les applications financières** (analyse des marchés d'actions, etc.) et **comptables** (gestion de trésorerie, de facturation, etc.) ;
- Les applications de **Business Intelligence (BI)** et de **Progiciels de Gestion Intégrée (ERP)** ;
- **Les services de messagerie** (Hotmail, Yahoo, Gmail), les **outils collaboratifs, bureautiques** (Word, Excel, PowerPoint), de **stockage des données** (Dropbox, OODrive, Box, OneDrive, iCloud, Amazon S3) ;
- **Les applications techniques et scientifiques** (modélisation, simulation, Dessin Assisté par Ordinateur (DAO), Conception Assisté par Ordinateur (CAO), etc.).

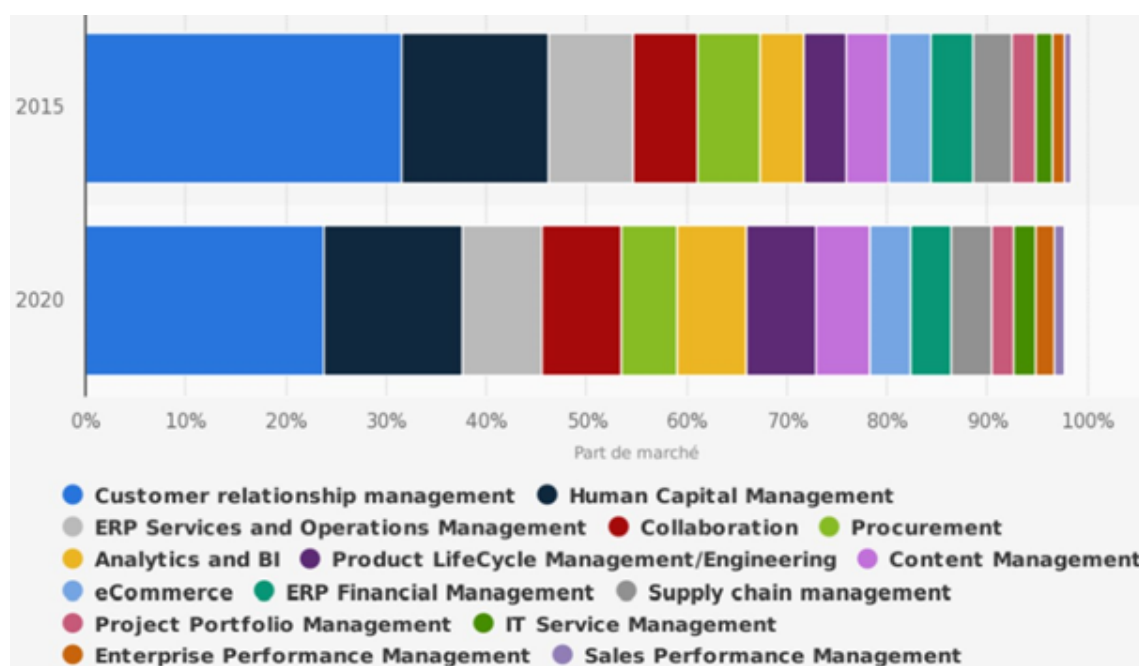


FIGURE 2.4 – Répartition du marché des applications cloud dans le monde en 2015 et en 2020.

### 2.1.8 Les principaux acteurs.

Avant de nous intéresser aux principaux acteurs du marché, il convient de distinguer les types d'entreprises proposant des ressources et des services de cloud computing :

- **Fournisseur de cloud** (cloud provider) : il s'agit d'une société qui propose des services de cloud computing au sens large (IaaS, PaaS, SaaS, etc.) ;
- **Intégrateur de cloud** (cloud builder) : il s'agit d'une société qui va mettre en place une infrastructure cloud à l'aide de solutions déjà existantes ;
- **Fournisseur d'application cloud** (cloud application provider) : il s'agit d'une entreprise centrée uniquement sur le logiciel, en fournissant seulement des solutions en mode SaaS ;
- **Courtier de cloud** (cloud broker) : il s'agit d'une société qui propose une interface en ligne permettant à ses clients d'accéder à son catalogue de services cloud. Il joue un rôle d'intermédiaire entre le client et le fournisseur cloud.

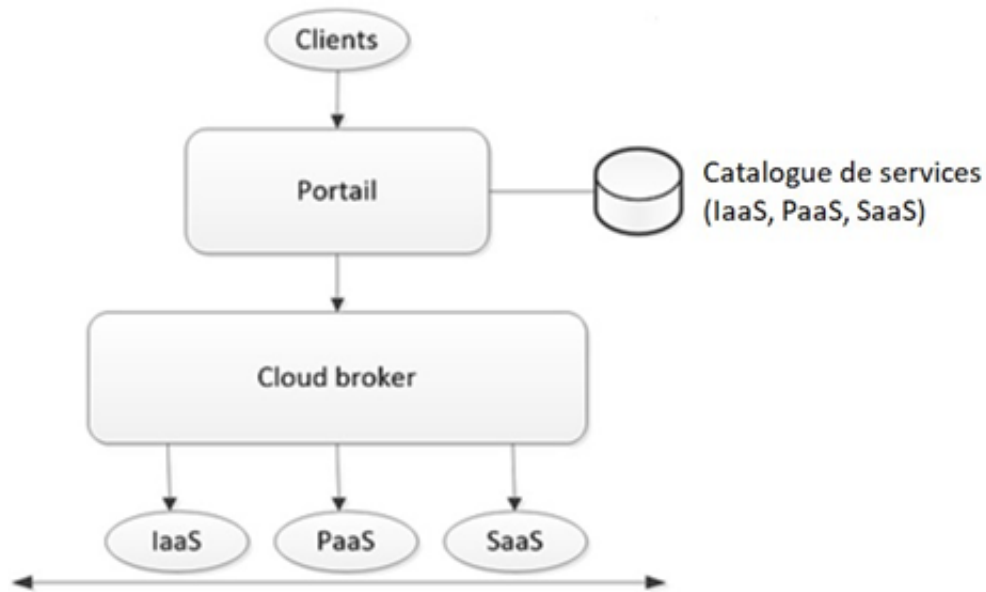


FIGURE 2.5 – Architecture de cloud brokering.

Le graphique ci-dessous nous permet d'en déduire les leaders mondiaux sur le marché du cloud computing. Nous pouvons constater que ce sont en majorité des sociétés américaines, ce qui pourrait poser un problème en termes de souveraineté :

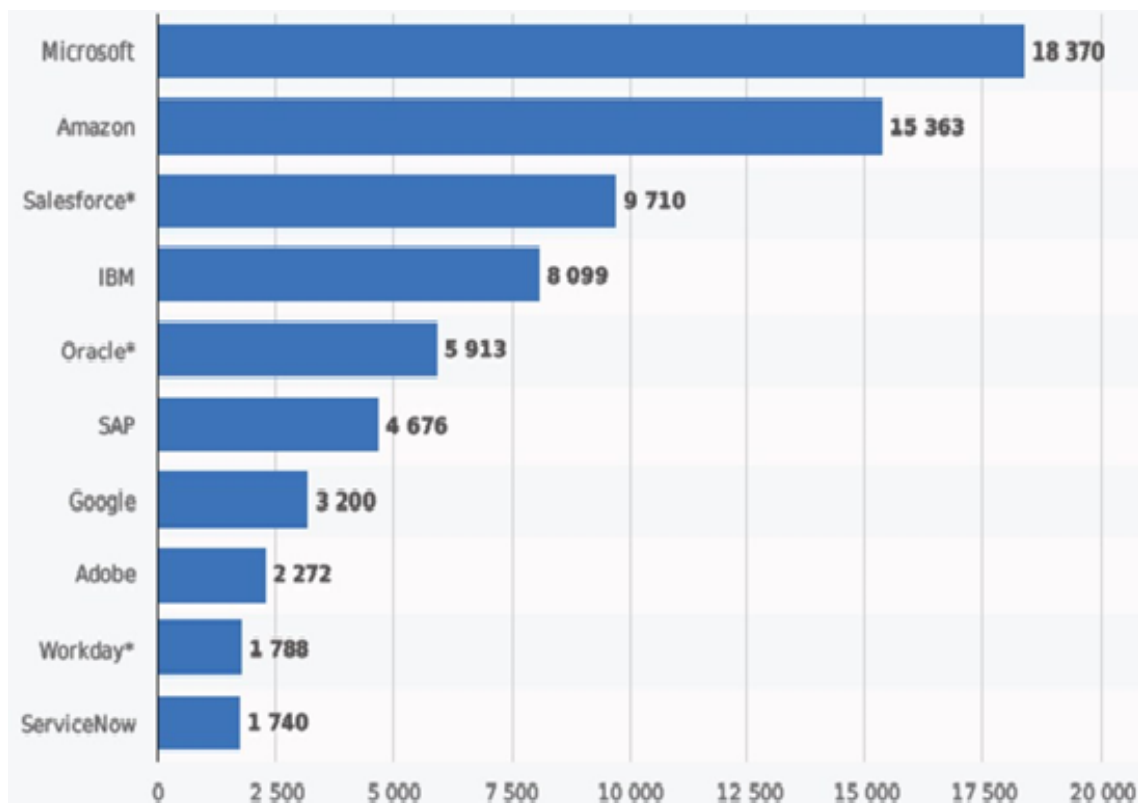


FIGURE 2.6 – Revenu de l'année 2017 en millions de dollars US des principaux fournisseurs cloud mondiaux.

### 2.1.9 Les principaux avantages, limites et contraintes.

Par rapport à une informatique traditionnelle, le cloud computing et ses technologies associées ont apporté de nombreux avantages, dont voici les principaux :

- **La mise à disposition des ressources quasi immédiate** : la réponse à une demande de services cloud (par exemple une demande d'un serveur virtuel) s'effectue avec une rapidité de disponibilité de moins d'une heure maximum et en libre-service ;
- **Les ressources sont accessibles à tout moment** : les infrastructures des fournisseurs cloud sont souvent conçues selon une architecture dite de Haute Disponibilité (High Availability – HA), ce qui permet de proposer des services cloud fiables à des taux de disponibilité très élevés (supérieures à 99
- **Les ressources sont évolutives à tout moment** : les environnements des fournisseurs cloud sont flexibles et peuvent s'adapter en cas de changement d'ordre de grandeur des demandes client (notamment en cas de montée en charge) ;
- **La connaissance des ressources consommées en temps réel** : les clients sont informés en toute transparence sur l'utilisation de leurs ressources cloud ;
- **Le paiement à l'usage (Pay-as-you-go ou pay-as-you-use)** : grâce au libreservice à la demande et au service mesurable, le coût devient proportionnel à l'usage, donc le client paie uniquement pour ce qu'il utilise (facture à la consommation) ;
- **Le transfert des dépenses d'investissement (CAPital EXpenditure – CAPEX) vers des dépenses d'exploitation (OPerating EXpenditure – OPEX)** : pour un client, utiliser un service hébergé chez un fournisseur cloud lui est plus profitable pour ses frais de fonctionnement, car cela lui permet de consommer ce dont il a besoin pendant la durée de ce besoin ;
- **Le recentrage sur le métier** : pour un client, l'administration technique des équipements informatiques d'une organisation (par exemple les serveurs mail) sera effectuée par le prestataire cloud. Ainsi, cela permet de recentrer toutes les ressources et compétences de la Direction des Systèmes d'Information (DSI) sur des applications métiers de l'organisation.

Malgré des avantages indéniables de l'apport du cloud computing, des contraintes et des limites peuvent être énumérées, dont voici les principales :

- **La sécurité** : il s'agit de l'élément majeur pour un fournisseur cloud. Il devra s'assurer que la protection de ses propres équipements et ressources (datacenters, matériels, systèmes hébergés, applications, données) soit garantie ;
- **La confidentialité** : en fonction de la criticité des applications du cloud, il devra veiller et demander des garanties à ce sujet (par exemple, si les machines virtuelles de clients différents sont hébergées sur un même serveur physique ou encore l'utilisation d'une base de données comportant des informations de plusieurs clients) ;
- **L'intégrité, la perte de gouvernance, de souveraineté et le vol de données** : bien connaître le fournisseur cloud devient un critère primordial pour le client (par exemple, en s'informant sur l'emplacement exacte des datacenters ou encore sur les mécanismes que le cloud provider a mis en place pour lutter contre l'altération et/ou la perte d'informations) ;
- **L'indisponibilité des services et la bande passante** : utiliser des services de cloud computing dépend fortement d'une connectivité à l'Internet : il s'agit d'un point de défaillance (Single Point Of Failure – SPOF) ;
- **Les impacts environnementaux** : un fournisseur cloud qui propose des services cloud implique le fonctionnement de nombreux équipements dans son infrastructure, souvent consommatrices d'énergie. Cependant, les apports du Green IT permettent par exemple de récupérer la chaleur émise par les datacenters pour le chauffage des bâtiments ;
- **La récupération des données et la réversibilité** : un fournisseur cloud se doit de proposer des API permettant aux clients de récupérer leurs données à tout moment, afin de les migrer chez un autre prestataire cloud ou éventuellement les rapatrier dans son infrastructure informatique interne.

## 2.2 Les aspects de sécurité du cloud.

La sécurité et la conformité émergent systématiquement comme les principales préoccupations des responsables informatiques lorsqu'il est question de Cloud Computing, des préoccupations encore plus accentuées lorsqu'il s'agit d'un Cloud public. La sécurité permet de garantir la confidentialité, l'intégrité, l'authenticité et la disponibilité des informations.

Certaines questions légitimes reviennent sans cesse :

- Mes données sont-elles sûres dans le Cloud ?
- Où sont stockées mes données ?
- Qui va avoir accès à mes données ?
- Aurais-je accès à mes données à n'importe quel moment ?
- Que deviendront mes données s'il y a interruption du service ?

La mise sur pied d'une solution de Cloud Computing comporte des problèmes de sécurité inhérents à la solution elle-même. Le fait de centraliser toutes les informations sur un site pose un grand nombre de problèmes. On peut citer comme problème potentiel :

- Une possible interruption massive du service.
- Une cible de choix pour les hackers
- Interface et API non sécurisé

Ce point de vulnérabilité du Cloud Computing fait l'objet depuis quelques années l'objet de recherches avancées. Il a été créé un organisme chargé de mettre sur pied des normes en matière de sécurité dans le Cloud Computing. Cet organisme s'appelle CSA (Cloud Security Alliance). Du travail de cet organisme, il en est ressorti certaines techniques utilisées de nos jours pour améliorer la sécurité du Cloud Computing. Parmi ces techniques on peut citer :

- La multi-location : cette technique permet de créer des instances d'une même donnée sur plusieurs sites différents. Elle permet une récupération facile en cas de désastre.
- Le chiffrement : le chiffrement de l'accès à l'interface de contrôle, le chiffrement des données dans le Cloud.
- L'isolation des machines virtuelles.

### **2.2.1 Les rappels sur la cryptographie.[20]**

L'échange d'informations sensibles a toujours été une problématique qui existe depuis des millénaires. Quel que soit le support et le moyen d'échanges de ces données, il est nécessaire d'assurer leur sécurité. Actuellement, les techniques permettant de protéger les échanges reposent sur la cryptologie.

**La cryptologie** est l'art et la science du chiffrement. Il regroupe *la cryptographie* et la *cryptanalyse*.

**La cryptographie** est le mécanisme permettant de transformer un message clair (information qui n'est pas protégée) en un message inintelligible (donnée qui est sécurisée) par celui qui ne possède pas les clés de chiffrement. Au cours des siècles, les procédés cryptographiques se sont développés :

- **Code** : ensemble des moyens employés pour remplacer les lettres d'un message à chiffrer ;
- **Chiffre** : ensemble des moyens employés pour remplacer les mots du message à coder ;
- **Stéganographie** : technique permettant de camoufler le message dans un support de manière à masquer sa présence.

La cryptanalyse est l'art de « casser » un message chiffré. Il s'agit, pour un cryptanalyste, de récupérer une information protégée afin d'en connaître le contenu, sans avoir les clés de chiffrement. Les techniques de cryptanalyse dépendent du niveau de l'offensive du cryptanalyste. Ils sont définis de la façon suivante :

- **Attaque à texte chiffré** (Ciphertext-only attack) : obtenir des informations sur la clé et sur le message en clair uniquement à partir du message chiffré. Il s'agit du type d'attaque le plus difficile et le plus coûteux en temps ;
- **Attaque à texte clair connu** (Known-plaintext attack) : obtenir les informations sur la clé, en ayant le message clair et le message chiffré ;
- **Attaque à texte clair choisi** (Chosen-plaintext attack) : obtenir les informations sur la clé, en pouvant générer le message chiffré du message en clair que l'on possède déjà ;
- **Attaque à texte chiffré choisi** (Chosen-ciphertext attack) : obtenir les informations sur la clé, en pouvant déduire le message en clair à partir du message chiffré que l'on possède déjà.

Les principaux moyens cryptographiques actuels sont les suivants :

- **Le chiffrement symétrique ou chiffrement à clé secrète partagée** : utiliser la même clé pour chiffrer et déchiffrer un message. L'avantage de ce chiffrement est sa rapidité dans les opérations de calculs, car les tailles de clés sont petites. L'inconvénient étant qu'une clé est nécessaire pour chaque couple de participants à un échange d'information. Quelques exemples d'algorithmes de chiffrement symétrique : Data Encryption Standard (DES), Advanced Encryption Standard (AES) ;



- **Le chiffrement asymétrique ou chiffrement à clé publique** : utiliser une clé différente pour le chiffrement (clé publique du destinataire utilisée par l'émetteur du message) et le déchiffrement d'un message (clé privée du destinataire). L'avantage étant une distribution aisée des clés aux participants à l'échange d'information. L'inconvénient étant les opérations de calculs plus lentes, car les tailles de clés sont grandes. Le Rivest Shamir Adleman (RSA) est un exemple d'algorithme de chiffrement asymétrique ;
- **Le hachage** : obtenir une empreinte numérique d'un message. Quelques exemples de fonctions de hachage : **Message Digest 5 (MD5)**, **Secure Hash Algorithm (SHA)** et ses dérivés (**SHA-1**, **SHA-256**, **SHA-512**).

En pratique, les chiffrements symétriques et asymétriques sont utilisés conjointement, afin d'ailler leurs avantages (le chiffrement symétrique pour chiffrer et déchiffrer, et le chiffrement asymétrique pour transmettre les clés de chiffrement symétrique).

### 2.2.2 Les objectifs de sécurité.

À l'instar des besoins de sécurité pour une infrastructure interne (On Premise), les objectifs de sécurité pour un environnement cloud sont les suivants :

- **Authentification** (Authentication) : s'assurer de l'identité d'une personne ou d'une entité (ou bien en détecter une usurpation), afin d'avoir la garantie que la personne / l'entité est bien celle qu'elle prétend être ;
- **Confidentialité** (Confidentiality) : s'assurer que les informations stockées dans le cloud ne sont accessibles qu'aux personnes autorisées (ou en d'autres termes, empêcher l'accès aux données à ceux qui n'en sont pas les destinataires) ;
- **Intégrité** (Integrity) : s'assurer que les informations n'ont pas été altérées, notamment pendant le transport des données dans un environnement cloud, afin d'avoir la garantie que les données sont complètes et exactes dans cet environnement ;
- **Non-répudiation ou Imputabilité** (Non-repudiation) : s'assurer que l'émetteur et le destinataire d'une information sont bien ceux qui prétendent être et que l'information envoyée est bien conforme à celle reçue. Techniquement, il s'agit d'une combinaison entre les mécanismes d'authentification et d'intégrité ;

- **Disponibilité** (Availability) : s'assurer que les services, les ressources et les réseaux informatiques soient accessibles ;
- **Contrôle d'accès** (Access control) : s'assurer que des moyens ont été mis en place pour limiter l'utilisation de systèmes ou d'applications ;
- **Fraîcheur** (Refreshing) : s'assurer lors d'une communication de données du caractère « récent » de ces informations, afin d'éviter des attaques dites « par rejeu » ;
- **Traçabilité** ou **Preuve** ou **Auditabilité** (Evidence) : s'assurer d'avoir les moyens de prouver la réalité des actions.

### 2.2.3 Les attaques potentielles.

Avant de recenser les différentes attaques probables sur des environnements en nuage, rappelons tout d'abord les types de surface d'attaque d'un système informatique (en effet, sur tout système accessible, ses surfaces d'attaques doivent être connues) :

- **La surface d'attaque réseau** : des ports ouverts sur les pare-feux ou des interfaces ouvertes sur les routeurs et les commutateurs du fournisseur cloud peuvent donner des suggestions d'offensives pour un attaquant ;
- **La surface d'attaque logicielle** : il s'agit de tous les points d'entrée et de sortie d'une application informatique, qui peut être susceptible de présenter des vulnérabilités exploitables pour un attaquant ;
- **La surface d'attaque humaine et physique** : les datacenters des fournisseurs cloud sont des sites physiques dont il convient d'assurer une sécurité plus qu'optimale pour éviter les vols de données.

Ainsi, nous pouvons recenser les types d'attaques possibles sur un environnement cloud :

- **Déni de service** (Denial of Service – DoS) : pour un attaquant, il s'agira de rendre indisponible les services et/ou les ressources du fournisseur cloud pendant un temps indéterminé. A l'heure actuelle, un attaquant utilisera plusieurs moyens pour qu'un service devienne indisponible, on parlera plutôt de déni de service distribuée (Distributed Denial of Service – DDoS) ;

- **Injection de malware** (Malware injection) : pour un attaquant, il tentera d'injecter un service malveillant ou une machine virtuelle malveillante dans l'environnement du fournisseur cloud, en vue des récupérer des informations ;
- **Canal auxiliaire, latéral ou caché** (Side-channel) : pour un attaquant, il s'agira d'exploiter les vulnérabilités matérielles et logicielles de l'environnement du fournisseur cloud, en vue de récupérer des données ;
- **Authentification** (Authentication) : pour un attaquant, il s'agira d'exploiter le processus d'authentification permettant l'accès au service ou à la ressource cloud, qui permet de vérifier l'identité d'un utilisateur ;
- **Homme du milieu** (Man In The Middle – MITM) : pour un attaquant, il tentera de se placer entre l'utilisateur du service cloud et les ressources du fournisseur cloud, en vue de récupérer les informations.
- **Physique** : pour un attaquant, il tentera de s'introduire dans l'un des centres de données du fournisseur cloud, en vue de récupérer les informations

## 2.2.4 Le cycle de vie de la sécurité des données dans le cloud.

Le Cloud Security Alliance a défini un modèle de cycle de vie de la protection des données dans le cloud à six étapes : création ou mise à jour d'une donnée (Create), transfert et stockage de la donnée dans un emplacement de stockage (Store), utilisation de la donnée (Use), partage de la donnée (Share), archivage de la donnée (Archive) et destruction de la donnée (Destroy).

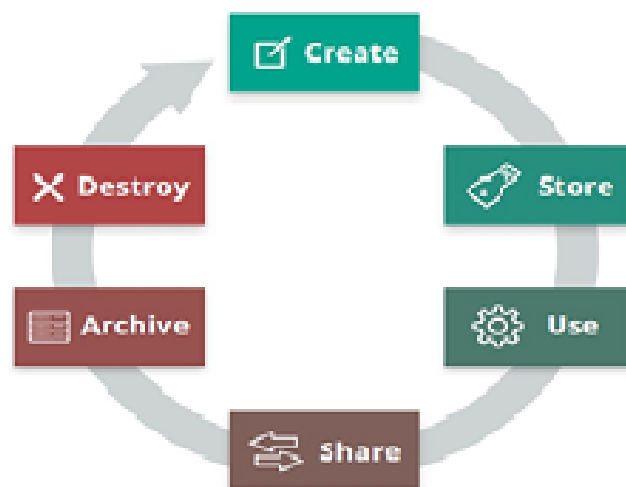


FIGURE 2.7 – Cycle de vie de la sécurité des données dans le cloud.

Ce modèle est un principe général, car une donnée ne passe pas forcément toutes ces étapes (par exemple, toutes les données qui ont été créées ne sont pas toutes détruites).

La sécurité absolue n'existe pas, donc le problème de sécurité reste le plus souvent un problème de confiance entre le fournisseur de service et le consommateur de service. Cette confiance se traduit par la signature d'un contrat nommé SLA (Service Level Agreement). Ce contrat Précise les taux de disponibilité du service. En règle générale, et pour la plupart des fournisseurs, ce taux est supérieur à 99

## 2.3 Gestion d'un projet de cloud computing.

### 2.3.1 Caractéristiques de la solution voulue.[21]

Dans ce tableau, nous présentons les caractéristiques recherchées dans la solution à mettre en place.

Tableau 2.2 – Caractéristiques de la solution voulue

Objectif	Définition	Comment
Fiabilité	Assurer l'authenticité des informations de tout le système	En connectant le système à une base de données qui sauvegarde les données en temps réel.
Intégrité	Assurer l'authenticité du système c'est-à-dire la non modification possible du contenu ou de la forme n'importe comment ou par n'importe qui.	En donnant les droits de modifications à des personnes désignées par l'administrateur.
Sécurité	Assurer la sécurité des informations.	En définissant un algorithme variable (régulièrement modifié pour prévention aux tentatives de décryptage) selon chaque système qui assurera la vérification de tous les entrées et sorties des informations du systèmes.
Disponibilité	Assurer l'accès permanent aux informations du système	En mettant en place des serveurs de réplication et de stockage pouvant gérer le load balancing
Évolutive	Permettre l'augmentation des fonctionnalités ou besoins d'un système sans interruption.	En prévoyant la possibilité que le système puisse grandir dans le même environnement.

## 2.3.2 Objectif du Projet

Le Cloud Computing traduit un ensemble des services d'infrastructure qui sont opérés par un hébergeur tiers. L'objectif de ce projet est de garantir une exploitation du système d'information plus souple, flexible, disponible et sécurisé en accord avec les besoins métiers à tout instant.

Notre solution doit garantir les services suivants :

- **Gestion unifiée** : A l'inverse d'une gestion et supervision des multiples systèmes, la gestion de l'infrastructure se fait à travers une « interface ».
- **Services à la demande** : Les besoins métiers sont variables. L'IT doit être réactif pour fournir des services performants et en un minimum de temps. Ainsi les niveaux de services fournis par les acteurs de l'IaaS répondent à de tels enjeux.
- **Interopérabilité** : Les infrastructures on-prémisses sont souvent soumises à des contraintes techniques propriétaires (ex : middlewares spécifiques pour certaines infrastructures physiques). L'IaaS réduit désormais les problématiques à la bonne définition du besoin technique.

Le cloud permet de se dégager des adhérences fortes qui lient les DSI aux produits et vendeurs, et donc des complexités techniques, logistiques, contractuelles. Elles peuvent d'avantage se concentrer sur les innovations et les besoins métiers.

Mais mener une transition vers l'IaaS, nécessite des bases techniques et processus solides.

## 2.3.3 Méthodes d'analyses

### 2.3.3.1 Modèle de cycle de vie en cascade.[9]

Mis au point dès 1966, puis formalisé vers 1970, le cycle de vie de projet en cascade est un type de cycle de vie, simple à comprendre et à implémenter, convient aux projets où la qualité a plus d'importance que les coûts ou les délais, et dont les besoins sont clairement définis et stables. Dans le cas contraire, la prise en compte de nouveaux besoins nécessite de dérouler toute la cascade depuis le début. De plus, le client n'est impliqué qu'au début du projet et il ne peut tester le produit qu'à la fin du processus. Dans le cadre d'un projet de gestion des identités et des accès, les besoins peuvent évoluer. En effet, le déploiement de nouveaux services implique notamment la définition de nouveaux profils ainsi que de nouveaux rôles applicatifs qui doivent être pris en compte, même après la phase

de spécification. De ce fait, ce modèle de gestion de projet ne convient pas à notre projets de gestion des sinistres.

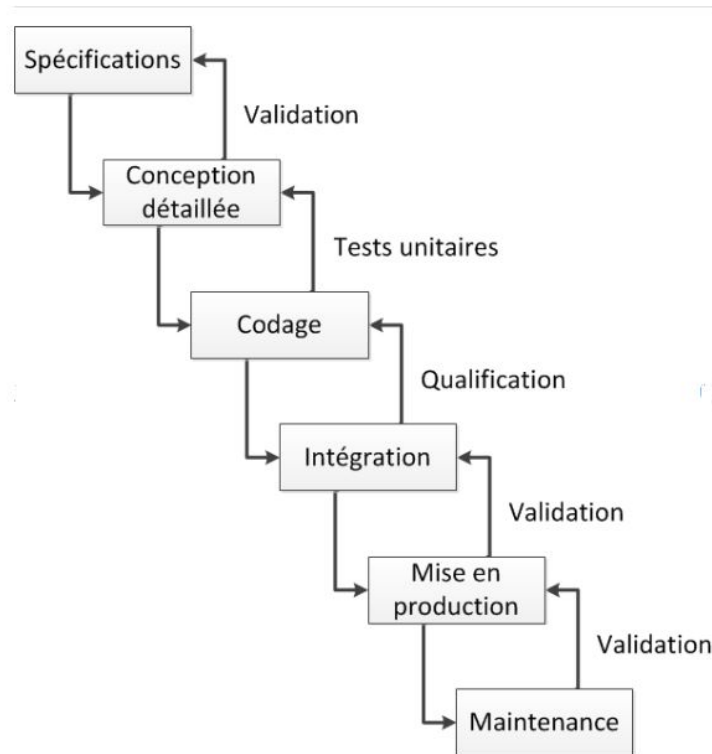


FIGURE 2.8 – Modèle du cycle de vie en cascade

### 2.3.3.2 Modèle de cycle de vie en V.[11]

A l’instar du modèle en cascade, celui en V prend difficilement en charge de nouveaux besoins ou la modification des spécifications. En effet, l’effet tunnel induit par les modèles séquentiels montre qu’une erreur dans la formulation ou l’interprétation des spécifications ne peut être détectée qu’à la fin du cycle. En effet, la maîtrise d’ouvrage n’est impliquée qu’en début et fin de cycle, ce qui peut représenter plusieurs mois d’intervalle pour un gros projet. Bien plus nombreuses que dans un cycle en V, les possibilités de prise en compte de nouveaux besoins restent faibles. En effet, dans le cadre de notre projet, après la phase de spécification, la mise à disposition d’un nouveau service, ne peut être prise en compte qu’au moment des tests d’intégration. De plus, ces changements impliqueraient la remise en cause du travail effectué jusqu’à la phase des tests unitaires.

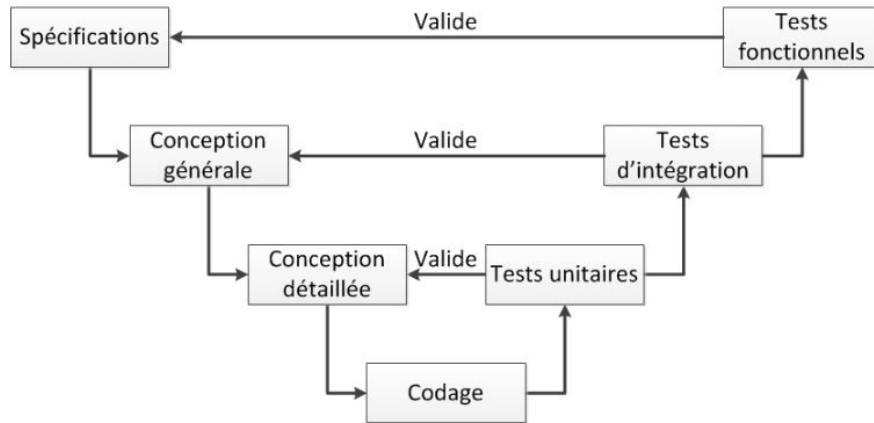


FIGURE 2.9 – Modèle de cycle de vie en V

### 2.3.3.3 Modèle de cycle de vie en spirale.[7]

Représenté à l'aide d'une spirale et proposé par Boehm en 1988, ce modèle est beaucoup plus général que le précédent. Chaque boucle de la spire représente une phase du développement celle la plus interne traite des premières phases (faisabilité). La plus externe traite de la livraison, chaque boucle traverse quatre sections :

- Définition des objectifs de la phase (ou boucle)
- Évaluation des risques et plan de gestion
- Développement et validation
- Planification de la phase suivante

le fait qu'il soit un méta-modèle entraine l'obligation de l'instanciation de chaque bouble, création d'une boucle de faisabilité, d'une boucle de prototypage, des boucles de développement itératif, etc.

De ce fait, il faut alors trouver le bon modèle de processus pour chaque boucle !

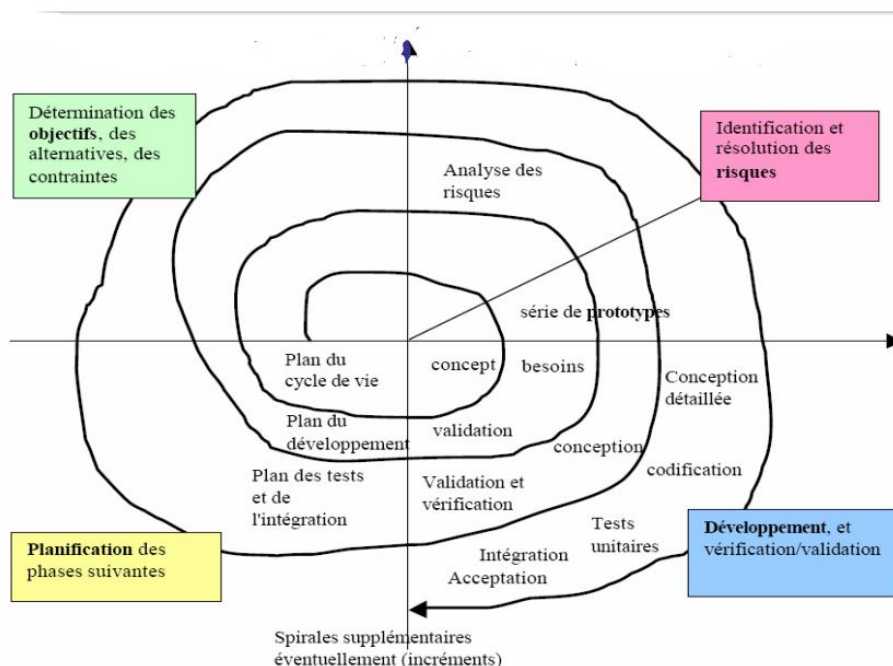


FIGURE 2.10 – Modèle de cycle de vie en spirale

#### 2.3.3.4 La méthode agile.[9]

Adopter les méthodes agiles constitue un acte courageux car il s'agit de quitter un système organisationnel, culturel et économique connu (étudié en cours et utilisé tout au long de la formation) pour s'orienter vers l'inconnu avant d'atteindre à nouveau un équilibre stable. Les raisons qui nous poussent à adopter une méthode agile sont nombreuses et variées. La capacité à s'adapter au changement, à livrer plus fréquemment et à accroître la qualité des logiciels ainsi développés figurent parmi les motivations les plus fréquentes. La bonne surprise réside dans le fait que la motivation des équipes est également une raison très souvent citée. Preuve est ainsi faite que les méthodes agiles représentent un système de valeurs et sont perçues par un grand nombre comme un changement culturel motivant, bénéfique pour tous.

L'agilité ou plutôt les méthodes agiles sont un groupe de processus et de pratiques pour le pilotage et la réalisation de projets. Toutes ces pratiques sont basés sur le manifeste agile, qui a été mis en place en 2001 et qui a pour but d'impliquer un maximum le client, ou bénéficiaire du projet, dans le développement, pour permettre une réactivité dans la réalisation de ses demandes.



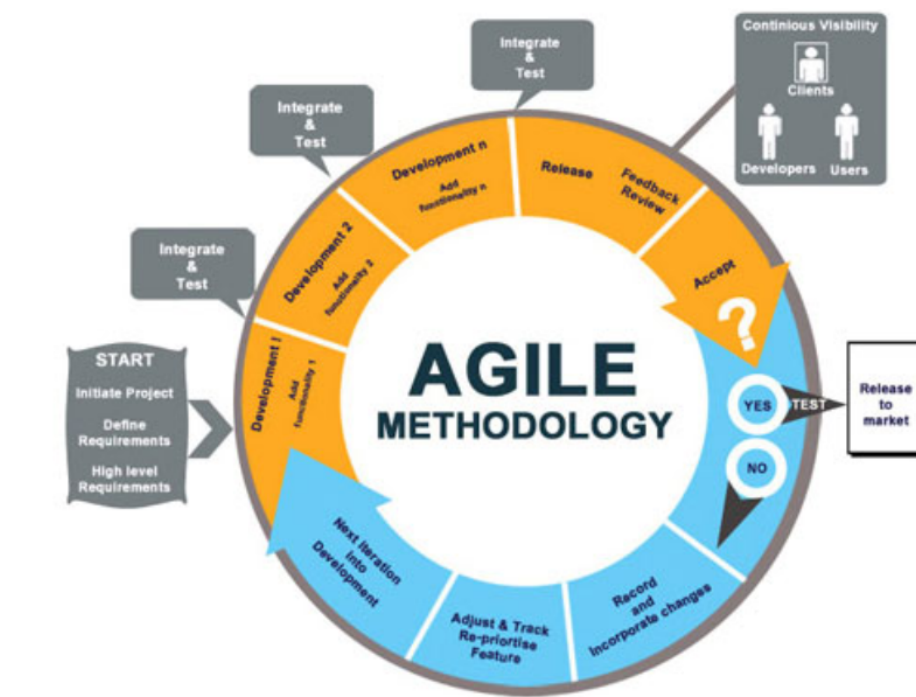


FIGURE 2.11 – Méthode agile

### 2.3.3.5 La méthodologie 2TUP ou cycle en Y

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de trois phases essentielles :

- **La branche fonctionnelle** capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.
- **La branche technique** capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système le sont indépendamment des fonctions à réaliser.
- **La phase de réalisation** consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins.

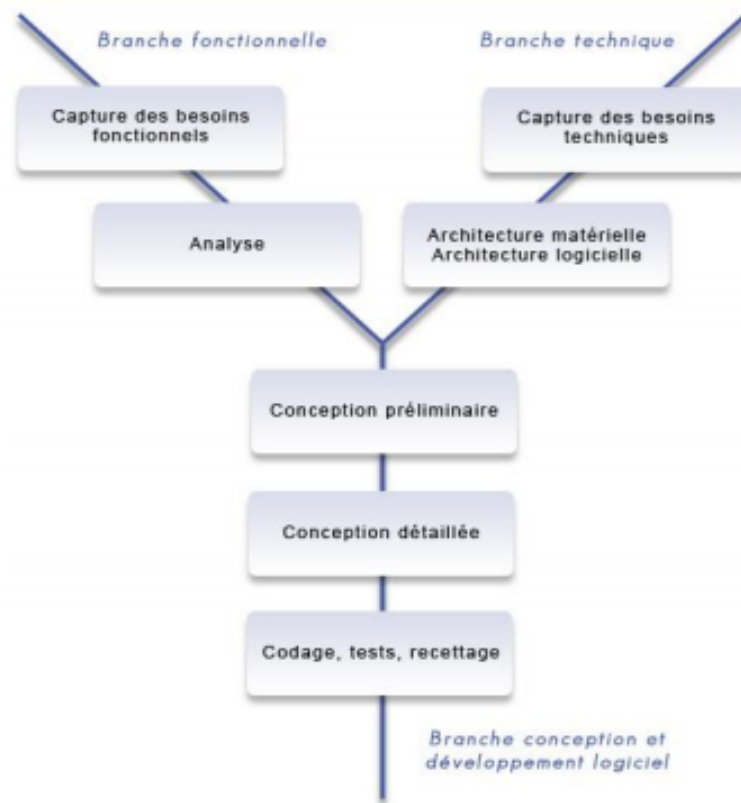


FIGURE 2.12 – Méthodologie 2TUP

## 2.4 Solutions du Cloud existante.

Dans cette section, nous présentons les différentes solutions de cloud et leurs caractéristiques afin de choisir la solution adéquate pour notre projet.

### 2.4.1 Solutions propriétaires.[6]

Comme solutions propriétaires, nous pouvons citer :

#### 2.4.1.1 VMwareCloud.

Les solutions de Cloud Computing VMware favorisent l'innovation et rendent l'environnement informatique plus efficace, plus flexible et plus fiable. VMware fournit à la direction informatique tout ce qui lui est nécessaire pour concevoir, faire fonctionner et gérer le Cloud, avec le personnel compétent, tout en quantifiant en permanence son impact. Avec l'aide de VMware, les clients font évoluer leurs « fondations » techniques, modèles organisationnels, processus d'exploitation et mesures financières. Ceci leur permet à la fois de bâtir une infrastructure de Cloud Computing et d'élaborer un

modèle d'opérations capable d'exploiter tous les avantages du Cloud Computing. Les solutions de Cloud Computing VMware optimisent les capacités du Cloud :

- **Déploiement de nouveaux services informatiques qui favorisent la croissance de l'entreprise** : Il devient plus facile et plus rapide de créer et fournir les services qui permettront à l'entreprise de se démarquer des autres.
- **Transformation de la direction informatique en moteur d'innovation** : Les ressources informatiques libérées peuvent être consacrées à la mise en place de services qui facilitent la réalisation des objectifs métiers.
- **Efficacité, flexibilité et fiabilité garantie.**

#### 2.4.1.2 Office 365.

C'est la version Cloud Computing de Microsoft avec des niveaux d'utilisation au choix : messagerie, office, partage et accès aux données. Avec Office 365, Microsoft optimise le Virtual Office, et offre une solution Cloud qui permet via un simple abonnement d'accéder à l'ensemble des données depuis n'importe quelle plateforme (PC, Smartphone, Tablette). Microsoft met en place cette offre personnalisée et adaptée aux différents besoins des entreprises.

Office 365 leur permet de choisir uniquement les modules utiles pour ses utilisateurs; en sélectionnant uniquement les options adaptées sans gaspillage .

L'objectif est de mettre en place une solution de Cloud Computing Office 365 pour l'entreprise afin de réduire les charges d'investissement et d'exploitation des serveurs et d'applications.

Le but recherché derrière cette démarche est d'externaliser la messagerie électronique, de permettre aux utilisateurs d'accéder à des documents partagés sur l'espace SharePoint online et de pouvoir communiquer à l'aide de la messagerie instantanée de la vidéo conférence et cela de façon intégrée et cohérente selon des règles d'accès précises à travers des rôles utilisateurs. Au besoin la solution sera intégrée en hybride avec le système d'information existant. La solution Cloud Office 365 proposée se focalise sur la mise en place des services suivants :

- **Externalisation de la messagerie** : Exchange online
- **Partage et gestion des documents sur Office 365** : SharePoint Online
- **Gestion de la communication Instantanée, Réunions et Conférences en ligne** : Lync Online.

## 2.4.2 Solutions libres.[6]

Comme solutions libres, nous pouvons citer :

### 2.4.3 Eucalyptus

Eucalyptus est un outil open source issue d'un projet de recherche de l'université de Californie. Cette solution est la plus connue, car elle est intégrée dans les distributions Ubuntu Server et Debian.

Eucalyptus est écrit en C, Java et Python et permet de créer des Clouds IaaS de type privé ou hybride. Il supporte les machines virtuelles Linux ainsi que les hyperviseurs Xen et KVM. Son avantage majeur est le fait qu'il est compatible avec Amazon EC2.

Il possède également une version entreprise (payante) de la société Eucalyptus Systems qui apporte des fonctionnalités supplémentaires comme le support de VMware.

### 2.4.4 OpenNubela

Il s'agit d'une plateforme purement open-source permettant de déployer des Clouds privés, hybrides et publiques. Elle est écrite en C++, Ruby et Shell et elle supporte les hyperviseurs Xen, KVM et VMware. Le support de Virtualbox est prévu à partir de la version 4.0 de VirtualBox. Sa puissance consiste dans ses connecteurs vers des fournisseurs d'IaaS sur les Clouds publics tels que : Amazon EC2 Web Service, Nimbus WSRF, ElasticHosts REST, etc.

OpenNebula est soutenu par le projet européen **RESERVOIR**, qui propose une architecture complète pour la gestion de Datacenter et la création de services Cloud.

### 2.4.5 OpenStack

Créé en juillet 2010 par la NASA et l'hébergeur américain Rackspace, OpenStack est une offre d'IaaS 100% open-source encore en développement qui a livré son code source récemment et qui permet aux sociétés de développer leurs propres solutions d'infrastructure du Cloud Computing.

Plus que trente fournisseurs soutiennent ce projet tels que : AMD, Intel, Dell et Citrix. OpenStack devrait également être intégré dans les prochaines versions d'Ubuntu comme c'est le cas pour Eucalyptus. Il comprend le logiciel OpenStackCompute pour la création automatique et la gestion de grands groupes de serveurs privés virtuels et le logiciel OpenStack Stockage pour optimiser la gestion de stockage, répliquer le contenu sur différents serveurs et le mettre à disposition pour une utilisation massive de données.

## 2.5 Critique générale.

### 2.5.1 Comparaison entre les solutions du cloud computing.

Dans les paragraphes précédents, nous avons présenté une liste de logiciels permettant de créer des solutions Cloud. Il est à présent temps de faire le choix de celui qui nous convient le mieux. Une comparaison est menée dans le tableau ci-dessous, selon plusieurs critères choisis en fonction des conseils trouvés dans l'état de l'art.

Tableau 2.3 – Comparaison entre les solutions Cloud

	<b>OpenStack</b>	<b>Eucalyptus</b>	<b>OpenNubela</b>
<b>Source Code</b>	Entièrement opensource, apache v2.0	Entièrement opensource, GPL v3.0	Entièrement opensource, apache v2.0
<b>Produit par</b>	Rackspace, NASA, Dell, Citrix, Cisco, Canonical et plus que 50 autres organisations	Apparu au début dans l'université Santa Barbara de l'université de Californie -Eucalyptus System Company	L'union Européenne
<b>But</b>	Créer et ouvrir des fonctionnalités de Cloud Computing en utilisant un logiciel opensource fonctionnant sur du matériel standard	Une réponse open source pour le Cloud commerciale EC2	Un Cloud privé pur
<b>Domaine d'utilisation</b>	Les sociétés, les fournisseurs de services, les chercheurs et les centres de données mondiaux qui cherchent à déployer à grande échelle leurs Cloud privés ou publiques	Les entreprises	Les chercheurs dans le domaine du Cloud Computing et de la virtualization

<b>Système d'exploitation supportés</b>	Linux et récemment Windows Exige x86 processor	Linux (Ubuntu, Fedora, CentOS, OpenSUSE et Debian)	Linux (Ubuntu, RedHat EnterpriseLinux, Fedora et SUSE Linux Enterprise Server)
<b>Langage de programmation</b>	Python	Java, C, Python	Java, C++, Ruby
<b>Stockage</b>	OpenStackStorage	Walrus	GridFTP, Comulus (version récente de GridFTP - XCP )
<b>Maturité</b>	Jeune, mais prometteur. Soutenu par de grands acteurs de différents secteurs (informatique, aéronautique, etc).	Aboutie, solution intégrée à Ubuntu Server, produit complet avec une interface de gestion web fonctionnelle.	Avancée, deuxième version stable, solution supportée par Debian.
<b>Hyperviseur</b>	Xen, KVM	Xen, KVM	Xen, KVM, VMware
<b>Installation</b>	Facile, installation automatisée et documentée.	Problématique, dépend de l'environnement réseau et matériel, difficulté en environnement hétérogène.	Manuelle, installation facile sur les distributions supportées (dont Debian et Ubuntu).
<b>Orientation</b>	Cloud public et privé	Cloud public et privé	Cloud privé
<b>Documentation</b>	Excellente, site bien fourni et facile d'accès avec à la fois un wiki contenant l'essentiel et une documentation officielle disponible et très détaillée.	Correcte, complète mais pas toujours à jour.	Complète, documentations, références de tous les fichiers de configuration, exemples. Manque d'aide sur un environnement complexe.

## 2.5.2 Comparaison des méthodes d'analyse.

Modéliser consiste à créer une représentation virtuelle d'une réalité de manière à faire ressortir les points auxquelles on s'intéresse, dans ce domaine deux approches se démarquent : UML et MERISE qui sont résumés dans le tableau suivant :

Tableau 2.5 – Comparaison entre les méthodes de modélisation

	Description	Points forts	Points faibles
<b>MERISE</b>	Méthode d'analyse, de conception et de gestion de projet informatique a été très utilisée dans les années 1970 et 1980 pour l'information massive des organisations	Revue de code permanente	Vérification de la concordance entre données et traitement. Étape de développement liée les unes aux autres.
<b>UML</b>	UML est la fusion de Trois méthodes, tout en restant simple et homogène. Ce langage est devenu le standard et termes de modélisation objet.	Il est devenu un standard en termes de modélisation. Il est polyvalent et performant. Il est orienté objet	Lourdeur dans la mise en place. Apprentissage assez long et rigoureux

## 2.6 Les scanners des vulnérabilités.[23]

Un **scanneur de vulnérabilité** est un programme conçu pour identifier des vulnérabilités dans une application, un système d'exploitation, ou un réseau, cloud.

Les scanners de vulnérabilité peuvent être utilisés dans des objectifs licites ou illicites :

- **objectifs licites** : les experts en sécurité informatique des entreprises utilisent les scanners de vulnérabilité pour trouver les failles de sécurité des systèmes informatiques et des systèmes de communications de leurs entreprises dans le but de les corriger avant que les pirates informatiques ne les exploitent ;

- **objectifs illicites** : les pirates informatiques utilisent les mêmes équipements pour trouver les failles dans les systèmes des entreprises pour les exploiter à leur avantage.

Il existe plusieurs programmes :

- **Nexpose**, un scanneur de vulnérabilité de Rapid7 (propriétaire de Metasploit).
- **Nessus**.

- **OpenVAS**, un scanneur de vulnérabilité libre.
- **Snort**, un système de détection d'intrusion.
- **Nmap**, un scanneur de ports.

Nous allons présenter une petite explication sur le meilleur scanneur dans cette partie (Nessus, Nmap).

### 2.6.1 Nessus

**Nessus** est un outil de sécurité informatique. Il signale les faiblesses potentielles ou avérées sur les machines testées.

Ceci inclut, entre autres :

- Les services vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles (lecture de fichiers confidentiels par exemple), des dénis de service.
- Les fautes de configuration (relais de messagerie ouvert par exemple)
- Les patches de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée.
- Les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe Hydra pour attaquer les mots de passe à l'aide d'un dictionnaire.
- Les services jugés faibles (on suggère par exemple de remplacer Telnet par SSH).
- Les dénis de service contre la pile TCP/IP .
- Scan les vulnérables des Cloud Computing.

### 2.6.2 Nmap

Nmap est un scanner de ports libre créé par Fyodor et distribué par Insecure.org. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'une machine distante.

Ce logiciel est devenu une référence pour les administrateurs réseaux car l'audit des résultats de Nmap fournit des indications sur la sécurité d'un réseau. Il est disponible sous Windows, Mac OS X, Linux, BSD et Solaris.

On utilise le Nmap parce que le OpenStack est installé sur une machine Virtual.



## 2.7 Choix des outils.

### 2.7.1 Choix de la méthode d'analyse.

Compte tenu du type du projet et ses spécificités, nous avons opté pour une démarche qui répond à notre besoin fonctionnelle, La méthode « 2TUP » est un processus de développement logiciel qui implémente le Processus Unifié.

Dans notre cas nous choisissons l'utilisation de la branche technique pour avoir identifié les besoins technique et l'architecture matérielle pour réaliser notre Projet.

Nous passons par les deux branches, branche fonctionnelle et branche technique ensuite on converge vers la phase de réalisation, ci-dessous la figure qui présente notre démarche par détail.

### 2.7.2 Choix de la solution à déployer.[19]

Dans les paragraphes précédents, nous avons présenté une liste des logiciels permettant de créer des solutions Cloud Computing. La mise en place d'un environnement de ce dernier pour des buts de recherche nécessite initialement le choix d'une solution :

- Open source sécurisée (Sous licence libre)
- Facile à installer et déployer
- Extensible
- Modulaire et innovante
- S'adaptant à tous types d'infrastructures existantes
- S'adressant à toutes les tailles d'entreprise
- Bien documenté
- Sous licence libre

Donc la solution qui convient le mieux et répond à nos besoins est **OpenStack**, la figure ci-dessous présente le pourcentage d'utilisation du logiciel OpenStack par rapport aux autres solutions selon Zenoss.com.

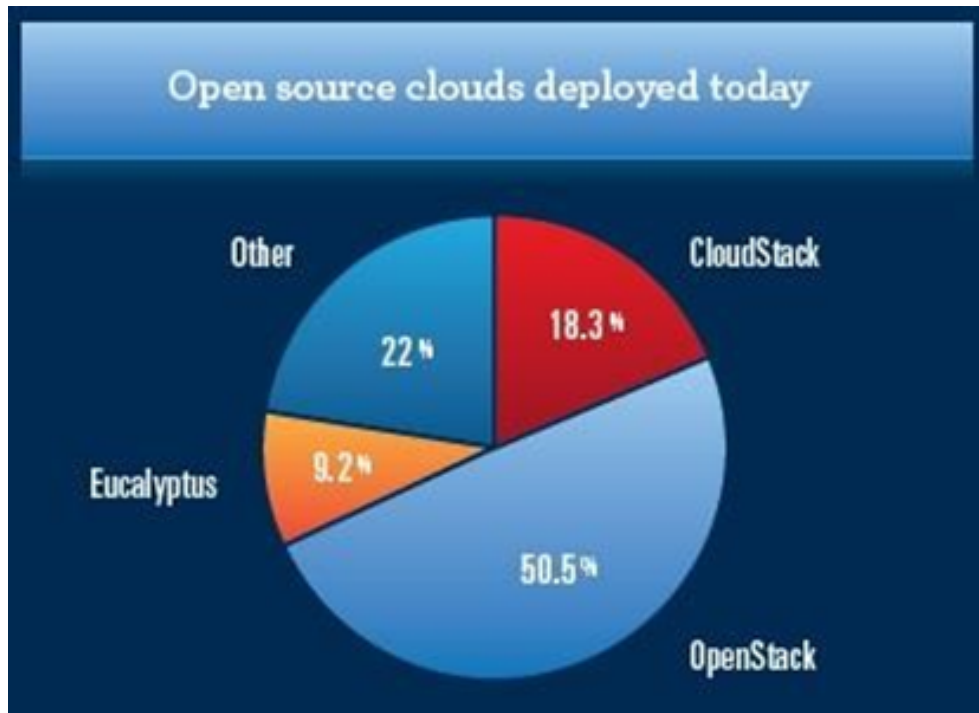


FIGURE 2.13 – Pourcentage d'utilisation d'OpenStack

### 2.7.3 Présentation

**OpenStack** est un logiciel libre qui permet la construction de Cloud privé et public de type IaaS sous licence Apache 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.

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

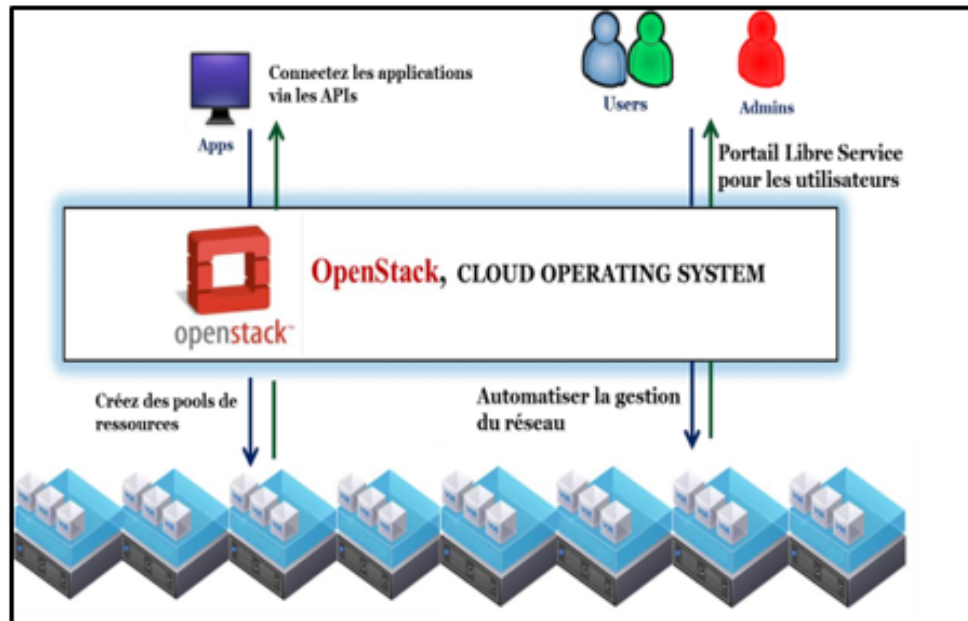


FIGURE 2.14 – Le rôle d’OpenStack

OpenStack est composé d’une série de logiciels et de projets au code source libre qui sont maintenus par la communauté incluant : OpenStackCompute (nommé Nova), OpenStack Object Storage (nommé Swift), et OpenStack Image Service (nommé Glance).

La figure suivante montre l’écosystème d’images d’OpenStack en se basant sur ses trois projets

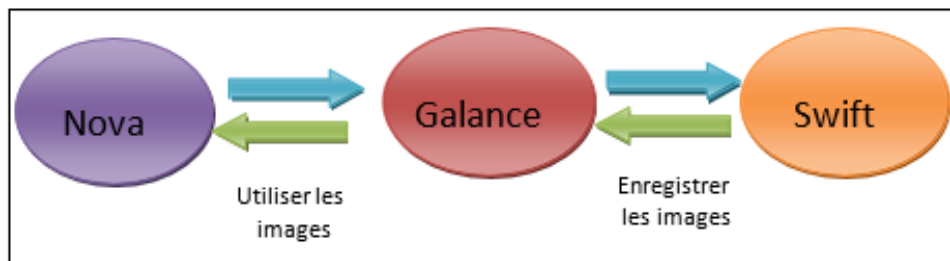


FIGURE 2.15 – Ecosystème d’images d’OpenStack

## 2.7.4 Architecture.[15]

Elle s’articule autour de trois composants :

### 2.7.4.1 OpenStackCompute (projet Nova).

**Compute** sert à la gestion de larges réseaux de machines virtuelles et d’une architecture redondante et évolutive. Elle fournit une interface d’administration et l’API nécessaire à l’orchestration du Cloud. Elle inclue : les gestions des instances serveurs, la gestion du réseau et les contrôle d’accès.

## Architecture de Nova

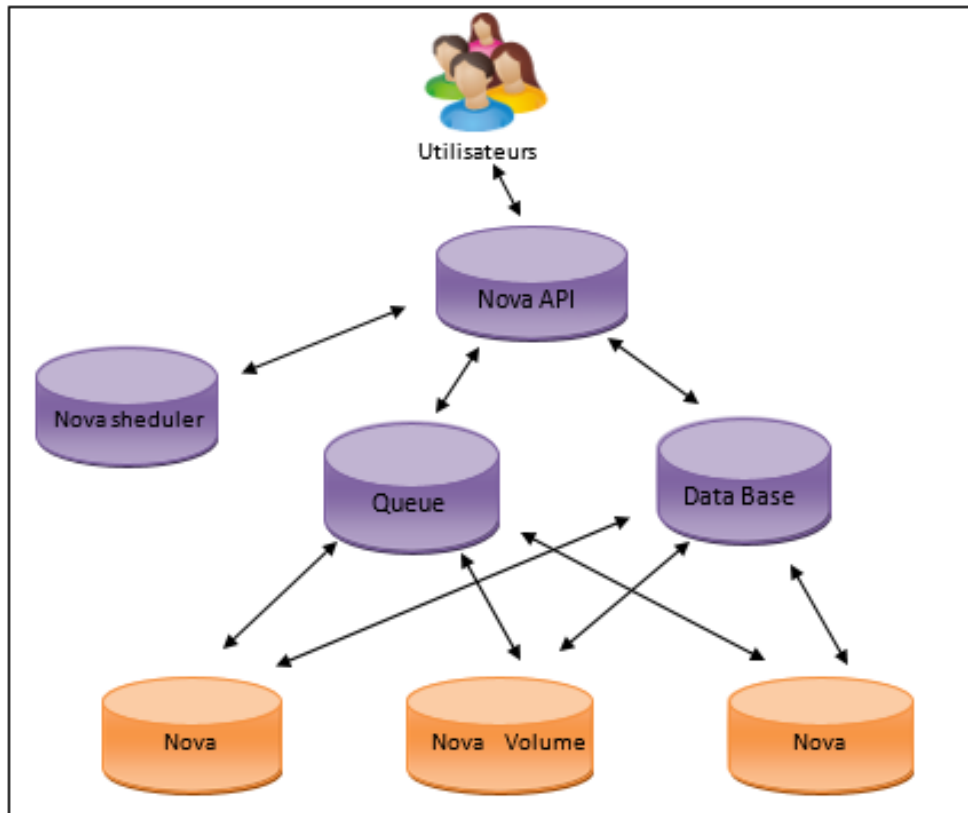


FIGURE 2.16 – L'architecture Nova.

Le tableau ci-dessous va nous permettre de comprendre l'architecture Nova Compute et les rôles de chaque composant :

Tableau 2.6 – Les composants de Nova

Le composant	Le rôle
<b>Nova</b>	
API	<ul style="list-style-type: none"> <li>- Cœur de Nova</li> <li>- Fonction Principale : Cloud Controller avec le service nova-api.</li> <li>- Compatible avec l'API Amazon EC2</li> <li>- Écoute sur le port 8773 pour EC2 API et 8774 pour OpenStack API</li> <li>- Initialise la plupart des activités</li> <li>- Renforce certaines fonctionnalités (ex : quotas).</li> </ul>

Scheduler	<ul style="list-style-type: none"> <li>- Principe simple : il prend une demande d'instance de machine virtuelle et détermine où (quel « Compute server ») doit-elle être exécutée.</li> <li>- Fonctionnement par algorithmes pour assurer un fonctionnement optimal.</li> <li>- Trois choix d'ordonnancement : <ul style="list-style-type: none"> <li>+ Simple : tente de trouver l'hôte le moins « chargé »</li> <li>+ Chance (celui par défaut) : choisit un hôte disponible au hasard depuis sa « Service Table »</li> <li>+ Zone : Prend un hôte au hasard depuis une zone « disponible »</li> </ul> </li> </ul>
Nova Compute	<ul style="list-style-type: none"> <li>- Créé et termine les instances de machines virtuelles</li> <li>- Reçoit et exécute des actions visant à mettre à jour les états des VM dans la base de données</li> <li>- Supporte plusieurs API : KVM, Xen, Citrix, VMware, Hyper-V.</li> </ul>
Nova Volume	<ul style="list-style-type: none"> <li>- Gère la création, l'attachement et le détachement de volumes persistants.</li> <li>- Compatible avec AoE, iSCSI (dont Solaris ZFS), Sheepdog, RBD, LeftHand (HP).</li> </ul>
Nova Network	<ul style="list-style-type: none"> <li>- Configure les interfaces bridge</li> <li>- Adapte les règles de pare-feu (Iptables)</li> <li>- Support des VLAN : chaque projet dispose de sa plage d'adresses IP accessibles via VLAN.</li> <li>- Deux types d'adresse IP pour une instance : <ul style="list-style-type: none"> <li>+ <i>Adresse fixe</i> : privée</li> <li>+ <i>Adresse provisoire</i> : publique deux gestionnaires de réseaux : <ul style="list-style-type: none"> <li>* <i>Flat</i> : adresse fixe attachée à l'interface bridge</li> <li>* <i>Flat DHCP</i> : adressage dynamique pour chaque interface bridge</li> </ul> </li> </ul> </li> </ul>
Data Base	<p>Enregistre la configuration et les états en temps réels pour une infrastructure Cloud : types d'instances disponibles, instances en cours d'utilisation, réseaux disponibles, projets.</p> <p>Supporte la plupart des SGBD : MySQL, PostgreSQL</p>

### 2.7.4.2 OpenStack Object Storage (projet Swift).

Object Storage sert à la création d'espace de stockage redondant et évolutif pour le stockage de plusieurs pétaoctets de données. Il ne s'agit pas réellement d'un système de fichier mais est surtout conçu pour le stockage à long terme de gros volumes. Il utilise une architecture distribuée offrant plusieurs points d'accès pour éviter les SPOF (Single Point Of Failure).

#### Architecture de Swift

Swift gère trois types d'objets différents :

- **Swift-Account** : Gère une base de données Sqlite3 contenant les objets de stockage
- **Swift-Container** : Gère une autre base de données Sqlite3 contenant la topologie des conteneurs
- **Swift-Object** : Topologie des objets réels enregistrés sur chaque nœud.

La figure suivante présente l'architecture Swift.

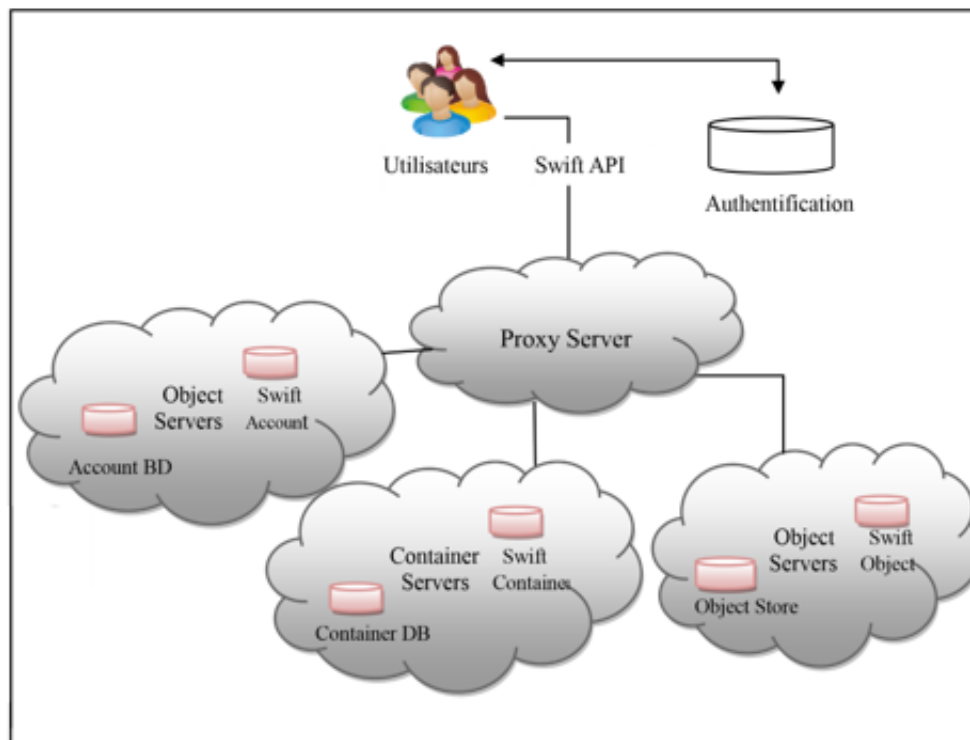


FIGURE 2.17 – L'Architecture de Swift.

### 2.7.4.3 OpenStack Imaging Service (projet Glance).

**Imaging Service** fournit les services de stockages, de découvertes, d'enregistrements et de distributions pour les images disques de machines virtuelles. Il fournit également une API compatible REST permettant d'effectuer des requêtes pour obtenir des informations sur les images hébergées par les différents magasins de stockages.

La figure ci-dessous montre l'interaction entre les différents composants d'OpenStack vu précédemment

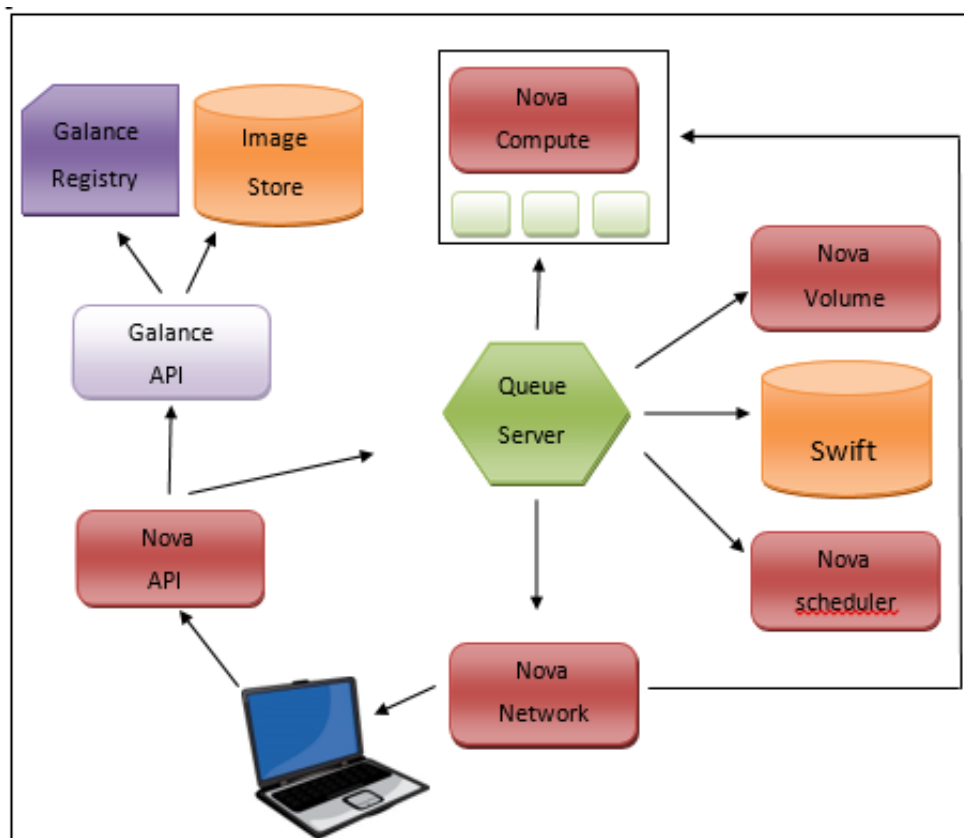


FIGURE 2.18 – L'Architecture d'OpenStack.

En plus des composants principaux il y a aussi d'autres composants complémentaires comme :

- **Quantum** : permet d'offrir une gestion des réseaux à la demande à l'intérieur de son Cloud. Le service permet aux utilisateurs de créer des réseaux à la demande et d'y attacher des machines virtuelles. Quantum a une architecture ouverte grâce à des plugins permettant de supporter différents fournisseurs de réseau ou des technologies réseaux différentes.
- **Cinder** : permet d'offrir des disques persistants pour les machines virtuelles. Ce service était inclus dans Nova à l'origine (sous le nom nova-volume) dans les versions précédentes de OpenS-

tack.

- **Horizon** : est une interface Web permettant d'agir sur les différents services d'OpenStack. Avec cette interface Web vous pouvez créer des machines virtuelles, assigner des adresses IP ou gérer le contrôle d'accès.
- **Keystone** : permet d'offrir une gestion de l'identité et des autorisations d'accès pour les différents services d'OpenStack.

Nous venons de présenter les notions fondamentales du Cloud Computing, ses enjeux, ses évolutions et son utilité ainsi que la technologie qui la constitue et les différents acteurs du secteur, une liste non exhaustive des applications permettant de créer des Cloud privés, puis de manière synthétique les outils technologiques et les méthodes qui ont contribué à produire ce travail.

Pour notre déploiement, notre choix s'est arrêté sur OpenStack. Pour tous ses avantages et surtout car il correspond exactement à ce que nous voulons déployer.

Ensuite une étude comparative des méthodes/langage de modélisation qui sera la méthodologie 2TUP. Pour finir, côté outils technologiques PostgreSQL en tant que SGBD, PowerAMC pour la modélisation, le Pycharm pour le codage.



## **Deuxième partie**

# **ANALYSE MODELISATION ET MISE EN PLACE DU PROJET**

## CHAPITRE 3

### PRÉSENTATION DE LA SOLUTION ET ANALYSE

Ce chapitre présente le cahier de stage et décrit la branche fonctionnelle du 2TUP. En effet, une première section du chapitre sera consacrée à la spécification et l'élaboration des différents cas d'utilisations.

Donc, elle sera orientée à énoncer les besoins fonctionnels auxquels devrait répondre le nuage privé à réaliser, ainsi que les besoins non fonctionnels qu'il devrait respecter.

La deuxième section de ce chapitre sera pour la partie technique. Dans laquelle le projet entame la phase technique qui permettra de décrire de manière détaillée l'architecture du nuage afin de faciliter sa réalisation.

### 3.1 Cahier de charges

#### 3.1.1 Nom du projet.

Étude et mise en place d'un système qui assure la disponibilité et la sécurité des données dans un cloud computing

#### 3.1.2 Origine et genèse du projet.

Le souci de toute entreprise est d'offrir un système assurant la disponibilité, la fiabilité, la traçabilité et la sécurité de ses informations, qu'on peut consulter régulièrement, peut ressortir en cas de litige ou d'audits pendant une **DUA** (*durée d'utilité administrative*) et, enfin, d'archives définitives de valeur historique ou patrimoniale.

Si l'informatique en entreprise est depuis longtemps associée à un objectif de "**l'accès continu aux données**", force est de constater que cela ressemble plus à un vœu pieu qu'à la réalité, même si cela continue à être annoncé périodiquement. :

Face à ce constat, la **cloud Computing** a pour objectif d'apporter de la disponibilité constant des données, la circulation de l'information quasi instantanée quelle que soit la distance et les sécurités de ces dernières.

### 3.1.3 Précision, objectifs et résultat.

Les objectifs visés dans ce projet sont :

- Le respect des délais fixés ;
- Le respect des coûts fixés ;
- La sécurisation des échanges ;
- La simplicité, l'ergonomie du système mise en place.

Ce projet a pour objectif principal de fournir une solution simple, d'utilisation économique, permettant la disponibilité et la sécurité des données des différents système de gestion au sein d'une entreprise par le biais d'un système de cloud computing.

### 3.1.4 Planification.

Pour mener ce projet à terme, il nous a fallu travailler en équipe et collaboration avec plusieurs autres personnes. Nous avons donc effectué un découpage des tâches, bien entendu plusieurs tâches pouvant s'exécuter simultanément, et d'autres pas. Selon les outils d'analyse utilisé, échelonner, permettra de mieux suivre l'avancer du projet.

Tableau 3.1 – Phase du Projet

Démarche	Branches	Durée
Etat de L'art de cloud Computing	Branche fonctionnelle	6 jours
Étude Comparative et choix de la solution	Branche fonctionnelle	14 jours
Analyse et Spécification des besoins	Branche fonctionnelle	30 jours
Spécifications techniques	Branche technique	24 jours
Conception	Branche réalisation	17 jours
Implémentation	Branche réalisation	25 jours
Tests	Branche réalisation	18 jours

Ici nous représentons les taches dans un diagramme de Gantt

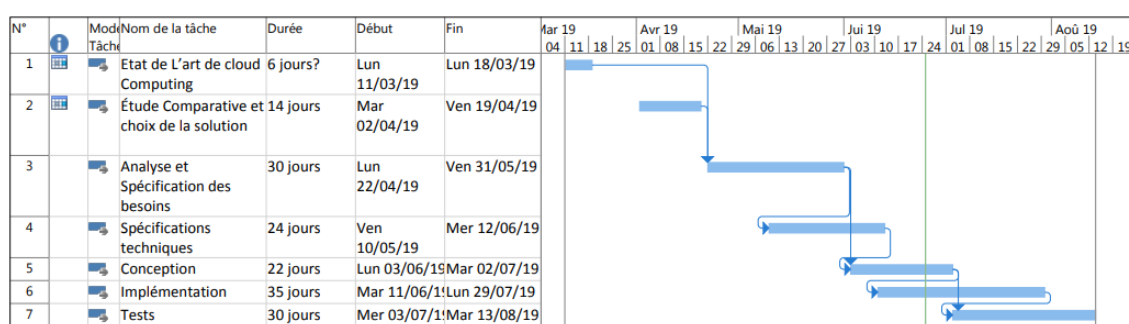


FIGURE 3.1 – Diagramme de Gantt

### 3.1.5 Estimation des ressources matérielles et financières.

Dans ce tableau, nous présentons les ressources matérielles et financières nécessaire pour le déploiement de la solution.

Tableau 3.2 – Estimation financière des ressources matériels

Désignations	Caractéristiques	Quantité	Coût unitaire.	Coût Total
Routeur mikrotik	RB2011UiAS-2HnD- : 128MB RAM, 5 ports 10/100 Mbit FE, 5 ports 10/100/1000 Mbit GE	2	210 000 FCFA	420 000 FCFA
Serveurs	Cœurs : 16 RAM : 128 Go Stockage : 2x30To 3 cartes réseaux.	4	950 000 FCFA	3 800 000 FCFA
Connexion inter-net	1Mbps	2	150 000 FCFA	300 000 FCFA
<b>TOTAL</b>				<b>4 520 000 FCFA</b>

## 3.2 Branche Fonctionnelle : Analyse et Spécification des Besoins

Cette partie décrit la branche fonctionnelle du 2TUP. .

### **3.2.1 Besoins fonctionnels**

Notre nuage s'adresse essentiellement à deux types d'utilisateurs : l'administrateur et les membres des projets. Cette première partie, sera pour énoncer et analyser les différents besoins fonctionnels et non fonctionnels du nuage.

Cette partie, est pour détailler l'ensemble des fonctionnalités que le nuage, à travers son portail, doit offrir aux utilisateurs. En effet, le système à réaliser doit répondre aux besoins fonctionnels suivants :

#### **3.2.1.1 Gestion d'images**

On parle d'images disques stockées par le service Glance. L'utilisateur pourrait consulter la liste des images autorisées pour les projets, les éditer. Aussi il sera possible de lancer des nouvelles instances de cette image, créer une nouvelle ou supprimer une existante.

#### **3.2.1.2 Gestion d'instances**

Une instance est une machine virtuelle en cours d'exécution ou dans un état connu comme «suspendue» qui peut être utilisé comme un serveur matériel. L'utilisateur pourrait consulter la liste d'instances des machines virtuelles actuelles plus quelques informations globales comme le projet auquel elles appartiennent, le serveur hôte, l'adresse IP, la taille, le statut et les actions en cours. Il aurait aussi les possibilités d'éditer, mettre fin, pause, redémarrer ou supprimer une instance. Aussi Il pourrait se connecter à la console VNC de l'instance ou créer une nouvelle.

#### **3.2.1.3 Gestion des volumes**

Le nuage permettrait à l'utilisateur de consulter la liste des volumes disques virtuels existants, la création d'un nouveau volume et la modification d'un ancien

#### **3.2.1.4 Gestion des flavors**

Un flavors est une configuration de matériel disponible dans un serveur. Chaque Flavor possède une combinaison unique d'espace disque et la capacité de mémoire. L'utilisateur pourrait consulter la liste des types d'instances disponibles, leurs spécifications en nombre de CPUs, mémoire, espace disque et créer des nouvelles définitions d'instance.

### **3.2.1.5 Gestion des projets**

Un projet est un groupement logique des utilisateurs au sein de Nova, utilisé pour définir les limites des ressources pour ce projet et l'accès aux images des machines virtuelles. Il serait possible de consulter les projets existants et leur statut et de créer des nouveaux projets.

### **3.2.1.6 Gestion des utilisateurs**

L'utilisateur aurait la possibilité de consulter la liste d'utilisateurs enregistrés, avec la possibilité d'ajouter ou d'éditer les détails mais pas d'ajouter l'utilisateur à plusieurs projets.

### **3.2.1.7 Gestion de la sécurité et de l'accès**

L'utilisateur pourrait consulter les adresses IP disponibles pour connecter les instances au réseau public avec la possibilité de création, les groupes de règles de pare-feu et leur interface d'édition et enfin la liste des clés SSH avec l'import ou la création de certificat.

## **3.2.2 Besoins non fonctionnels**

### **3.2.2.1 Simplicité d'un service à la demande**

Un utilisateur peut de manière unilatérale, immédiatement et généralement sans intervention humaine, avoir à sa disposition les ressources informatiques dont il a besoin (temps de calcul de serveurs, capacité de stockage, etc.).

### **3.2.2.2 Extrême flexibilité**

Les ressources mises à disposition ont une capacité d'adaptation forte et rapide à une demande d'évolution, généralement de manière transparente pour l'utilisateur.

### **3.2.2.3 Accès léger**

L'accès aux ressources ne nécessite pas d'équipement ou de logiciel propriétaire. Il se fait au travers d'applications facilement disponibles (parfois libres), généralement depuis un simple navigateur Internet.

### 3.2.2.4 Sûreté

Un évènement indésirable ne devrait pas se produire pendant l'accès d'une machine virtuelle aux ressources informatiques.

### 3.2.2.5 Vivacité

Une action souhaitée par une machine virtuelle arrivera nécessairement à être réalisée pour garantir la progression du programme.

## 3.2.3 Identification des acteurs. [17]

Cette phase a pour objectif de décrire le comportement attendu de l'application. Pour cela l'utilisation du diagramme des cas d'utilisation qui représente un élément essentiel de la modélisation orientée objet assure des bonnes résultats. Elle permet de modéliser les fonctionnalités de l'application de point de vue besoins utilisateur. Elle sert aussi à définir et à modéliser le produit à développer.

### 3.2.3.1 Les acteurs du système

Les acteurs qui manipuleront notre application sont :

- **L 'administrateur** : Il possède les droits administratifs qui lui permettrait de contrôler tout le nuage et lui permettrait d'accéder à l'interface d'administration sur le portail « Dashboard » ainsi qu'à tous les autres projets.
- **Membre du projet «Membre»** : C'est un membre d'un ou de plusieurs projets qui sont propres. Il n'aurait accès qu'à son (ses) projet(s).

Par la suite les principaux cas d'utilisations qui assurent toutes les tâches exécutées par le système seront mis en place. Avant l'accès au nuage, l'utilisateur (administrateur, membre) doit s'authentifier avec la saisie de son login, et son mot de passe. Après vérification, si l'utilisateur est accepté, il aura accès au nuage et selon son rôle des projets et des fonctionnalités s'activeront sinon on aura un message d'erreur.

### 3.2.3.2 Diagramme de cas d'utilisation Générale

Dans ce diagramme, nous présentons tous cas d'utilisations concernant l'administrateur

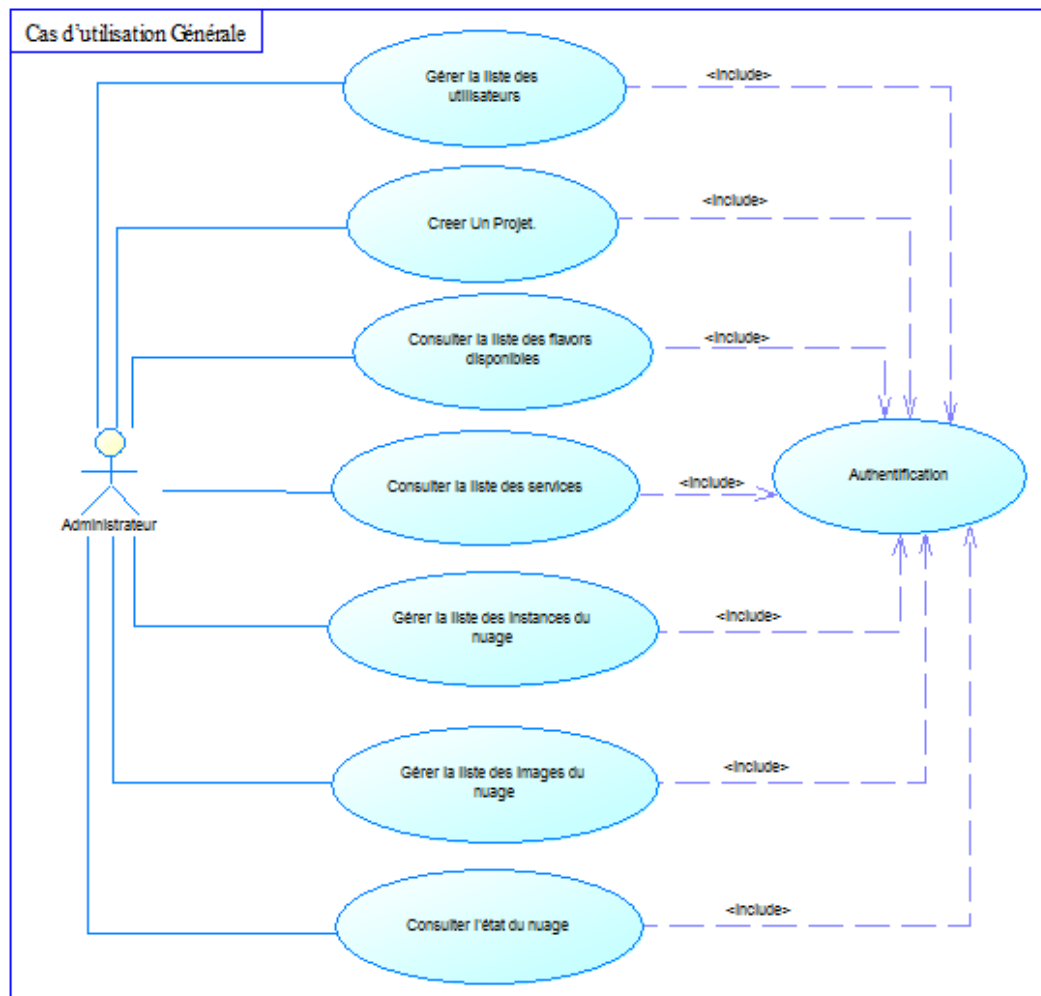


FIGURE 3.2 – Diagramme des cas d'utilisation Générale

### 3.2.4 Raffinements des Cas d'Utilisations

#### 3.2.4.1 Cas d'utilisation « Consulter l'état du nuage »

L'administrateur pourrait : -

- Consulter l'usage des serveurs par projet, utilisation actuelle en nombre de CPU virtuels, RAM et Disques puis compteur en CPU et espace disque (GB) par heures.
- Générer un rapport.



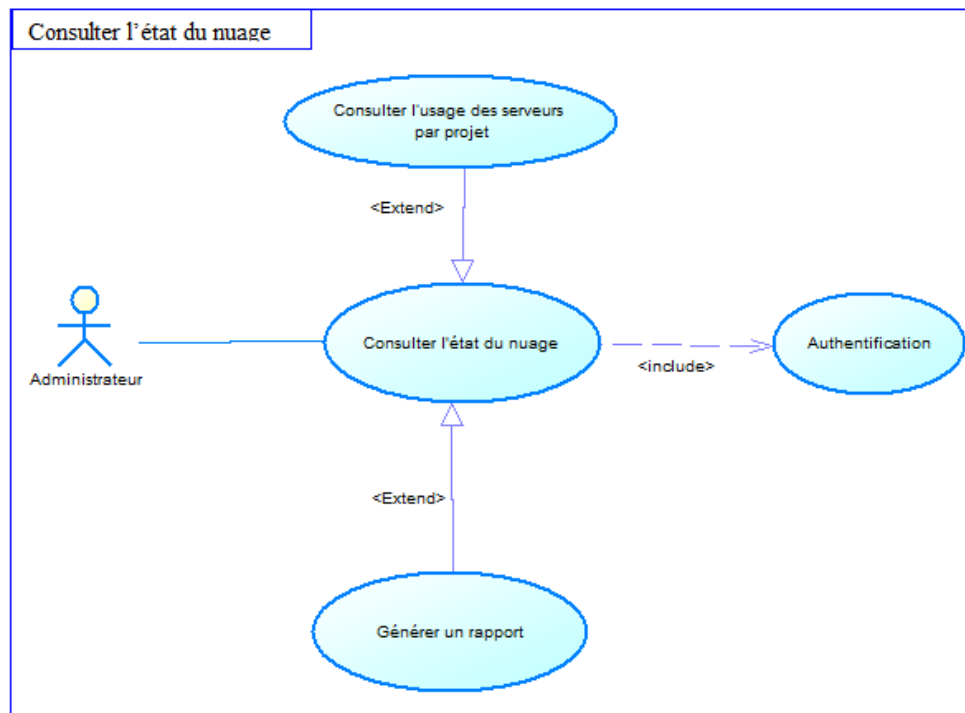


FIGURE 3.3 – Cas d'utilisation « Consulter l'état du nuage »

### 3.2.4.2 Cas d'utilisation « Gérer les instances »

L'administrateur pourrait gérer les instances existantes sur le nuage, il aurait la possibilité de : -

- Consulter la liste des instances existantes et leurs détails.
- Editer les détails d'une instance.
- Mettre fin à une instance.
- La suspendre.
- La redémarrer.
- La supprimer.
- Connecter à sa console VNC

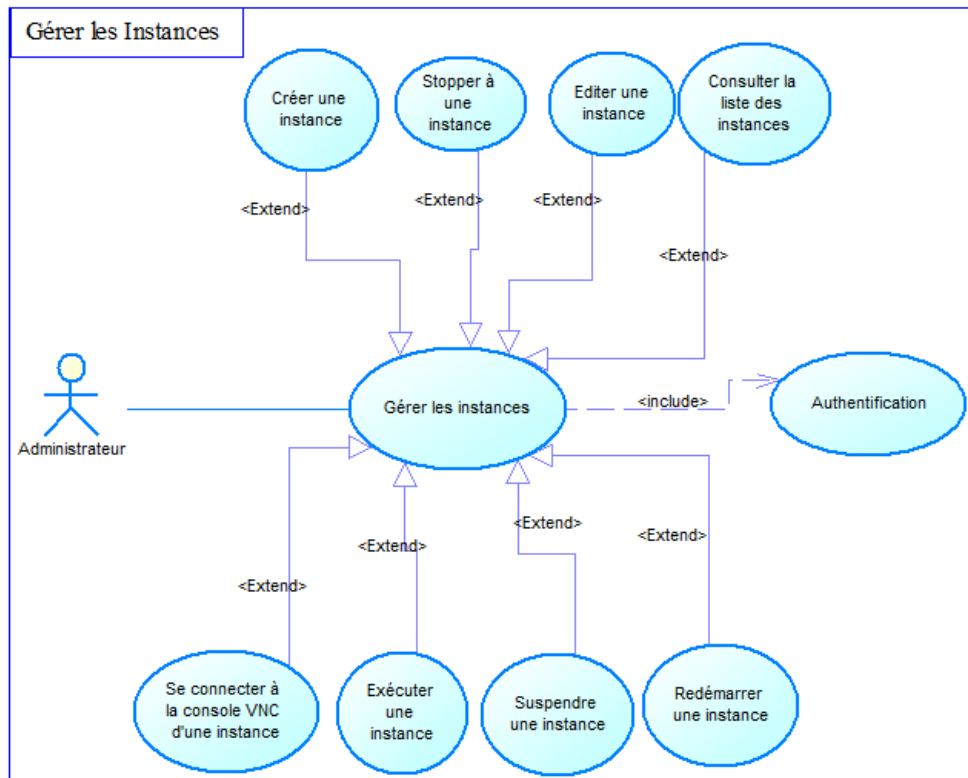


FIGURE 3.4 – Cas d'utilisation « Gérer les instances»

### 3.2.4.3 Cas d'utilisation « Gérer les services »

L'administrateur pourrait :

- Consulter la liste des services (Volume, Glance, Nova, Keystone..) activés,
- Modifier l'état des services (activé / désactivé)
- Gérer les serveurs hôte et leurs statut (activé/désactivé).

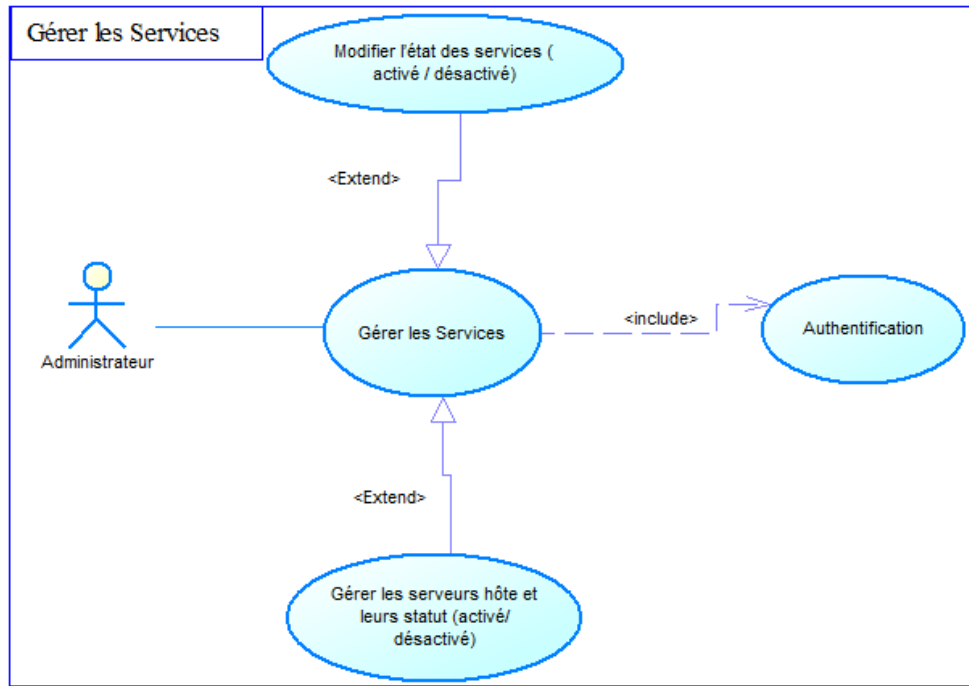


FIGURE 3.5 – Cas d'utilisation « Gérer les services »

#### 3.2.4.4 Cas d'utilisation « Gérer les Flavours »

L'administrateur pourrait :

- Consulter la liste de Flavours actuellement disponibles qui pourraient être utilisés pour lancer une instance.
- Créer des Flavours personnalisées.
- Editer des Flavours.
- Supprimer des Flavours existants.

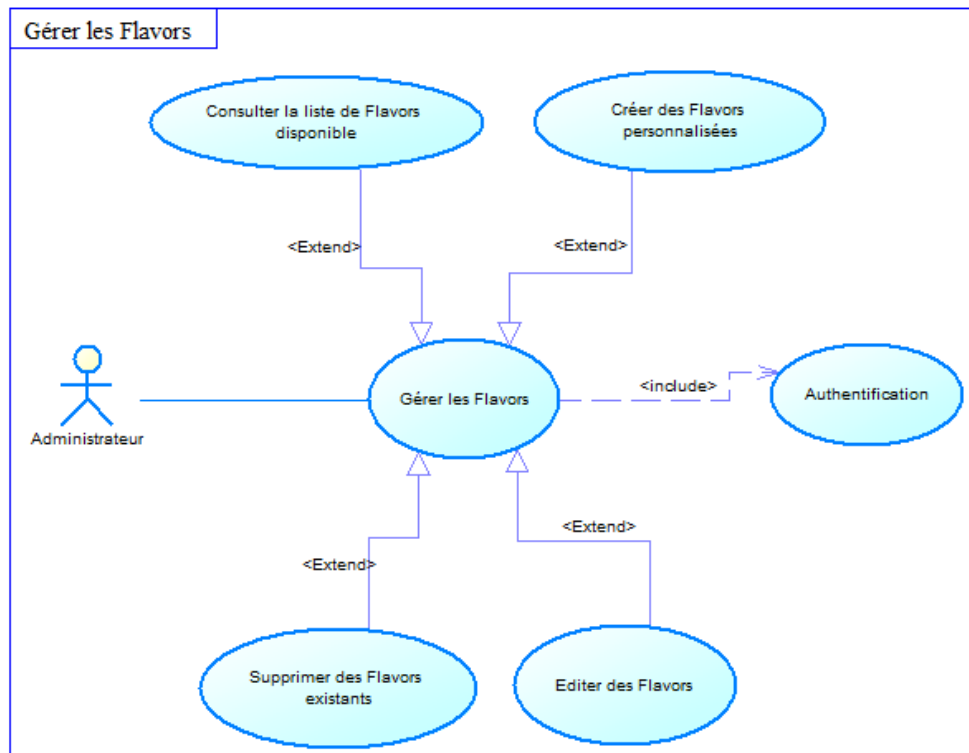


FIGURE 3.6 – Cas d'utilisation « Gérer les Flavors »

### 3.2.4.5 Cas d'utilisation « Gérer les Images »

L'administrateur aurait la possibilité de :

- Consulter la liste des images disponibles.
- Editer les détails d'une image (nom, noyau ID, Ramdisk ID, architecture, format, public ou privé).
- Supprimer des images si elles ne sont plus nécessaires.

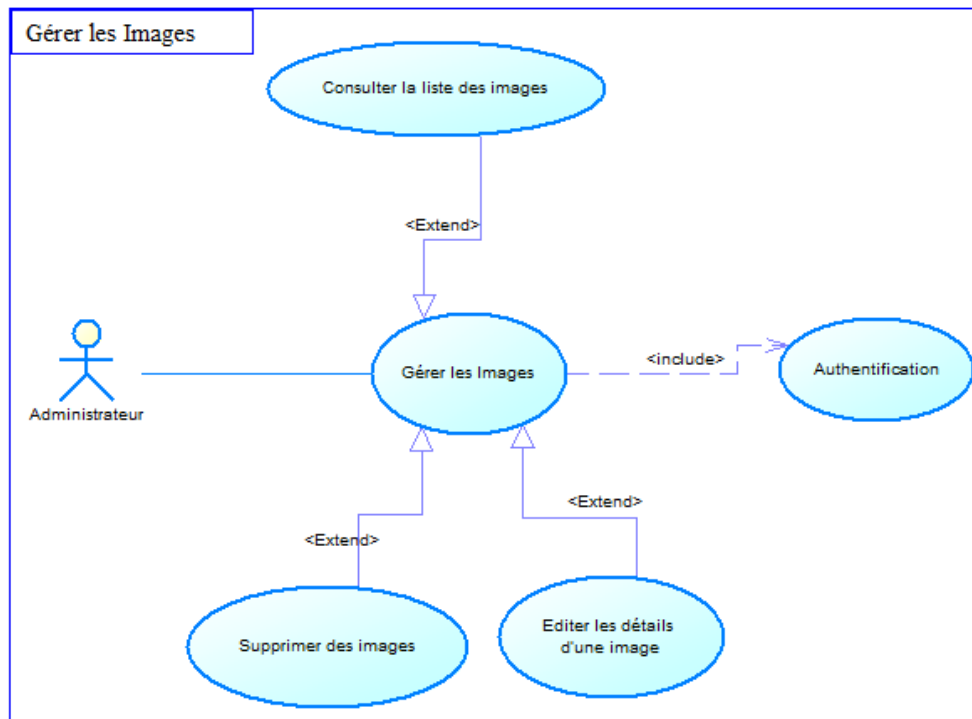


FIGURE 3.7 – Cas d'utilisation « Gérer les Images »

#### 3.2.4.6 Cas d'utilisation « Gérer les Projets »

L'administrateur aurait la possibilité de gérer les projets existants sur le nuage, ainsi il pourrait :

- Consulter la liste des projets disponibles (locataires) qui ont été créés, leurs détails et leurs utilisations.
- Créer des nouveaux projets.
- Affecter des utilisateurs à un projet.
- Modifier les détails d'un projet.
- Supprimer un projet.

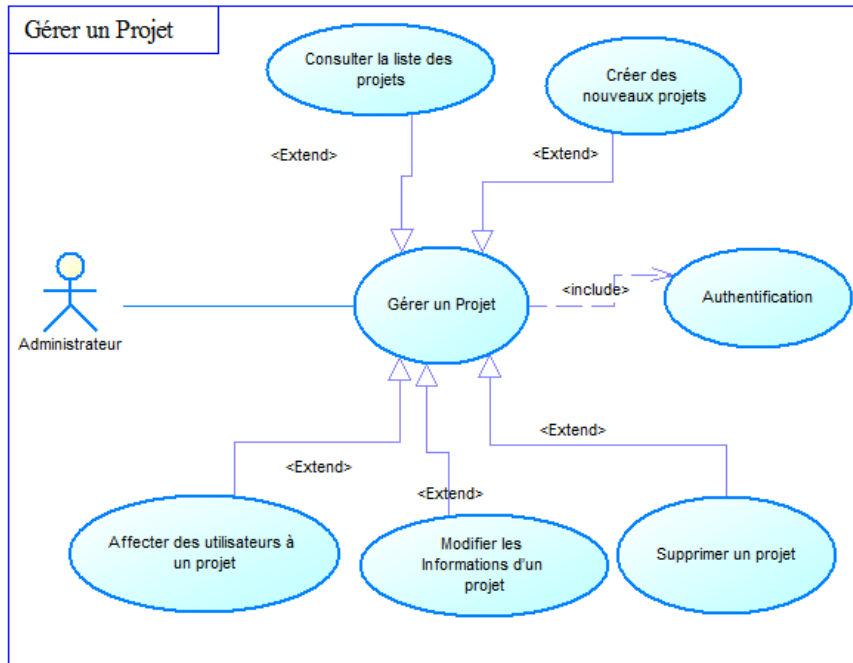


FIGURE 3.8 – Cas d'utilisation « Gérer les Projets »

### 3.2.4.7 Cas d'utilisation « Gérer les Utilisateurs »

L'administrateur pourrait gérer les comptes utilisateurs existants sur le nuage, ainsi il pourrait :

- Consulter la liste des utilisateurs qui ont été créés et leurs détails.
- Désactiver / Supprimer / Créer les utilisateurs.

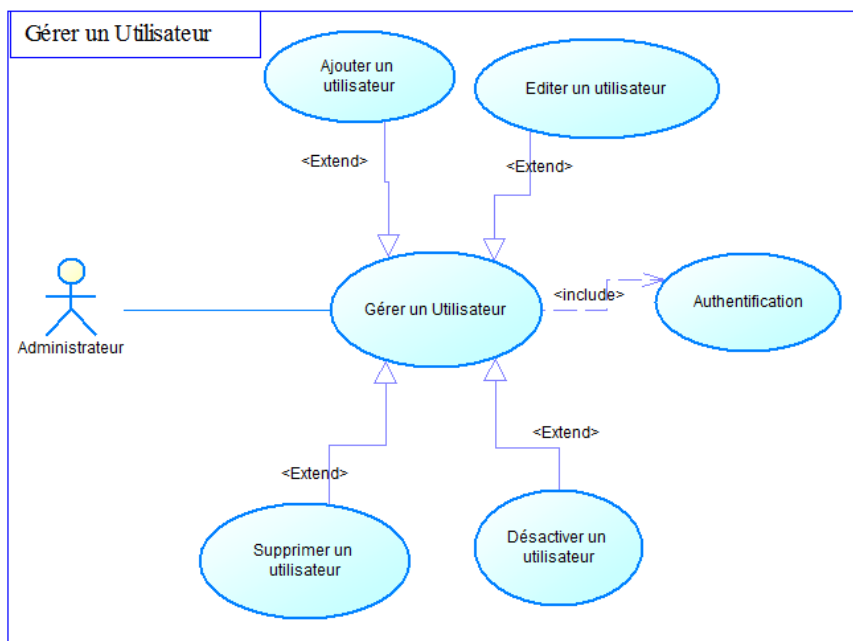


FIGURE 3.9 – Cas d'utilisation « Gérer les Utilisateurs »

### 3.2.4.8 Diagramme de cas d'utilisation « Membre d'un projet »

La figure ci-dessous présente le diagramme de cas d'utilisation d'un membre d'un projet existant dans le nuage.

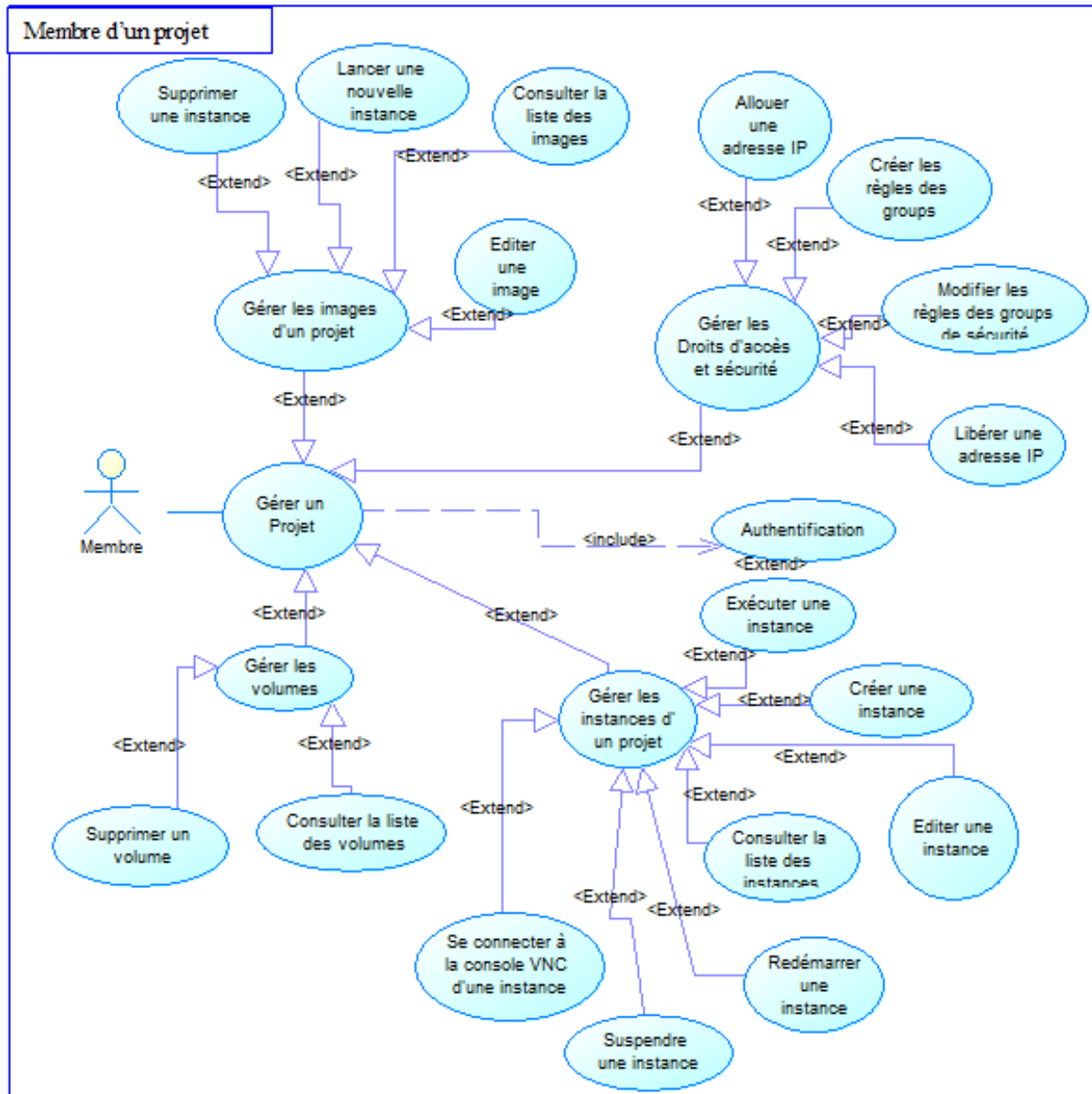


FIGURE 3.10 – Cas d'utilisation « Membre d'un projet »

Un membre ne pourrait consulter et gérer que les ressources des projets auxquels il appartient, ainsi il serait possible pour lui de :

**Consulter les états de ces projets :** Un membre pourrait consulter l'usage des serveurs d'un projet sélectionné, utilisation actuelle en nombre de CPU virtuels, RAM et Disques puis compteur en CPU et espace disque (GB) par heures

**Gérer les instances et les volumes :** Un membre pourrait consulter la liste des instances existantes et aurait la possibilité de les éditer, de créer ou de modifier des volumes disques virtuels.

**Gérer les images et leurs instances :** Un membre aurait la possibilité de consulter la liste des images autorisées pour le projet et lancer de nouvelles instances.

**Gérer la sécurité et l'accès :** Un membre pourrait consulter la liste des adresses IP disponibles pour connecter les instances au réseau public avec la possibilité de création des groupes de règles et de Pare-feu.

Il aurait aussi la possibilité de consulter la liste des clés SSH et de créer de certificat.

### **3.2.5 Diagrammes d'activité**

Les diagrammes d'activité système, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multithreads ou multi-processus). Le diagramme d'activité est également utilisé pour décrire un flux de travail (workflow).

Ce paragraphe, sera consacré pour présenter quelques diagrammes d'activités les plus significatifs.

#### **3.2.5.1 Diagrammes d'activité globale**

La figure ci-dessous montre le diagramme d'activité globale : les scénarios de quelques cas.



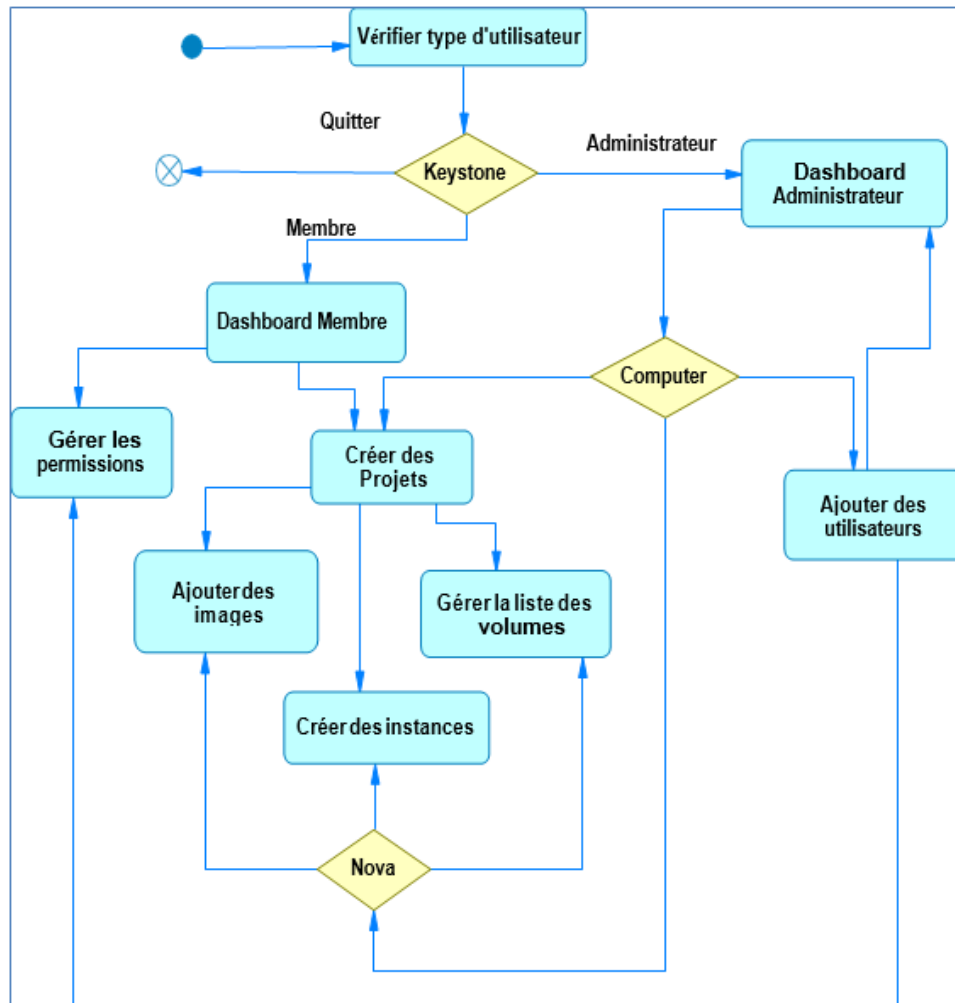


FIGURE 3.11 – Diagramme d'activité globale

### 3.2.5.2 Diagrammes d'activité « Créer une instance »

La figure ci-dessous montre les étapes à suivre pour construire une instance « machine virtuelle »

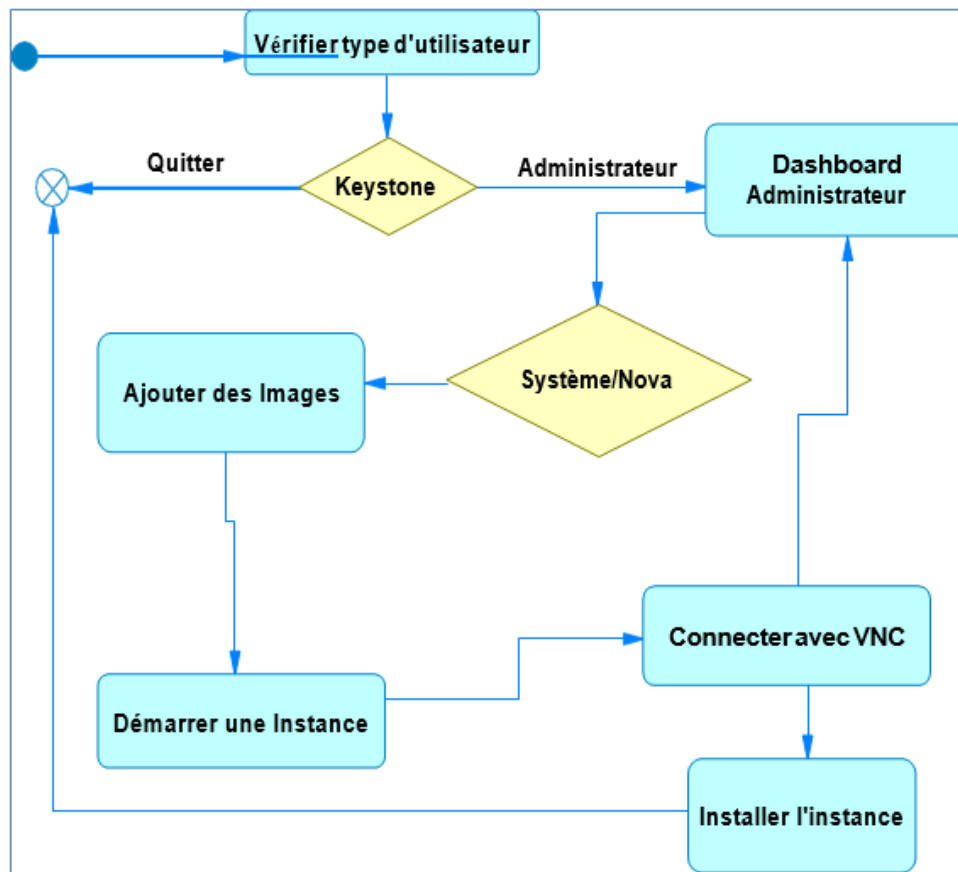


FIGURE 3.12 – Diagramme d'activité « créer une instance »

### 3.3 Branche Techniques : Environnement Matériel et Logiciel

#### 3.3.1 Architecture Physique

L'infrastructure physique du Cloud est un assemblage de serveurs, d'espaces de stockage et de composants réseau organisés de manière à permettre une croissance supérieure à celle que l'on obtient avec les infrastructures classiques. Ces composants doivent être sélectionnés pour leur capacité à répondre aux exigences d'extensibilité, d'efficacité, de robustesse et de sécurité.

La couche IaaS du Cloud Computing comprend trois parties essentielles :

- **Partie réseau** qui regroupe des routeurs, des switchs et des firwalls.
- **Partie stockage** SAN qui comprend principalement des baies.
- **Partie compute** qui est constituée des châssis regroupant des serveurs blades

### 3.3.2 Partie de Stockage

Le SAN est une technologie de stockage en réseau qui fournit l'espace disque rapide et fiable. C'est un réseau physique en fibre optique, il connecte l'ensemble des unités de stockages et des serveurs. Dans ce réseau, les données stockées sont routées et structurées via des commutateurs FC. Cette technologie est basée sur le protocole Fibre Channel, qui autorise le transfert de données entre périphériques sans surcharger les serveurs.

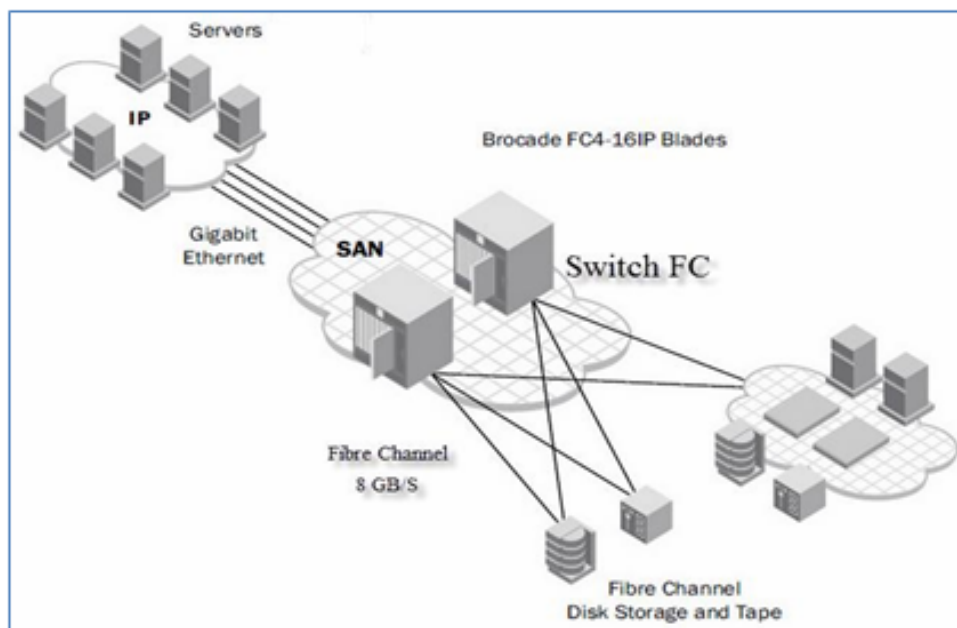


FIGURE 3.13 – Architecture-SAN "Storage Area Network"

### 3.3.3 Baie de stockage

Une baie de stockage est un équipement de sauvegarde de données informatiques qui comporte principalement un ensemble de disques, permettent d'emmagasiner et de gérer de grandes quantités de données généralement à travers un réseau de stockage dite SAN.

Les baies de stockage utilisent différentes techniques d'agrégat de disques, nommées RAID qui gèrent la cohérence et la répartition des données sur plusieurs disques durs. Les disques qui existent sur le marché sont : FC, SATA, SAS mais le meilleur c'est le FC.

Les baies utilisent aussi des protocoles de stockage comme iSCSI ou FC. Mais ce dernier est le plus performant et il peut aller jusqu'à 10GB/s.

### 3.3.4 Serveur

Le châssis est un équipement qui héberge un ensemble de serveurs lames et fournit une source d'alimentation électrique unique pour ces serveurs en mutualisant plusieurs unités d'alimentations électriques, assurant ainsi une redondance et permettant une tolérance aux pannes. Les connexions réseau sont incluses dans le châssis. Cela permet de connecter un serveur lame à différents supports physiques (paire torsadée ou fibre optique) et de mettre en place des configurations avancées (agrégation de ports).

La figure suivante illustre un exemple d'un châssis :



FIGURE 3.14 – Exemple d'un châssis

### 3.3.5 Serveurs blades

Un serveur lame ou blade est un serveur de la taille d'une carte d'extension PCI, intégrant processeur, mémoire vive, interface réseau et disque dur, dont la compacité simplifie la gestion de l'espace, économise la consommation d'énergie, et autorise l'installation d'un grand nombre de serveurs. Tenant sur une simple carte PCI, le serveur lame permet de ranger dans un seul châssis des dizaines de serveurs. Chaque lame est un serveur à part entière, souvent dédié à une seule application.

- En effet, chaque lame a six connecteurs Réseaux (3 carte bi-port) : Une carte pour l'administration des blades : une path sur ETH 1 et l'autre sur ETH 2 de châssis.
- Une carte pour le LAN : une path sur ETH 3 et l'autre sur ETH 4.
- Une Carte pour le stockage : une path sur FC 1 et l'autre sur FC 2.

Nous notons que la couche de virtualisation s'installe sur les serveurs blades, tel que chaque blade héberge un certain nombre de machines virtuelles.

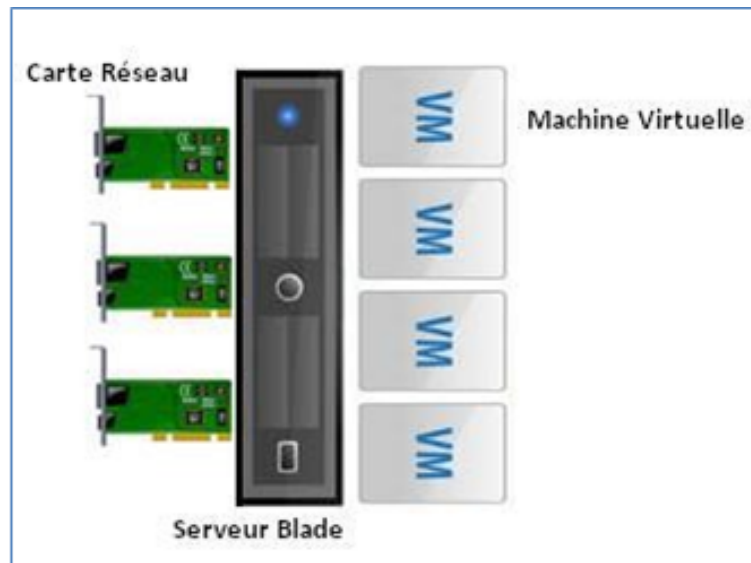


FIGURE 3.15 – Représentation d'un serveur blade

Lorsqu'il s'agit d'une architecture IaaS, il y'a trois parties primordiales que nous devons respecter lors de la conception de notre propre architecture, à savoir :

- **Partie réseau** qui regroupe des routeurs, des switchs et des firwalls.
- **Partie stockage SAN** qui comprend des baies.
- **Partie compute** qui est constituée des châssis regroupant des serveurs blades.

En outre, nous devons respecter également trois critères critiques : redondance, (disponibilité de service), performances et sécurité. Autrement dit, il faut concevoir une architecture hautement sécurisée qui assure toujours une disponibilité de service 24/24 ,7/7 et une redondance physique de sorte que les ressources qui sont requises pour le calcul, le réseau et le stockage demeurent disponibles et les données qui sont stockées dans le Cloud IaaS peuvent être récupérées facilement en cas de défaillance matérielle.

### 3.4 Architecture Réseau.[18]

L'architecture réseau dans laquelle notre solution est déployée est représentée par la figure 29. Cette architecture est composée de :

- Un serveur OpenStack dans lequel est déployé OpenStack et qui a comme rôle la gestion du nuage.
- Serveurs reliés entre eux par un Switch.
- Les postes de développeurs de la Société.
- Une instance de réseau privé de la société destiné aux communications entre les VMs

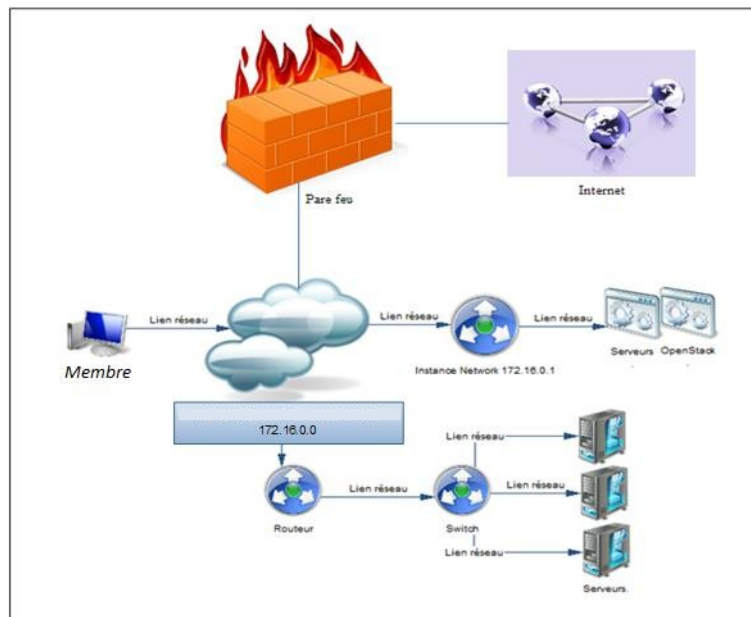


FIGURE 3.16 – Architecture réseau

### 3.5 Diagramme de déploiement du système.

La figure ci dessous illustre le diagramme de déploiement du système. Notre nuage est composé principalement d'un serveur OpenStack : ce serveur représente à la fois le contrôleur du nuage qui exécute les services Glance, Swift, Cinder, Keystone et Nova et il représente le nœud de calcul qui fonctionne Nova et l'hyperviseur KVM.

Le nuage est composé aussi d'un nœud Pc d'utilisateur (administrateur, membre de projet) puisque un ordinateur client est nécessaire pour regrouper les images d'interfaçage avec les serveurs, un nœud Datacenter qui supervise le serveur OpenStack par les images et les volumes de stockage.

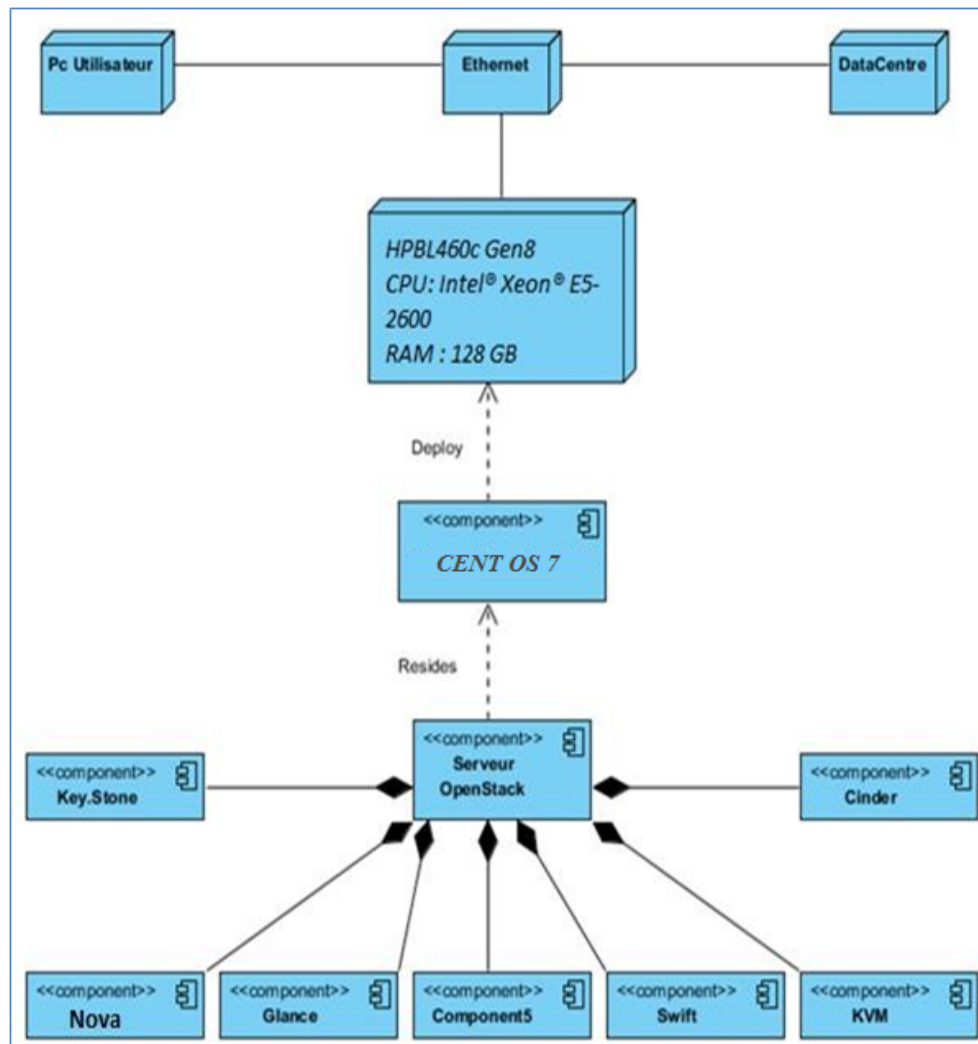


FIGURE 3.17 – Diagramme de déploiement du système

### 3.6 Environnement Logiciel. [15]

Le tableau ci-dessous permet de donner une idée sur l'environnement logiciel utilisé pendant l'implémentation de l'outil d'administration de la plateforme Cloud Computing :

Tableau 3.3 – Configuration logiciel

Shell	Toutes les commandes sont rédigées en terminal.
Openstack Devstack	Framework qui permet le développement d'un Cloud privé (IaaS).
My SQL	Système de gestion de bases de données installé et utilisé lors d'installation de tous les composants d'OpenStack.
<ul style="list-style-type: none"> <li>• PowerAMC</li> <li>• Visual Paradigm for UML</li> <li>• PyCharm</li> </ul>	Gestion de besoin et Conception du nuage privé

Cette partie est pour exposer les différentes phases de réalisation illustrées par la figure ci dessous  
En effet, notre travail est composé de trois étapes :

- **Planification** : Écrire le scénario de déploiement, finaliser les choix d'architectures, et s'assurer que le matériel requis soit disponible.
- **Déploiement** : Installer les composants d'OpenStack, et enfin les configurer.
- **Utilisation et test** : Utiliser OpenStack afin d'accueillir les utilisateurs finaux.

Ces étapes seront détaillées dans la suite

### 3.6.1 Planification du déploiement d'OpenStack

Il existe de nombreuses méthodes pour le déploiement d'OpenStack.

- **Nœud unique** : un seul serveur exécute tous les services nova et également conduit toutes les instances virtuelles.
- **Deux nœuds** : Un nœud de contrôleur nuage exécute les services nova à l'exception de nova-compute, et un nœud de calcul fonctionne nova-compute
- **Plusieurs nœuds** : Un minimum de quatre nœuds est le meilleur pour l'exécution de plusieurs instances virtuelles qui nécessitent beaucoup de puissance de traitement.



### 3.7 Architecture de Solution Openstack.[13]

Trois éléments interagissent avec tous les composants du système. Horizon est l'interface graphique que les administrateurs peuvent plus facilement utiliser pour gérer tous les projets. Keystone gère la gestion des utilisateurs autorisés, et Neutron (Quantum) définit les réseaux qui fournissent une connectivité entre les composants.

Nova peut sans doute être considérée comme l'OpenStack de base. Il gère l'orchestration des charges de travail. Ses instances de calcul nécessitent généralement une certaine forme de stockage persistant qui peut être soit à base de blocs (Cinder) ou orienté objet (Swift). Nouvelle nécessite également une image pour lancer une instance. Regards gère cette demande, de sorte qu'il peut éventuellement utiliser Swift que leur stockage back-end.

L'architecture OpenStack avait cherché à faire de chaque projet aussi indépendante que possible, ce qui donne aux utilisateurs la possibilité de déployer un sous-ensemble de la fonctionnalité et de l'intégrer avec d'autres systèmes et technologies qui offrent des fonctions similaires ou complémentaires. Néanmoins, cette indépendance ne doit pas masquer le fait que d'un cloud privé entièrement fonctionnel est susceptible de nécessiter pratiquement toutes les fonctionnalités de fonctionner sans heurts, et les éléments devra être étroitement intégré.

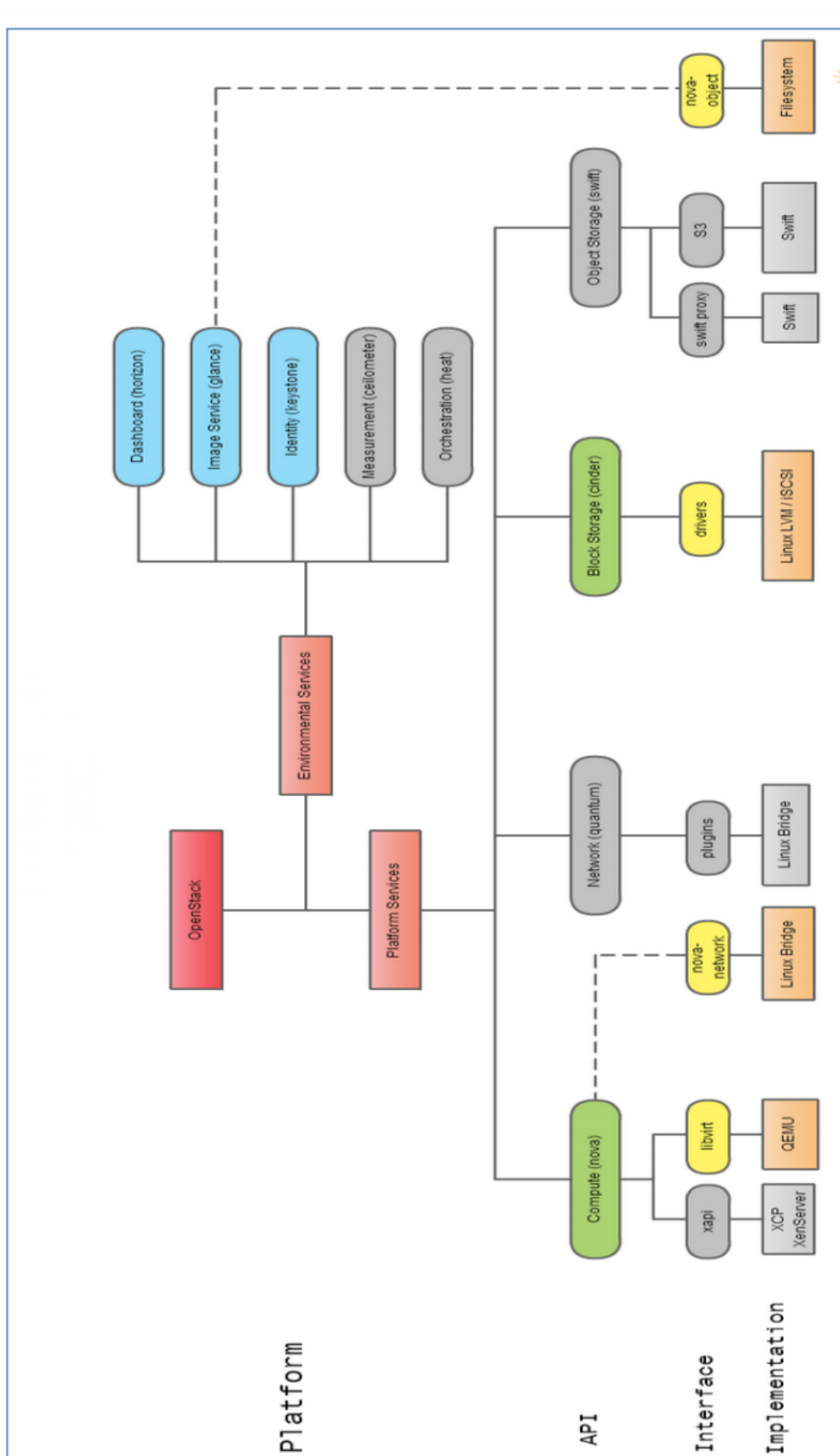


FIGURE 3.18 – Architecture Générale OpenStack

### 3.8 Diagramme de Déploiement

Un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.

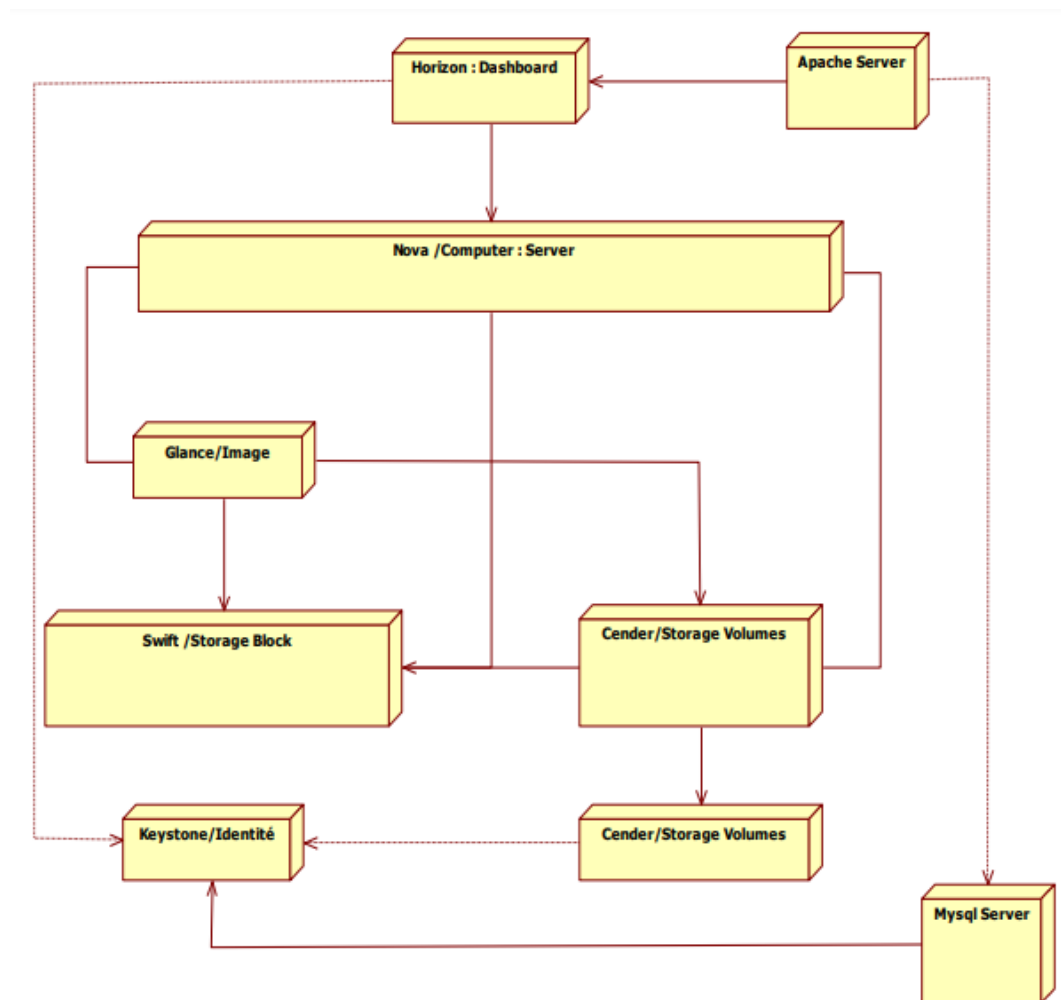


FIGURE 3.19 – Diagramme de déploiement OpenStack

#### 3.8.1 Diagrammes de séquences

Avec les diagrammes de séquences système, l'UML fournit un moyen graphique pour représenter les interactions entre un acteur et le système au cours de l'exécution du cas d'utilisation.

Ce paragraphe, sera consacré pour présenter quelques diagrammes de séquences les plus significatifs.

### 3.8.2 Diagrammes de séquences globales

La figure ci dessous montre le diagramme système globale de quelque cas d'utilisation : les scénarios de quelques cas.

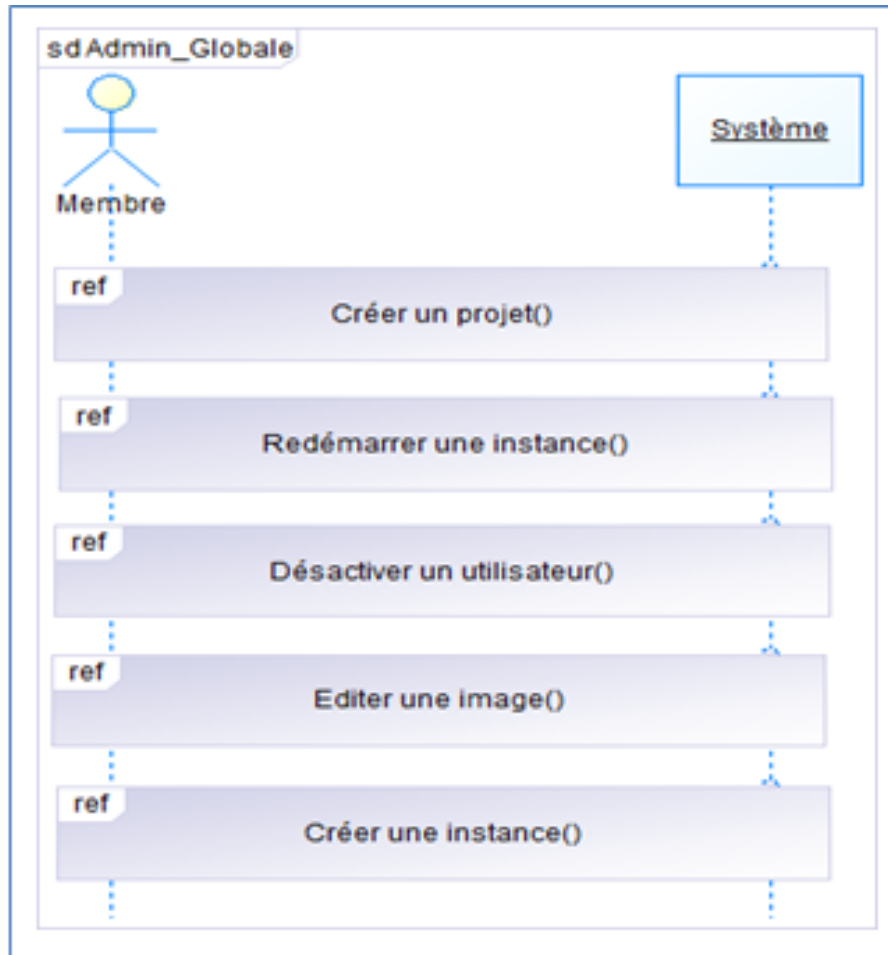


FIGURE 3.20 – Diagramme de séquences globale

### 3.8.3 Diagramme de séquences « scénario d'authentification »

**Acteur :** Un membre.

**Pré conditions :** Le membre, doit avoir un compte valide dans le système.

**Déclencheur :** Un membre veut consulter l'état du projet auquel il appartient.

**Description :** Ce cas d'utilisation permet à un membre du projet de s'identifier pour accéder au nuage à travers le Dashboard.

Scénario principal :

- Un membre accède au Dashboard.

- Une interface d'authentification s'affiche.
- Le membre entre ses données (login, mot de passe) et tape le bouton « valider ».
- Les différents services propres au nuage s'affichent

Scénario alternatif :

- Les données saisies sont erronées.
- Un message d'erreur s'affiche.

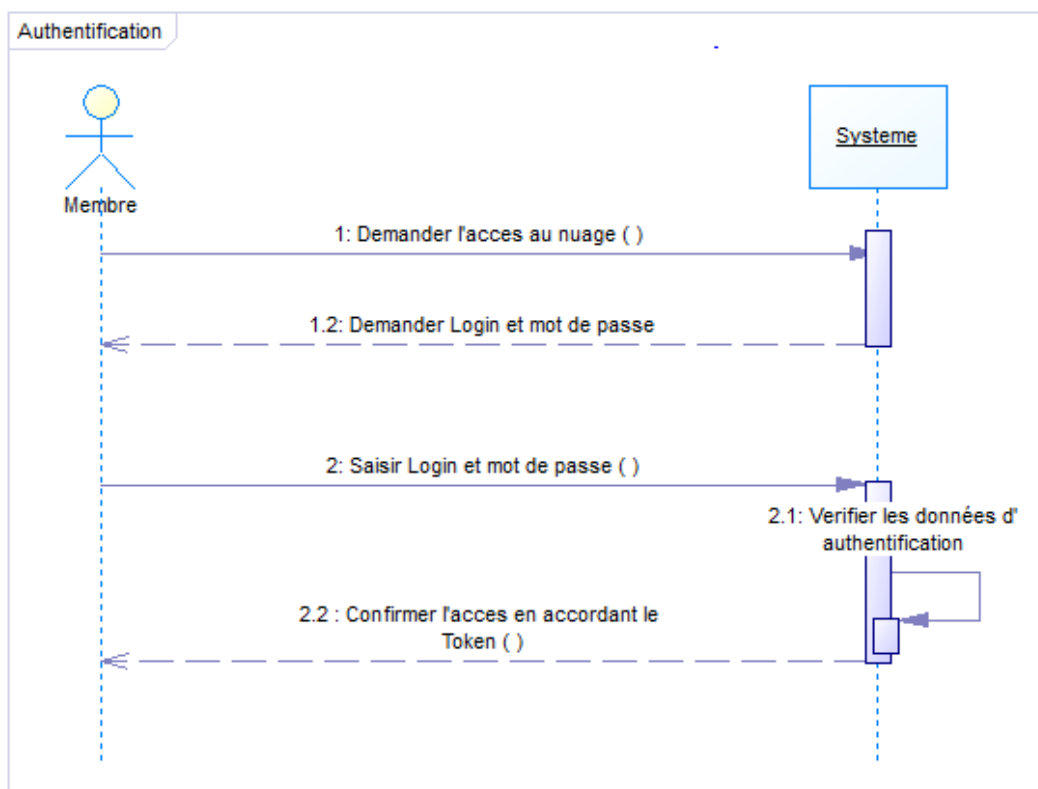


FIGURE 3.21 – Scénario d'authentification

### 3.8.4 Diagramme de séquences « Créer un projet »

**Acteur :** Administrateur.

**Pré conditions :** L'administrateur a passé l'étape d'authentification avec succès.

**Déclencheur :** Un user veut créer un nouveau projet.

**Description :** Ce cas d'utilisation permet à l'administrateur de créer un nouveau projet dans le nuage et l'accorder aux utilisateurs.

Scénario principal :

- Connecter au nuage en tant qu'administrateur.
- Sélectionner le lien Projets dans le menu.
- Choisir de créer un nouveau projet en cliquant sur un bouton « créer ».
- Remplir tous les champs et affecter des membres au projet.
- Valider l'opération.
- Le nouveau projet est ajouté à la liste des projets

Scénario alternatif :

- Des champs obligatoires ne sont pas été remplis
- Un message d'erreur s'affiche.

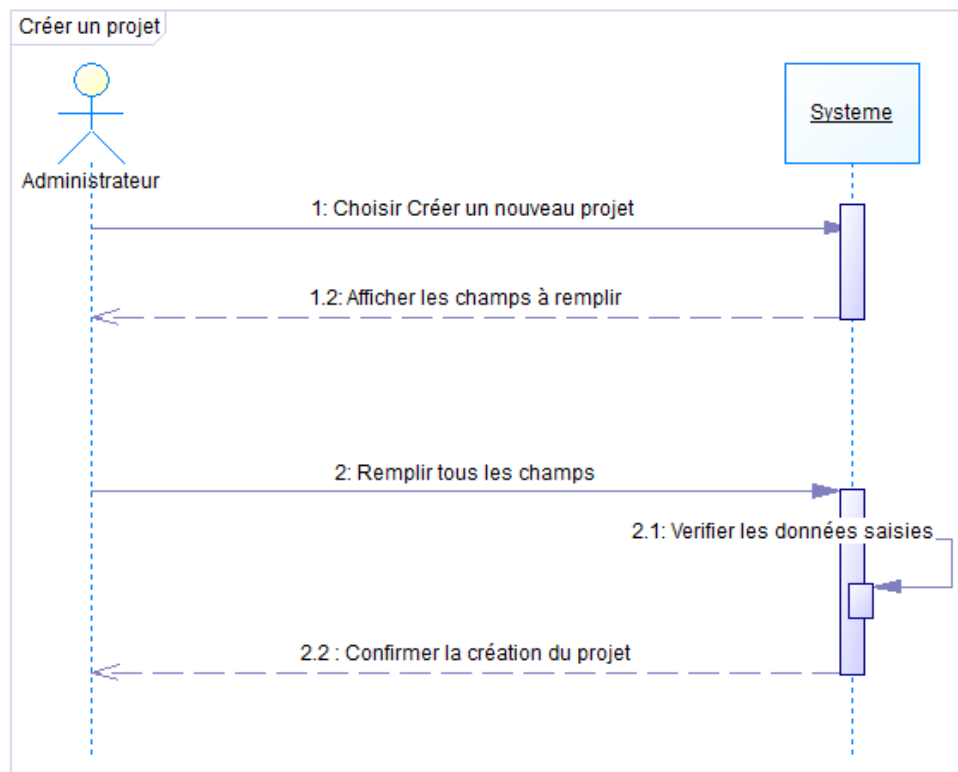


FIGURE 3.22 – Diagramme de séquences « créer un projet »

### 3.8.5 Diagramme de séquences « scenario création d'une instance »

**Acteur :** Membre.

**Pré conditions :** Le membre a passé l'étape d'authentification avec succès.

**Déclencheur :** Un membre d'un projet veut créer une nouvelle instance d'une machine virtuelle déjà existante dans le nuage.

**Description :** Ce cas d'utilisation permet à un membre d'un projet de créer une nouvelle instance d'une machine virtuelle du nuage.

Scénario principal :

- Connecter au nuage en tant que membre.
- Choisir le projet dans lequel la machine existe.
- Sélectionner le lien « Images et snapshot » dans le menu.
- Choisir la machine cible et choisir de créer une nouvelle instance.
- Remplir tous les champs.
- Valider l'opération.

Une nouvelle instance est ajoutée à la liste des instances.

Scénario alternatif :

- Des champs obligatoires ne sont pas été remplis.
- Un message d'erreur s'affiche.

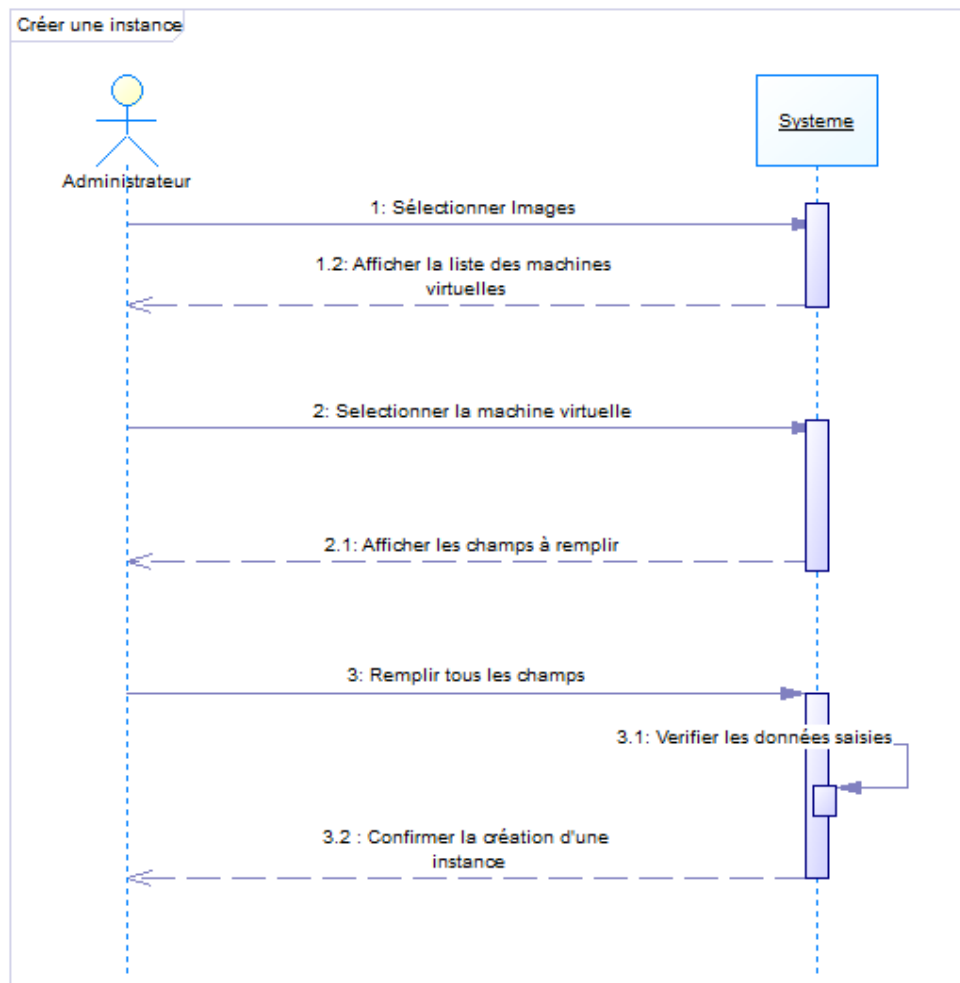


FIGURE 3.23 – Diagramme de séquences « Créer une instance »

### 3.8.6 Diagramme de séquences d'entités globales

La figure ci-dessous montre le diagramme de séquences d'entités globales d'administrateur pour trois cas d'utilisation.



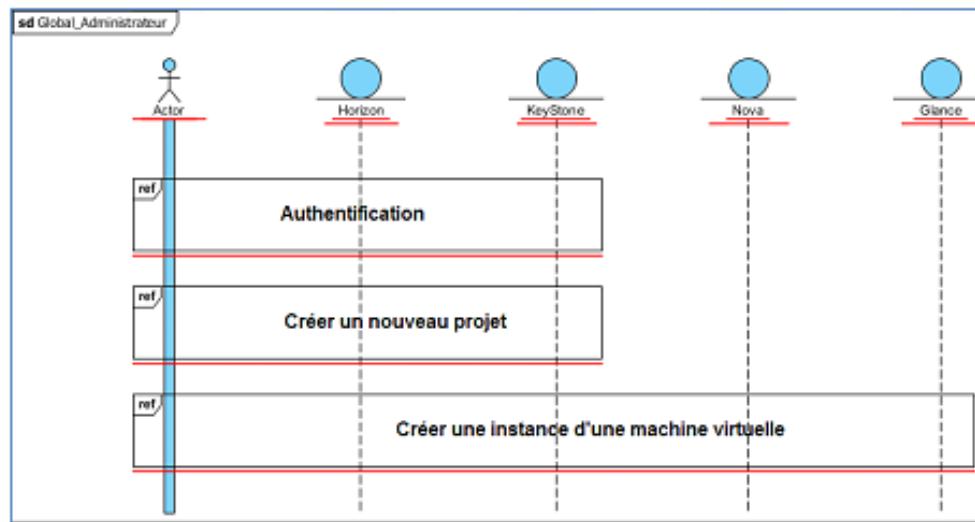


FIGURE 3.24 – Diagramme de séquences d'entités globales

### 3.8.7 Diagramme de séquences entité « authentification »

Pour accéder aux différents services du nuage, l'utilisateur (administrateur, membre d'un projet) doit s'identifier. Ainsi après la saisie de ses informations (nom d'utilisateur / mot de passe) dans l'interface d'authentification d'Horizon, le service d'Identity de Keystone vérifie les données d'utilisateur en lui accordant un Token. Selon ce Token une liste des projets s'affichera.

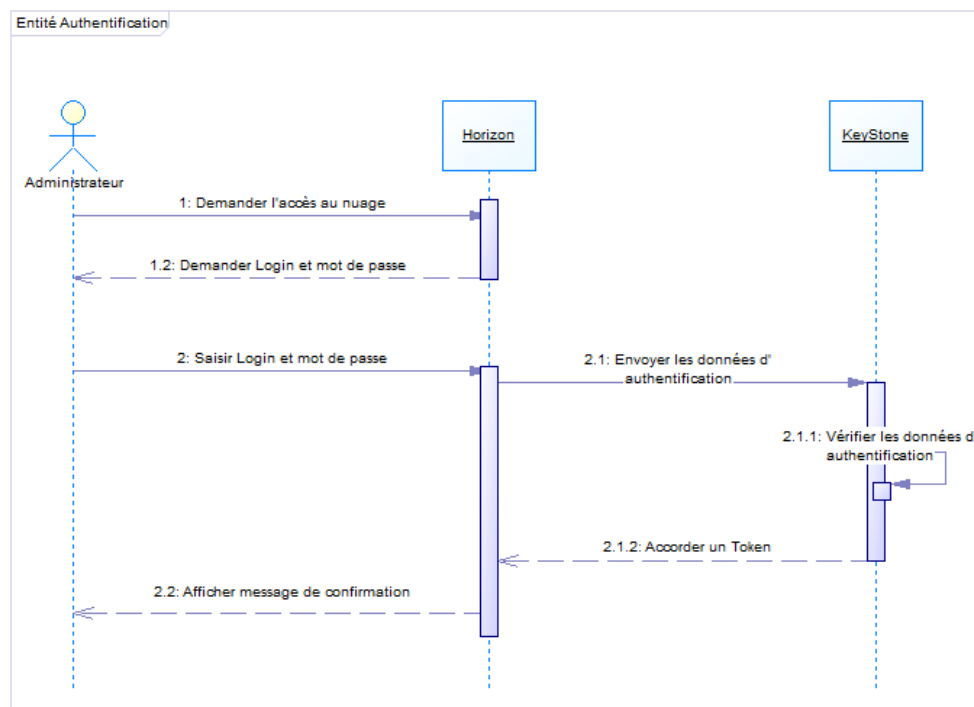


FIGURE 3.25 – Diagramme de séquences « authentification »

### 3.8.8 Diagramme de séquences entité « Créer un projet »

Si l'administrateur veut créer un nouveau projet (Tenant), il doit remplir tous les champs (Nom, description, les membres du projet . . .) qui s'affiche dans la fenêtre d'Horizon. Puis il confirme ses choix en cliquant sur un bouton nommé «Terminer». Keystone va vérifier les données et met à jour sa base des données en modifiant le tableau «Tenant».

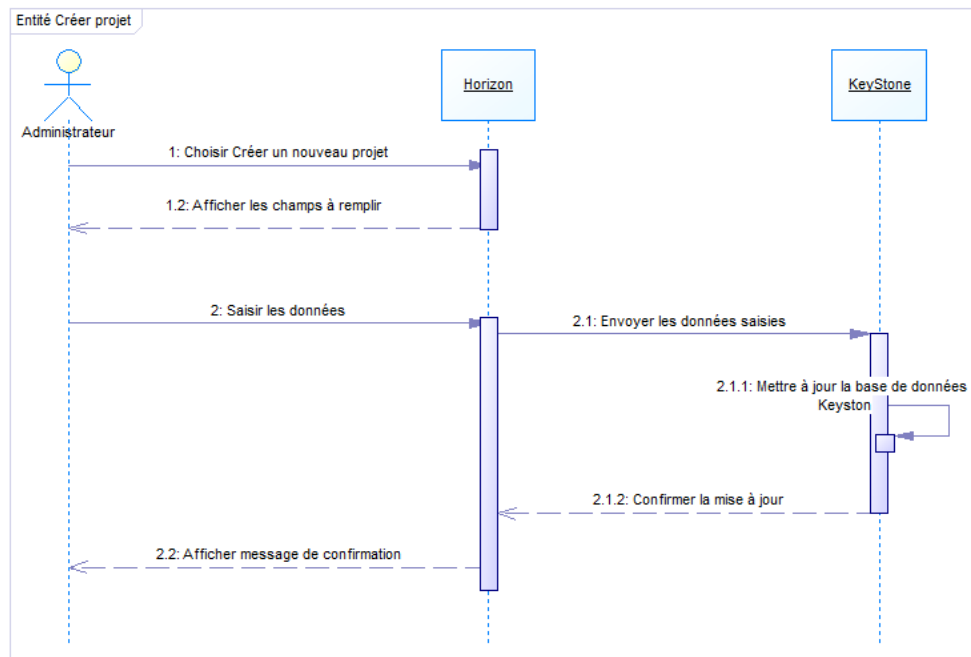


FIGURE 3.26 – Diagramme de séquences entité « Créer un projet »

### 3.8.9 Diagramme de séquences entité « créer une Instance »

L'utilisateur se connecte via un navigateur web sur horizon. En fonction de son profil, il a le droit ou non de créer une instance.

Cette création nécessite la communication d'Horizon avec Keystone en premier lieu et Keystone avec Glance et Nova en second lieu. Après la création de cette instance l'utilisateur peut lui accéder via SSH ou VNC.

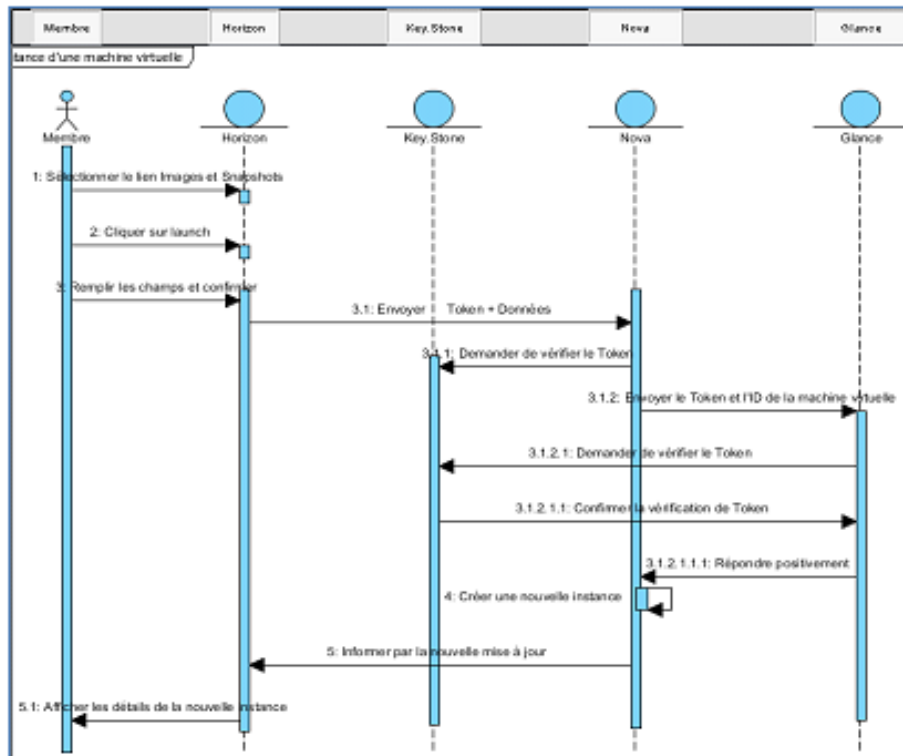


FIGURE 3.27 – Diagramme de séquences entité « créer une Instance »

A travers ce chapitre, l'exposition des besoins tant fonctionnels que non fonctionnels du projet est faite.

De plus, la présentation des diagrammes des cas d'utilisations et des diagrammes d'activités en définissant les acteurs et les détails des cas d'utilisations qui sont mis en place. Et les différents scénarios d'utilisation et interaction entre les différents composants de l'architecture Openstack.

Dans notre contexte, nous l'avons déployé sur un nœud simple. Dans les sections qui suivent, détaillent l'installation, la configuration ainsi que la méthode d'utilisation et de test d'OpenStack

## CHAPITRE 4

### IMPLEMENTATION – TEST - BILAN

Ce chapitre présente l’installation des composants d’identité, d’images et virtualisation sur une seule machine. Il s’agit plutôt d’une configuration de développement mais néanmoins fonctionnelle. OpenStack est un logiciel libre qui permet la construction de cloud privé et public. OpenStack 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 est composé d’une série de logiciels et de projets au code source libre qui sont maintenus par la communauté incluant : OpenStack Compute (nommé Nova), OpenStack Object Storage (nommé Swift), et OpenStack Image Service (nommé Glance).

## 4.1 Implémentation.

### 4.1.1 Les Composants.[16]

OpenStack possède une architecture modulaire qui comprend de nombreux composants. Voici la liste des composants intégrés à OpenStack.

- **Compute** : Nova (application)
- **Object Storage** : Swift (stockage d’objet)
- **Image Service** : Glance (service d’image)
- **Dashboard** : Horizon (interface Web de paramétrage et gestion)

- **Identity** : Keystone (gestion de l'identité)
- **Network** : Neutron (auparavant nommé Quantum) (gestion des réseaux à la demande)
- **Storage** : Cinder (service de disques persistants pour les machines virtuelles)

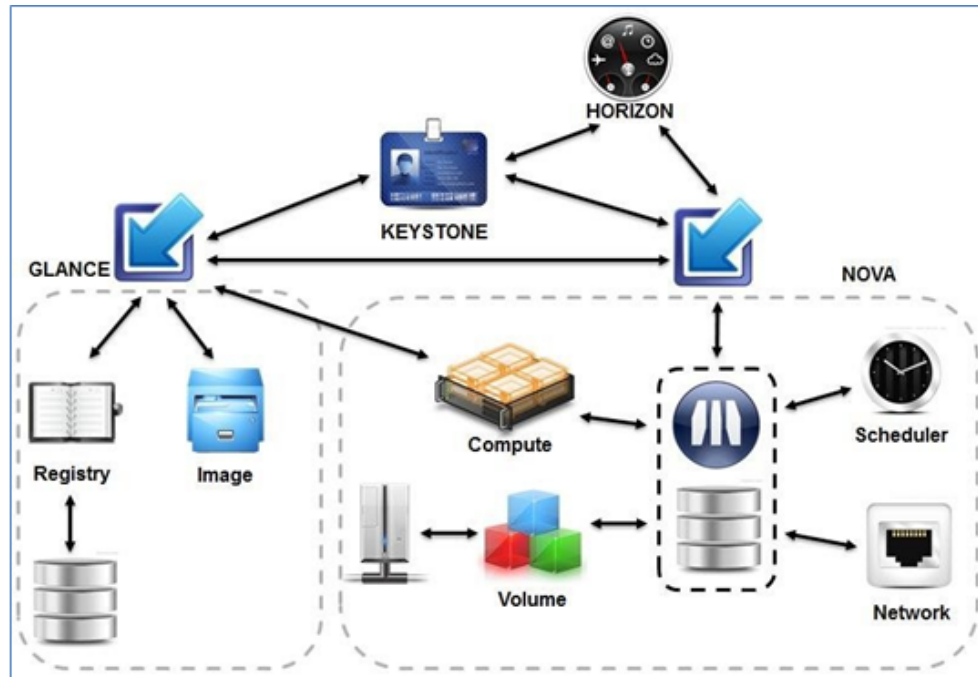


FIGURE 4.1 – Liste des Composants d'OpenStack

Plan de travail :

- Sélection des modules à installer
- Définition de l'architecture matérielle
- Déploiement du système d'exploitation sur les Os
- Installation et vérification des Prérequis
- Installation de keystone
- Installation de glance
- Installation de Nova
- Installation de Cinder
- Test

### 4.1.2 Prérequis

- Disposer des droits d'administration.
- Disposer d'une connexion à Internet configurée et activée.
- Avoir les dépôts d'activés
- Un processeur supportant la virtualisation matérielle (test sur la page KVM)
- Disposer d'un disque dur ou d'une partition non formatée pour LVM

### 4.1.3 Les paquets à installer

- kvm,libvirt-bin,virtinst.
- mysql-server,python-mysqldb
- bridge-utils
- Il est nécessaire de configurer le réseau en IP Fixe

Nous rappelons que pour les testes, tous les services OpenStack seront installés sur la même machine

### 4.1.4 Préparation du système

#### 4.1.4.1 Réseau

Modifiez avec les droits d'administration le fichier /etc/network/interfaces en ajoutant les adresses ip fixe (172.20.203.220).

Ajouter les interfaces réseaux de serveur DNS pour se connecter à l'internet.

#### 4.1.4.2 Serveur NTP

Le serveur NTP étant nécessaire à la bonne synchronisation du cloud, installez le paquet ntp avec la commande suivante :

```
#sudo apt - getinstallntp
```

#### 4.1.4.3 RabbitMQ

RabbitMQ est un courtier de messages se basant sur le standard AMQP afin d'échanger avec différents clients. C'est le service qui permet aux composants OpenStack de communiquer entre eux. Installez les paquets : *rabbitmq – server, memcached, python – memcache*

#### 4.1.4.4 Mysql

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

#### 4.1.4.5 Keystone

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

- **Gestion des utilisateurs :**

La gestion des utilisateurs s'articule autour de 3 objets :

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

- **Gestion des services et points d'accès**

La gestion des différents services, comme Glance pour les images ou Swift pour le stockage. La définition des points d'accès à ces différents services, les url et ports pour y accéder

#### 4.1.4.6 Préparation de la base de données Mysql

Ici, on commence par créer la base MySQL.

La commande suivante crée un utilisateur et sa base de données nommés "keystone". Changez SQL-PASSWD par un mot de passe root.

```
mysql – uroot – p << EOF
CREATEDATABASEkeystone;
GRANTALLPRIVILEGESONkeystone.*TO'keystone'@'%IDENTIFYBY'SQLPASSWD'
FLUSHPRIVILEGE;
OEF
```

#### **4.1.4.7 Installation**

L'ensemble des commandes nécessaire à l'installation sera mis a l'annexe du document.

## **4.2 Tests**

Cette étape permet de tester les différentes fonctionnalités attendues du nuage privé. Ainsi elle permet de lier l'étape de virtualisation aux autres étapes.

Elle consiste à :

- **Ajouter des projets.**
- **Ajouter des utilisateurs.**
- **Télécharger l'image test déjà virtualisé.**
- **Télécharger des volumes.**
- **Lancer des instances.**
- **Configurer les accès.**

### **4.2.1 Interface Authentification au nuage**

Il existe deux types d'utilisateur du nuage, administrateur ou membre d'un projet.

Selon le type d'utilisateur des interfaces ou d'autres s'affichent après l'authentification.

### **4.2.2 Authentification**

La première étape qui devrait être effectué par l'administrateur pour qu'il puisse se connecter à l'horizon est l'authentification comme la montre la figure ci-dessous.





FIGURE 4.2 – Dashboard d’authentification au nuage

### 4.2.3 Vue d’ensemble « OverView »

Une fois connecté, en fonction des privilèges d’accès, l’utilisateur est autorisé à accéder à des projets spécifiques. Ce qui suit est une page d’aperçu pour un projet appartenant à l’utilisateur **admin**.



FIGURE 4.3 – Vue d’ensemble de Nuage

### 4.2.4 Projets

La figure ci-dessous montre la phase de création d’un projet ainsi que La liste les projets disponibles (Tenants) qui ont été créés.

Créer un projet

Informations du projet \* Membres du projet Groupes du projet Quota \*

ID de Domaine default

Nom de Domaine Default

Nom \* ARC Corporate

Description Project Hopitaux Cameroun

Activé ☒

Annuler Créer un projet

FIGURE 4.4 – Création d'un Projet « Tenant »

## 4.2.5 Utilisateurs

On peut créer de nouveaux utilisateurs et / ou désactiver / supprimer des utilisateurs existants depuis le bouton « Editer ».

Créer un Utilisateur

ID de Domaine default

Nom de Domaine Default

Nom d'utilisateur \* KEMKA

Description Administrateur Principal du projet OpenStack

E-mail kemka@arccorp.com

Mot de passe \* \*\*\*\*\*

Confirmer le mot de passe \* \*\*\*\*\*

Projet primaire admin +

Rôle admin

☒ Activé

Annuler Créer un Utilisateur

Description :  
Créer un nouvel utilisateur et définir les propriétés liées en incluant le Projet Primaire et le Rôle.

FIGURE 4.5 – Création d'un utilisateur

## 4.2.6 Ajout des Images

Ici on présente la liste des Images actuellement disponibles qui peuvent être utilisés pour lancer une instance.

L'administrateur ou l'utilisateur Membre de Projet a les droits de créer des images personnalisées sur cette page.

FIGURE 4.6 – Création d'un Image

## 4.2.7 Accès et Sécurité

Les groupes de sécurité sont des ensembles de règles de filtrage IP qui sont appliqués à la configuration réseau d'une VM. Après sa création, on peut ajouter des règles à un groupe de sécurité.

Les paires de clés sont des identifiants SSH injectés dans les images lors de leur lancement. L'action de créer une nouvelle paire de clés enregistre la clé publique et télécharge la clé privée (fichier .pem)

La figure ci-dessous montre la création d'un groupe de sécurité, la mise à jour d'un groupe existant ainsi que la génération des clés privées.

FIGURE 4.7 – Création d'un groupe de sécurité

## 4.2.8 Création d'une Instance « MV »

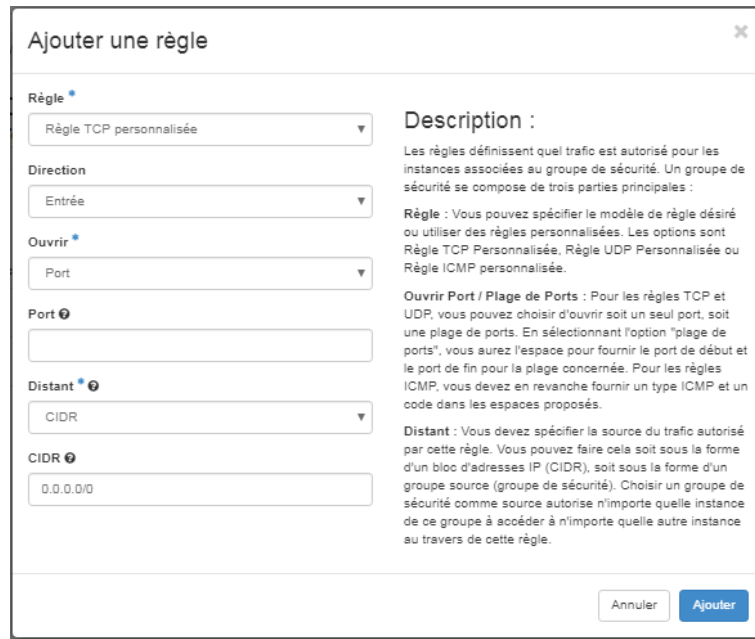
Après avoir déterminé tous les prérequis nécessaire dans notre Nuage Openstack, nous pouvons maintenant lancer une instance « machine virtuelle » depuis une image ou depuis l'onglet « instance, créer une instance »

La figure ci-dessous montre les différentes étapes à suivre pour terminer l'instanciation d'une machine virtuelle depuis l'onglet instance se basant sur une image ISO

FIGURE 4.8 – Création d'une instance « machine virtuelle »

## 4.2.9 La rédaction des règles d'un groupe

Appuyez sur le bouton Modifier les règles à côté du groupe de sécurité que vous souhaitez ajouter des règles ou modifier. Nous allons utiliser le TicCloudsec Groupe de sécurité, d'abord ajouter une règle pour permettre les connexions SSH entrantes sur le port 22.



**Ajouter une règle**

**Règle**

Règle TCP personnalisée

**Direction**

Entrée

**Ouvrir**

Port

**Port**

**Distant**

CIDR

**CIDR**

0.0.0.0/0

**Description :**

Les règles définissent quel trafic est autorisé pour les instances associées au groupe de sécurité. Un groupe de sécurité se compose de trois parties principales :

**Règle :** Vous pouvez spécifier le modèle de règle désiré ou utiliser des règles personnalisées. Les options sont Règle TCP Personnalisée, Règle UDP Personnalisée ou Règle ICMP personnalisée.

**Ouvrir Port / Plage de Ports :** Pour les règles TCP et UDP, vous pouvez choisir d'ouvrir soit un seul port, soit une plage de ports. En sélectionnant l'option "plage de ports", vous aurez l'espace pour fournir le port de début et le port de fin pour la plage concernée. Pour les règles ICMP, vous devez en revanche fournir un type ICMP et un code dans les espaces proposés.

**Distant :** Vous devez spécifier la source du trafic autorisé par cette règle. Vous pouvez faire cela soit sous la forme d'un bloc d'adresses IP (CIDR), soit sous la forme d'un groupe source (groupe de sécurité). Choisir un groupe de sécurité comme source autorise n'importe quelle instance de ce groupe à accéder à n'importe quelle autre instance au travers de cette règle.

Annuler Ajouter

FIGURE 4.9 – La rédaction des règles d'un groupe

Maintenant, l'accès ssh est activé provenant de l'adresse IP spécifiée à toutes les machines virtuelles qui ont ce groupe de sécurité associé avec elles.



**Créer une paire de clés**

**Nom de la paire de clés**

arc-key

Les paires de clés sont des informations d'authentification SSH qui sont injectées dans les images lorsqu'elles sont lancées. Créer une nouvelle paire de clés enregistre la clé publique et télécharge la clé privée (fichier .pem).

Protégez et utilisez cette clé comme vous le feriez avec toute clé SSH privée.

Annuler Créer une paire de clés

FIGURE 4.10 – L'interface de création le paire de clés.

Téléchargez et enregistrez le fichier de clé. Il sera utilisé pour se connecter à des machines virtuelles à partir de l'extérieur.

## 4.3 Bilan

Dans cette section, nous avons présenté le déploiement des composants d'identité, d'images et virtualisation sur une seule machine. Il s'agit plutôt d'une configuration de développement mais néanmoins fonctionnelle. Nous retenons que OpenStack est un logiciel libre qui permet la construction de cloud privé et public. ce dernier 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.

## CONCLUSION GÉNÉRALE

Au cours de notre travail, nous avons fait une étude sur la disponibilité et la sécurisation des données dans un système Cloud Computing. On a commencé par donner les définitions de base nécessaires à la compréhension du Cloud, son architecture et ses différents types (privée, public, hybride) et services (IaaS, PaaS, SaaS), ensuite on a présenté et détaillé les différentes solutions libres permettant de mettre en place un Cloud en faisant une étude comparative entre elles. Ceci nous a permis d'avoir une idée précise et complète sur les solutions disponibles du Cloud et surtout de choisir celle qui nous convient le mieux. Afin de mettre en place notre Cloud sous OpenStack puis sécuriser les données et les rendre disponibles, on a débuté par utiliser le formalisme UML en traçant les diagrammes de cas d'utilisation et de séquences, ceci nous a aidés à définir les besoins des utilisateurs. Nous avons fait par la suite implémenter OpenStack qui a nécessité des Prérequis matériels et logiciels.

L'implémentation de notre solution a été faite sous le système d'exploitation CentOS 7 qui a été installé sur une machine virtuelle. Ce projet étant très ambitieux, nous nous sommes vite heurtés à de nombreux problèmes, que ce soit dû aux solutions de Cloud ou à leur implémentation, notamment en ce qui concerne la sécurité.

## BIBLIOGRAPHIE

- [1] MONGUILLON Germain. *SUPINFO International University*.  
<https://www.supinfo.com/articles/single/6765-methodologies-developpement-logiciel>, Consulté le 03/04/2019, 12h21.
- [2] Atul Jha Johnson D Kiran Murari Murthy Raju Vivek Cherian Yogesh Girikumar, OpenStack Beginner's Guide (for Ubuntu - Precise), v3.0, 7, May 2012, 83 pages.
- [3] Guillaume Harry . *Capella University*. <https://www.researchgate.net/figure/Cycle-de-vie-de-projet-en-cascade-Ce-type-de-cycle-de-vie-simple-a-comprendre-et-a-fig9-256846411>, Consulté le 08/05/2019, 04h40.
- [4] CISCO, Les bases du Cloud Computing : revaloriser les technologies de l'information, Mai 2011, 7 pages.
- [5] Wygwam, Le Cloud Computing : Réelle révolution ou simple évolution ? ,83 pages.
- [6] Maxime Besson, Virtualisation et cloud open source, décembre 2012, Smile, 50 pages.
- [7] Guillaume Harry . *Capella University*. [www.researchgate.net/figure/Cycle-de-vie-de-projet-en-V-A-linstar-du-modele-en-cascade-le-modele-en-V-prend-fig10-256846411](http://www.researchgate.net/figure/Cycle-de-vie-de-projet-en-V-A-linstar-du-modele-en-cascade-le-modele-en-V-prend-fig10-256846411), Consulté le 08/05/2019, 04h41.
- [8] A.-M. Hugues, D. Wells *Génie logiciel – Cycle de vie* , Renaud Marlet, 2005-2007.
- [9] Sinouhe DARTIGALONGUE. *SUPINFO International University*.  
[www.supinfo.com/articles/single/6890-presentation-methodes-agile-methode-scrum](http://www.supinfo.com/articles/single/6890-presentation-methodes-agile-methode-scrum), Consulté le 08/05/2019, 06h49.

- [10] Thibaud Chardonnens, Les enjeux du Cloud Computing en entreprise, Université de Fribourg, Suisse, 91pages.
- [11] Abdalah SLAMA. *SUPINFO International University*. [www.supinfo.com/articles/single/3093-comparatif-methodes-agiles](http://www.supinfo.com/articles/single/3093-comparatif-methodes-agiles), Consulté le 09/05/2019, 05h20.
- [12] Nicolas GREVET, Le Cloud Computing : Evolution ou Révolution ?, M2IRT 2009, 128 pages.
- [13] [http ://www.openstack.org/software/](http://www.openstack.org/software/), consulté le 19 Avril 2019 à 01h20.
- [14] [http ://openstack-folsom-install-guide.readthedocs.org/en/latest/](http://openstack-folsom-install-guide.readthedocs.org/en/latest/), consulté le 01 Avril 2019 à 14h13.
- [15] Ken Pepple, Depolying Openstack, O'Reilly, July 2011 : First Edition, 86 pages.
- [16] [http ://docs.openstack.org/folsom/openstack-compute/admin/content/about\\_thedashboard.html](http://docs.openstack.org/folsom/openstack-compute/admin/content/about_thedashboard.html), consulté le 05 Avril 2019 à 18h02.
- [17] [http ://docs.openstack.org/developer/swift/overview-large-objects.html](http://docs.openstack.org/developer/swift/overview-large-objects.html), consulté le 10 Mai 2019.
- [18] [http ://docs.openstack.org/folsom/openstackcompute/admin/content/conceptualarchitecture.html](http://docs.openstack.org/folsom/openstackcompute/admin/content/conceptualarchitecture.html), consulté le 18 Mai 2019 à 12h19.
- [19] [https ://www.redsen-consulting.com/fr/inspired/intelligence-collective/gestion-connaissances](https://www.redsen-consulting.com/fr/inspired/intelligence-collective/gestion-connaissances), consulté le 12 Février 2019 à 23h40.
- [20] Pierre-Alain FOUQUE, Applications de la cryptographie : protocoles de communication, Cryptographie appliquée, 10 nov. 2003.
- [21] [http ://informatiqueservicesplus.ca/quels-sont-les-caracteristiques-du-cloud-computing/](http://informatiqueservicesplus.ca/quels-sont-les-caracteristiques-du-cloud-computing/), consulté le 15 Avril 2019 à 00h19.
- [22] [https ://www.developpez.net/forums/d544945/general-developpement/alm/methodes/xup/planned-utilisation-2tup/](https://www.developpez.net/forums/d544945/general-developpement/alm/methodes/xup/planned-utilisation-2tup/), consulté le 10 Mai 2019 à 19h27.
- [23] Jean-François Pillou, [https ://www.commentcamarche.net/contents/48-les-scanners-de-vulnerabilites-balayage-de-ports](https://www.commentcamarche.net/contents/48-les-scanners-de-vulnerabilites-balayage-de-ports), Consulté le 12 Mai à 14h50



## ANNEXE A

### A.1. Installation

Installez les paquets keystone,python-keystone,python-keystoneclient,python- mysqldb

Puis supprimer la base de données SQLite :

```
rm /var/lib/keystone/keystone.db
```

#### A.1.1. Configuration

On doit ouvrir avec les droits d'administrateur le fichier /etc/keystone/keystone.conf pour modifier les sections suivantes : en remplaçant ADMPASSWD par le mot de passe root et et SQLPASSWD par le mot de passe root.

Redémarrez keystone :

```
sudo service keystone restart
```

- synchronisez la base de données : `sudo keystone-manage db-sync`

#### A.1.2. Création des utilisateurs

Chaque commande ci-dessous contient l'authentification définie dans le fichier keystone.conf et utilisée par le client python sous la forme `–token admin-token root –endpoint url-du-service-keystone 172.20.203.220 :35357/v02/`.

##### A.1.2.1. Création du compte administrateur

```
keystone –token ADMPASSWD –endpoint http ://172.20.203.220 :35357/v2.0/ user-create –name=admin  
–pass=root –email=admin@example.com
```

##### A.1.2.2. Création du compte interne du service Glance

```
keystone –token ADMPASSWD –endpoint http ://172.20.203.220 :35357/v2.0/ user-create –name=glance  
–pass=root –email=glance@example.com
```

### **A.1.2.3. Création du compte interne du service Nova**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ user-create --name=nova  
--pass=root --email=nova@example.com
```

### **A.2. Création des rôles**

Pour les rôles utilisateurs vous avez le choix entre :

Admin : donne le droit de modifier la configuration des services (ex :allouer une plage d'adresse IP, un quota d'espace disque pour un projet etc...)

Member : permet de gérer le contenu du projet (création d'instances de machines, ajout d'un disque virtuel a l'une d'elles etc...)

Les rôles KeystoneAdmin et KeystoneServiceAdmin sont des rôles internes nécessaires.

#### **A.2.1. Rôle admin**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ role-create --name=admin
```

#### **A.2.1.1. Rôle Membre**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ role-create --name=Member
```

#### **A.1.2.2. Rôle KeystoneAdmin**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ role-create --name=KeystoneAdmin
```

#### **A.1.2.3. Rôle KeystoneServiceAdmin**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ role-create --name=KeystoneServiceAdmin
```

### **A.3. Création des Tenants**

#### **A.3.1. Tenant admin**

Le Tenant admin permet à ses membres d'administrer les services.

#### **A.3.2. Tenant service**

Le Tenant interne des services.

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ tenant-create --  
name=service
```

#### **A.3.3. Définition des rôles**

Il faut pour cela utiliser les ID affichés lors de la création des Users, Roles et Tenants.

L'User "admin" a un Role admin sur le Tenant "admin"

```
keystone --token root --endpoint http://192.168.1.250:35357/v2.0/  
user-role-add --user-id c97c87b3ed894401975dd6d757b40330  
--role-id 3d945f41e08e4e2db1584fdb8f05d333  
--tenant-id 0f71e86d30e247d3b1216fe5f2f3aa50
```

Comme ce n'est pas pratique de recopier les IDs, les erreurs de frappe seront évitées grâce à l'outil `awk`. Il s'agira de définir les rôles ainsi :

- L'User "admin" a un Role "KeystoneAdmin" sur le Tenant "admin".
- L'User "admin" a un Role "KeystoneServiceAdmin" sur le Tenant "admin".
- L'User "glance" a un Role "admin" sur le Tenant "service".
- L'User "nova" a un Role "admin" sur le Tenant "service".

Voici les commandes correspondantes :

- `keystone user-role-add --user-id 'keystone user-list | awk '/ admin / print 2 '' --role-id 'keystone role-list | awk '/ KeystoneAdmin / print 2 '' --tenant-id 'keystone tenant-list | awk '/ admin / print 2 ''`
- `keystone user-role-add --user-id 'keystone user-list | awk '/ admin / print 2 '' --role-id 'keystone role-list | awk '/ KeystoneServiceAdmin / print 2 '' --tenant-id 'keystone tenant-list | awk '/ admin / print 2 ''`
- `keystone user-role-add --user-id 'keystone user-list | awk '/ glance / print 2 '' --role-id 'keystone role-list | awk '/ admin / print 2 '' --tenant-id 'keystone tenant-list | awk '/ service / print 2 ''`
- `keystone user-role-add --user-id 'keystone user-list | awk '/ nova / print 2 '' --role-id 'keystone role-list | awk '/ admin / print 2 '' --tenant-id 'keystone tenant-list | awk '/ service / print 2 ''`

#### **A.4. Création d'un utilisateur supplémentaire**

Il s'agira dans l'exemple qui suit de la création d'un compte utilisateur, d'un projet supplémentaire et définition du rôle avec la variable d'environnement `USER` (remplacer par ce que vous voulez, c'est juste un exemple)

Le rôle "Member" est suffisant. Remplacez `USRPASSWD` par un mot de passe de votre choix. L'User `USER` (demo) a un Role "Member" sur le Tenant `USER` (demo ici).

##### **A.4.1. User**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ user-create --name=demo --pass=root --email=demo@example.com
```

##### **A.4.1. Tenant**

```
keystone --token ADMPASSWD --endpoint http://172.20.203.220:35357/v2.0/ tenant-create --name=demo
```

##### **A.4.1. Rôle**

```
keystone --token ADMPASSWD --endpoint http://192.168.1.250:35357/v2.0/ user-role-add --user-id 13247a59ad844458ad36c0bd06451376 --role-id 84697b61736c439288900904bdf4a48d
```

–tenant-id c6f05a03b4aa482c91b61a2230356618

## **A.5. Création des services et leurs points d'accès**

### **A.5.1. Le service Keystone**

```
keystone –token ADMPASSWD –endpoint http ://172.20.203.220 :35357/v2.0/ service-create –  
name=keystone –type=identity –description='Keystone Identity Service'
```

### **A.5.2. Le point d'accès Keystone**

```
keystone –token ADMPASSWD –endpoint http ://172.20.203.220 :35357/v2.0/  
endpoint-create –region RegionOne – service-id=41905e02540d48228166c6d06ddcd9f0 – pu-  
blicurl=http ://172.20.203.220 :5000/v2.0 – internalurl=http ://172.20.203.220 :5000/v2.0 – admi-  
nurl=http ://172.20.203.220 :35357/v2.0
```

Les services et points d'accès des autres services seront ajoutés après l'installation du composant bien qu'il soit possible de les définir dès maintenant.

## **A.6. Utilisation**

Il y a plusieurs façons possibles de s'identifier en lançant une commande keystone.

La méthode d'identification utilisée précédemment avec le mot de passe d'administration (variable admin-token définie dans le fichier keystone.conf) avec les arguments –endpoint et –token

Pour éviter de refaire un export des variables à chaque ouverture de terminal, on peut les exporter automatiquement.

Il suffit de créer un fichier .novarc dans un dossier contenant les lignes suivantes :

```
export OS-TENANT-NAME=admin export OS-USERNAME=admin export OS-PASSWORD=root  
export OS-AUTH-URL=http ://172.20.203.220 :5000/v2.0/
```

Ensuite on ajoute la ligne suivante à la fin de fichier .bashrc

Enfin en execute le fichier .novarc

Les variables seront exportées comme variables d'environnement et on peut utiliser toutes les commandes sous la forme simple sans ressaisir les informations d'authentification.

```
keystone user-list
```

### **A.6.1. Glance**

La prochaine étape est l'installation du service d'images Glance.

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

### **A.6.2. Préparation de la base de données Mysql**

La commande suivante crée un utilisateur et sa base de données nommés « glance » avec le mot de passe « root ».

```
mysql -u root -p «EOF CREATE DATABASE glance;
```

```
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'root';
```

```
FLUSH PRIVILEGES; EOF
```

#### **A.6.1. Installation**

Installez les paquets glance, glance-api, glance-client, glance-common, glance-registry, python-glance

#### **A 6.2. Configuration**

Il faut aussi créer les services et points d'accès correspondants pour Keystone

```
keystone service-create --name=glance --type=image --description='Glance Image Service'
```

```
keystone endpoint-create --region RegionOne --service-id=39bbd3107c4c4153a408a3b6a34ef931  
--publicurl=http://172.20.203.220:9292/v1 --internalurl=http://172.20.203.220:9292/v1 --adminurl=http://172.20.203.220:9292/v1
```

#### **A.6.3. Nova**

Passez maintenant à l'installation de Nova, la gestion des instances des machines virtuelles, de notre espace disque et du réseau.

#### **A.6.4. Préparation de la base de données Mysql**

La commande suivante crée un utilisateur et sa base de données nommés "nova". Avec un mot de passe « root ».

```
mysql -u root -p «EOF CREATE DATABASE nova; GRANT ALL PRIVILEGES ON nova.* TO  
'nova'@'root' IDENTIFIED BY 'SQLPASSWD'; EOF
```

##### **A.6.4.1. Installation**

Nous allons installer les paquets nova-api nova-cert nova-common nova-compute apt-get install nova-compute-kvm nova-doc nova-network nova-objectstore nova-scheduler novnc nova-consoleauth nova-volume python-nova python-novaclient.

##### **A 6.4.2. Configuration**

Création des services et points d'accès pour Keystone, au nombre de 2 : les services de type compute (auquel on donne le nom de "nova") et de type volume (auquel on donne le nom de "volume").

## ANNEXE B

### B.1 Les scanners des vulnérabilités

#### B.1.1 Nessus

Pour utiliser Nessus suivre les étapes suivantes :

Une fois Nessus installé et tous les plugins installés, lancez cette commande via le terminal :

```
/etc/init.d/nessusdstart
```

Nessus est maintenant lancé. Rendez vous à l'adresse *http ://127.0.0.1 :8834* ou *http ://votrema-  
chine :8834* pour vous connecter à Nessus. Laissez-le s'initialiser, puis vous serez redirigés vers la page de login où vous devrez vous identifier.

Pour effectuer un scan, cliquez sur New Scan et indiquez le nom du scan, la description, la police et la liste des hôtes à scanner.

Ici, mettez l'adresse IPv4 de la Target, pour nous ce sera 192.168.1.1 Le scan devrait se lancer. Attendez un moment, le temps que Nessus scan la machine(Cloud), puis une fois que Nessus vous indiquera que le scan est terminé, cliquez sur le scan pour afficher le résultat du scan.

#### B.1.2 Nmap

On utilise le Nmap parce que le OpenStack est installé sur une machine Virtuel.

# Table des matières

<b>Dédicace</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Sommaire</b>	<b>v</b>
<b>Glossaire</b>	<b>vi</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Résumé</b>	<b>xii</b>
<b>Abstract</b>	<b>xiii</b>
<b>Introduction Générale</b>	<b>1</b>
<b>Partie I ETAT DE L'ART</b>	<b>3</b>
<b>Chapitre 1 GESTION DES DONNÉES DANS UNE ENTREPRISE</b>	<b>4</b>
1.1 La donnée dans un monde numérique. [19] . . . . .	4
1.1.1 Le stockage des données numériques. . . . .	6
1.1.1.1 Les technologies. . . . .	6
1.1.1.2 Les principaux modes de stockage. . . . .	7
1.1.2 Les principaux défis pour la gestion des données numériques. . . . .	7
1.2 Étude de l'existant . . . . .	8
1.3 Critique de l'existant / Problématique . . . . .	9

<b>Chapitre 2</b>	<b>LE CLOUD COMPUTING - GESTION DE PROJET</b>	<b>11</b>
2.1	Généralités sur le cloud computing. . . . .	11
2.1.1	Les origines.[5] . . . . .	11
2.1.2	Bénéfices du cloud Computing.[4] . . . . .	12
2.1.2.1	Pour le fournisseur . . . . .	12
2.1.2.2	Pour l'entreprise . . . . .	13
2.1.3	Les principales caractéristiques.[4] . . . . .	14
2.1.4	Éléments constitutifs du Cloud Computing.[6] . . . . .	14
2.1.4.1	La virtualisation . . . . .	14
2.1.4.2	Le Datacenter . . . . .	15
2.1.4.3	La Plateforme collaborative . . . . .	15
2.1.5	Les modèles de service.[5] . . . . .	16
2.1.5.1	IaaS : Infrastructure as a Service . . . . .	16
2.1.5.2	PaaS : Plateform as a Service. . . . .	16
2.1.5.3	SaaS : Software as a Service. . . . .	16
2.1.5.4	Avantages et inconvénients des services. [10] . . . . .	19
2.1.6	Les modèles de déploiement.[4] . . . . .	20
2.1.7	Les principales applications. . . . .	21
2.1.8	Les principaux acteurs. . . . .	22
2.1.9	Les principaux avantages, limites et contraintes. . . . .	24
2.2	Les aspects de sécurité du cloud. . . . .	25
2.2.1	Les rappels sur la cryptographie.[20] . . . . .	26
2.2.2	Les objectifs de sécurité. . . . .	28
2.2.3	Les attaques potentielles. . . . .	29
2.2.4	Le cycle de vie de la sécurité des données dans le cloud. . . . .	30
2.3	Gestion d'un projet de cloud computing. . . . .	31
2.3.1	Caractéristiques de la solution voulue.[21] . . . . .	31
2.3.2	Objectif du Projet . . . . .	32
2.3.3	Méthodes d'analyses . . . . .	32
2.3.3.1	Modèle de cycle de vie en cascade.[9] . . . . .	32
2.3.3.2	Modèle de cycle de vie en V.[11] . . . . .	33
2.3.3.3	Modèle de cycle de vie en spirale.[7] . . . . .	34



2.3.3.4	La méthode agile.[9]	35
2.3.3.5	La méthodologie 2TUP ou cycle en Y	36
2.4	Solutions du Cloud existante.	37
2.4.1	Solutions propriétaires.[6]	37
2.4.1.1	VMwareCloud.	37
2.4.1.2	Office 365.	38
2.4.2	Solutions libres.[6]	39
2.4.3	Eucalyptus	39
2.4.4	OpenNubela	39
2.4.5	OpenStack	39
2.5	Critique générale.	40
2.5.1	Comparaison entre les solutions du cloud computing.	40
2.5.2	Comparaison des méthodes d'analyse.	42
2.6	Les scanners des vulnérabilités.[23]	42
2.6.1	Nessus	43
2.6.2	Nmap	43
2.7	Choix des outils.	44
2.7.1	Choix de la méthode d'analyse.	44
2.7.2	Choix de la solution à déployer.[19]	44
2.7.3	Présentation	45
2.7.4	Architecture.[15]	46
2.7.4.1	OpenStackCompute (projet Nova).	46
2.7.4.2	OpenStack Object Storage (projet Swift).	49
2.7.4.3	OpenStack Imaging Service (projet Glance).	50

## **Partie II ANALYSE MODELISATION ET MISE EN PLACE DU PRO- JET** **52**

### **Chapitre 3 PRÉSENTATION DE LA SOLUTION ET ANALYSE** **53**

3.1	Cahier de charges	53
3.1.1	Nom du projet.	53
3.1.2	Origine et genèse du projet.	53
3.1.3	Précision, objectifs et résultat.	54

3.1.4	Planification. . . . .	54
3.1.5	Estimation des ressources matérielles et financières. . . . .	55
3.2	Branche Fonctionnelle : Analyse et Spécification des Besoins . . . . .	55
3.2.1	Besoins fonctionnels . . . . .	56
3.2.1.1	Gestion d'images . . . . .	56
3.2.1.2	Gestion d'instances . . . . .	56
3.2.1.3	Gestion des volumes . . . . .	56
3.2.1.4	Gestion des flavors . . . . .	56
3.2.1.5	Gestion des projets . . . . .	57
3.2.1.6	Gestion des utilisateurs . . . . .	57
3.2.1.7	Gestion de la sécurité et de l'accès . . . . .	57
3.2.2	Besoins non fonctionnels . . . . .	57
3.2.2.1	Simplicité d'un service à la demande . . . . .	57
3.2.2.2	Extrême flexibilité . . . . .	57
3.2.2.3	Accès léger . . . . .	57
3.2.2.4	Sûreté . . . . .	58
3.2.2.5	Vivacité . . . . .	58
3.2.3	Identification des acteurs. [17] . . . . .	58
3.2.3.1	Les acteurs du système . . . . .	58
3.2.3.2	Diagramme de cas d'utilisation Générale . . . . .	58
3.2.4	Raffinements des Cas d'Utilisations . . . . .	59
3.2.4.1	Cas d'utilisation « Consulter l'état du nuage » . . . . .	59
3.2.4.2	Cas d'utilisation « Gérer les instances » . . . . .	60
3.2.4.3	Cas d'utilisation « Gérer les services » . . . . .	61
3.2.4.4	Cas d'utilisation « Gérer les Flavors » . . . . .	62
3.2.4.5	Cas d'utilisation « Gérer les Images » . . . . .	63
3.2.4.6	Cas d'utilisation « Gérer les Projets » . . . . .	64
3.2.4.7	Cas d'utilisation « Gérer les Utilisateurs » . . . . .	65
3.2.4.8	Diagramme de cas d'utilisation « Membre d'un projet » . . . . .	66
3.2.5	Diagrammes d'activité . . . . .	67
3.2.5.1	Diagrammes d'activité globale . . . . .	67
3.2.5.2	Diagrammes d'activité « Créer une instance » . . . . .	68

3.3	Branche Techniques : Environnement Matériel et Logiciel . . . . .	69
3.3.1	Architecture Physique . . . . .	69
3.3.2	Partie de Stockage . . . . .	70
3.3.3	Baie de stockage . . . . .	70
3.3.4	Serveur . . . . .	71
3.3.5	Serveurs blades . . . . .	71
3.4	Architecture Réseau.[18] . . . . .	72
3.5	Diagramme de déploiement du système. . . . .	73
3.6	Environnement Logiciel. [15] . . . . .	74
3.6.1	Planification du déploiement d'OpenStack . . . . .	75
3.7	Architecture de Solution Openstack.[13] . . . . .	76
3.8	Diagramme de Déploiement . . . . .	78
3.8.1	Diagrammes de séquences . . . . .	78
3.8.2	Diagrammes de séquences globales . . . . .	79
3.8.3	Diagramme de séquences « scénario d'authentification » . . . . .	79
3.8.4	Diagramme de séquences « Créer un projet » . . . . .	80
3.8.5	Diagramme de séquences « scénario création d'une instance » . . . . .	81
3.8.6	Diagramme de séquences d'entités globales . . . . .	83
3.8.7	Diagramme de séquences entité « authentification » . . . . .	84
3.8.8	Diagramme de séquences entité « Créer un projet » . . . . .	85
3.8.9	Diagramme de séquences entité « créer une Instance » . . . . .	85

## **Chapitre 4 IMPLEMENTATION – TEST - BILAN 87**

4.1	Implémentation. . . . .	87
4.1.1	Les Composants.[16] . . . . .	87
4.1.2	Prérequis . . . . .	89
4.1.3	Les paquets à installer . . . . .	89
4.1.4	Préparation du système . . . . .	89
4.1.4.1	Réseau . . . . .	89
4.1.4.2	Serveur NTP . . . . .	89
4.1.4.3	RabbitMQ . . . . .	90
4.1.4.4	Mysql . . . . .	90
4.1.4.5	Keystone . . . . .	90

4.1.4.6	Préparation de la base de données Mysql . . . . .	90
4.1.4.7	Installation . . . . .	91
4.2	Tests . . . . .	91
4.2.1	Interface Authentification au nuage . . . . .	91
4.2.2	Authentification . . . . .	91
4.2.3	Vue d'ensemble « OverView » . . . . .	92
4.2.4	Projets . . . . .	92
4.2.5	Utilisateurs . . . . .	93
4.2.6	Ajout des Images . . . . .	94
4.2.7	Accès et Sécurité . . . . .	94
4.2.8	Création d'une Instance « MV » . . . . .	95
4.2.9	La rédaction des règles d'un groupe . . . . .	95
4.3	Bilan . . . . .	96
<b>Conclusion Générale</b>		<b>97</b>
<b>Bibliographie</b>		<b>A</b>
<b>Chapitre A</b>		<b>AA</b>
<b>Chapitre B</b>		<b>BA</b>
B.1	Les scanners des vulnérabilités . . . . .	BA
B.1.1	Nessus . . . . .	BA
B.1.2	Nmap . . . . .	BA
<b>Table des matières</b>		<b>BB</b>