



INFORMATIQUE ET TELECOMMUNICATIONS

ETUDE ET MISE EN PLACE D'UNE SOLUTION DE CLOUD COMPUTING PRIVÉ BASÉE SUR UN ALGORITHME DE SUPERVISION DISTRIBUÉ : APPLICATION AUX SERVICES IAAS : CAS DE MATRIX TÉLÉCOMS

Mémoire présenté et soutenu en vue de l'obtention du Diplôme d'INGENIEUR DE
CONCEPTION EN RESEAUX

Par

NZOGANG SOH Etienne Borel

Licence en Ingénierie des Réseaux et Télécommunications

Matricule : 14A734S

Sous la Direction de

Dr NLONG II Jean Michel

Chargé de Cours

Devant le jury composé de :

Président : Pr DANWE Raïdandi

Rapporteur : Dr NLONG II Jean Michel

Examineur : Prof. Dr.-Ing. habil. KOLYANG

Invité : Ing NDA Edwige LONGMENE

Année Académique 2015 / 2016

DEDICACE

A
mes parents.

REMERCIEMENTS

Je ne saurais entamer la rédaction de ce mémoire sans dire un grand merci à tous ceux qui de près ou de loin ont contribué au déroulement de ma formation, je pense particulièrement :

- ❖ A monsieur et madame **NZOGANG** qui n'ont cessés de me prodiguer de bons conseils, je suis reconnaissant du soutien moral, financier et matériel.
- ❖ Au président du jury **Pr DANWE Raïdandi**, directeur de l'ISS
- ❖ Au rapporteur **Dr NLONG II Jean Michel** mon encadreur pour sa disponibilité, son suivi, ses encouragements, ses conseils.
- ❖ A l'examineur **Prof. Dr.-Ing. habil. KOLYANG**
- ❖ A tout le personnel de MATRIX TELECOMS en particulier l'**Ing. Edwige LONGMENE** mon encadreur professionnel pour sa disponibilité, son suivi, ses encouragements, ses conseils.
- ❖ A toute l'équipe pédagogique de l'ISS de Maroua pour l'encadrement et les connaissances reçues.
- ❖ A toutes ma famille principalement mes frères et sœurs pour leur soutien moral, financier et matériel.
- ❖ A tous mes **camarades** de l'ISS de Maroua, ainsi qu'à mes **amis** pour leur solidarité.
- ❖ A toutes mes connaissances bien aimées dont les noms ne figurent pas ici, je vous dis merci.

Merci pour tous ces efforts déployés afin que je devienne un homme responsable et dynamique.

TABLE DES MATIERES

DEDICACE	<i>i</i>
REMERCIEMENTS	<i>ii</i>
LISTE DES SIGLES ET DES ABREVIATIONS	<i>v</i>
RESUME	<i>viii</i>
ABSTRACT	<i>ix</i>
LISTE DES TABLEAUX	<i>x</i>
LISTE DES FIGURES ET ILLUSTRATIONS	<i>xi</i>
INTRODUCTION GENERALE	<i>1</i>
Chapitre I : CONTEXTE ET PROBLEMATIQUE	<i>3</i>
I.1 Présentation de l'entreprise	<i>3</i>
I.1.1 Historique de Matrix Télécoms	<i>3</i>
I.1.2 Fiche signalétique, organigramme et situation géographique de matrix	<i>4</i>
I.1.3 Services offerts, architecture et missions de MATRIX TELECOMS	<i>6</i>
I.2 Contexte et problématique	<i>8</i>
I.3 Méthodologie	<i>9</i>
I.4 Objectifs	<i>9</i>
Chapitre II : GENERALITES SUR LE CLOUD COMPUTING	<i>11</i>
II.1 Les concepts du Cloud Computing	<i>11</i>
II.1.1 Définition	<i>11</i>
II.1.2 Les services du Cloud	<i>12</i>
II.1.3 Les modèles de déploiement	<i>16</i>
II.2 Les technologies du Cloud Computing	<i>18</i>
II.2.1 La virtualisation	<i>18</i>
II.2.2 Les services web	<i>27</i>
II.2.3 Le web 2.0	<i>31</i>
II.3 Les solutions du Cloud Computing	<i>32</i>
II.3.1 Les solutions propriétaires	<i>32</i>
II.3.2 Les solutions open source	<i>33</i>
Chapitre III : ALGORITHME DE SUPERVISION DISTRIBUE	<i>40</i>
III.1 Notion de système distribué	<i>40</i>
III.2 Algorithmes distribués	<i>41</i>
III.2.1 Mémoire partagé	<i>42</i>

III.2.2 Agents mobiles	42
III.2.3 Passage de message	43
III.3 Principe d'élection dans un système distribué	43
III.4 Caractérisation des systèmes distribués	43
III.5 La réplication des données	44
III.5.1 La réplication des données synchrone	45
III.5.2 La réplication des données asynchrone	45
III.5.3 Les différents types de réplicateurs	46
III.6 Présentation de notre solution	46
Chapitre IV : IMPLEMENTATION DU PROJET ET CHOIX TECHNIQUES	52
IV.1 Etude et planification projet	52
IV.1.1 Analyse du besoin	52
IV.1.2 Elaboration de la charte du projet	54
IV.1.3 Diagramme de cas d'utilisation	56
IV.1.4 Estimation des ressources matérielles et financière	57
IV.2 Choix technique	58
IV.2.1 La solution de virtualisation employée	58
IV.2.2 Le portail web des services IAAS	58
IV.2.3 Synthèse des outils utilisés	59
Chapitre V : RESULTATS ET COMMENTAIRES	61
V.1 Résultats	61
V.1.1 Architecture de tests	61
V.1.2 Quelques interfaces	63
V.2 Commentaires	69
CONCLUSION ET PERSPECTIVES	71
BIBLIOGRAPHIE	72
GLOSSAIRE	76
ANNEXE 1 : Explication du fonctionnement d'Eucalytus & OpenNebula	78
2.1 Fonctionnement d'Eucalyptus	78
2.2 Fonctionnement d'OpenNebula	80
ANNEXE 3 : Documentation de Xen orchestra	82
ANNEXE 4 : Mise en place de la réplication DRBD entre 2 XenServer	84

LISTE DES SIGLES ET DES ABREVIATIONS

API: Application Programming Interface

CA: Chargé d’Affaire

CC: Cluster Controller

CDMA : Code Division Multiple Access

CLC: Cloud Controller

CLI: Command Line Interpreter

CM: Crédit Management

CORBA: Common Object Request Broker Architecture

CPU: Central Processing Unit

CRM: Customer Relationship Management.

CSG & INFRA: Customer Service Group & Infrastructure

DAF : Direction des Affaires Financières

DD : Disque Dur

DG (ADG/DGA) : Direction général (Administrateur Directeur Général/Directeur Général Adjoint)

DMC : Direction Marketing et Commerciale

DNS: Domain Name Server

DRBD: Distributed Replicated Block Device

DRH : Direction des Ressources Humaines

DS/SUY : Directeur Succursale Yaoundé

DSD : Direction Succursale Douala

DSI : directeur des systèmes d’informations.

DSY : Direction Succursale Yaoundé

DT (DTA/ADT) : Direction Technique (Directeur Technique Adjoint/ Administrateur Directeur Technique)

EC2: Elastic Compute Cloud

ERP: Enterprise Resources Planning.

GNU/Linux: GNU is Not Unix/Linux

GPL: General Public License

GSM: Global System for Mobile communications

GUI: Graphical User Interface

HTTP: HyperText Transfert Protocol
HVM: Hardware Virtual Machine
IAAS: Infrastructure as a Service
ICC: International Computer Center
ICCNET: International Computer Center NETwork
IDR: Informatica Data Replication
IP: Internet Protocol
ISS: Institut Supérieur du Sahel
JSON: JavaScript Object Notation
KVM: Keyboard, Video, Mouse
LAN: Local Area Network
LVM: Logical Volume Manager
NDA: Network and Development Administrator
NIST: National Institute of Standard and Technology
NOC: Network Operating Center
NoSQL: No Structured Query Language
OS: Operating system
PAAS: Platform as a Service
PME/PMI: Petites et Moyennes Entreprises/Petites et Moyennes Industries
POP: Point Of Presence
RAM: Random Access Memory
RCY: Responsable Commercial Yaoundé
RDP: Remote Desktop Protocol
REST: Representational State Transfer
RFB: Remote Frame Buffer
RSCRY : Responsable Clientèle et Recouvrement Yaoundé
RTY : Responsable Technique Yaoundé
SAAS: Software as a Service
SIC : Système d'Information et de Communication
SMTP: Simple Mail Transfer Protocol
SOA : service oriented architecture
SOAP: Simple Object Access Protocol
SQL: Structured Query Language

SRDF: Symmetrix Remote Data Facility
SSH: Secure SHell
UDDI: Universal Description, Discovery, and Integration.
UIT: Union Internationale des Télécommunications
URL: Uniform Resource Locator
VCL: Virtual Computing Laboratory
VDS: Virtual Delicated Server
VM: Virtual Machin
VMM: Virtual Machine Monitor
VNC: Virtual Network Computing
VPN: Virtual Private Network
VPS: Virtual Private Server
WBS: Work Breakdown Structure
WSDL: Web Services Description Language
XCP: Xen Cloud Platform
XML: eXtensible Markup Language
XML-RPC: eXtensible Markup Language- Remote Procedure Call

RESUME

Le présent document présente le processus de conception et de réalisation d'une solution de Cloud Computing privé afin d'offrir des services IAAS dans une architecture distribuée et supervisée. Cette solution a été conçue pour l'entreprise MATRIX TELECOMS afin d'améliorer le rendement et les performances de ses employés. Car en effet l'achat d'ordinateurs puissant et le temps pour l'installation de l'environnement de travail coûtent énormément d'argent à l'entreprise. Cette plateforme nous permettra donc de servir des machines virtuelles personnalisées et puissantes aux employés et est constituée d'un ensemble d'outils allant du Serveur de machines virtuelles jusqu'au serveur de bureau distant en passant par un centre de supervision.

Dans ce projet nous avons mis en place un pool de serveurs de machines virtuelles (maître/esclaves) qui tournent sous l'OS XenServer, ensuite nous avons mis sur pied XenCenter qui est notre centre de supervision, puis pour l'accès à distance aux machines virtuelles par les clients nous avons deux serveurs tournant sous l'OS Windows Server 2012 R2 ; l'un contenant le contrôleur de domaine Active Directory, le DNS et notre autorité de certification pour signer nos certificats en local. Et l'autre contenant Xen Desktop qui utilise le protocole VNC et permet de fournir les bureaux distants aux clients receiver (machines clientes). Enfin nous avons configuré DRBD afin de faire de la distribution des tâches et surtout répliquer les machines virtuelles du maître sur les esclaves afin de rendre notre architecture tolérante aux pannes, surtout celles du serveur maître.

Pendant la réalisation de ce projet, nous avons rencontré plusieurs difficultés notamment au niveau de l'obtention des ressources matériels et de la compréhension de ce concept nouveau qu'est le Cloud Computing.

Mots clés : **Cloud Computing, supervision, distribué, répliquer, virtuelle.**

ABSTRACT

This paper presents the process of design and implementation of a private cloud computing solution to provide IAAS services in a distributed and supervised architecture. This solution was designed for the company MATRIX TELECOMS to improve the efficiency and performance of its employees. For indeed, the purchase of powerful computers and the time required for the installation of the working environment cost a lot of money to the company. This platform will allow us to serve personalized and powerful virtual machines to employees and will consist of a set of tools moving from the virtual machine server to the remote desktop server through a monitoring center.

In this project, we have set up a pool of virtual machine servers (master / slaves) which run on the XenServer OS, we then established XenCenter which is our supervision center and for the remote access to virtual machines by customers, we have two servers running Windows Server 2012 R2 OS; one containing the Active Directory domain controller, the DNS, and our certification authority to locally sign our certificates. And the other container Xen Desktop that uses the VNC protocol and allows to provide remote offices to customer receiver (client machines). Finally we configured DRBD to make the distribution of tasks and especially replicate the master's virtual machines on the slaves to make our architecture to be fault tolerant, especially those of the master server.

During the realization of this project, we encountered several difficulties, especially at the level of obtaining material resources and the understanding of this new concept that is Cloud Computing.

Keywords: Cloud computing, supervision, distributed, replicate, virtual.

LISTE DES TABLEAUX

<i>Tableau 1.1 : fiche signalétique de MATRIX Télécoms</i>	4
<i>Tableau 2.1 : Synthèse des techniques de virtualisation (adaptée de Mahjoub)</i>	23
<i>Tableau 2.2 : Comparaison des hyperviseurs</i>	25
<i>Tableau 4.1 : Valeur des critères à atteindre</i>	53
<i>Tableau 4.2 : estimation financière des ressources matériels</i>	57
<i>Tableau 4.3 : Synthèse des outils utilisés pour l'implémentation</i>	59
<i>Tableau 5.1 : récapitulatif des équipements et de leurs caractéristiques</i>	62

LISTE DES FIGURES ET ILLUSTRATIONS

<i>Figure 1.1 : Organigramme de MATRIX Télécoms</i>	5
<i>Figure 1.2 : Plan de localisation de MATRIX Télécoms</i>	6
<i>Figure 1.3 : Architecture réseau cœur de MATRIX Télécoms</i>	7
<i>Figure 2.1: Niveaux de service Cloud</i>	12
<i>Figure 2.2 : Les éléments constitutifs des services IAAS du Cloud Computing</i>	14
<i>Figure 2.3 : Principe de la virtualisation</i>	19
<i>Figure 2.4 : principe de l'isolation</i>	20
<i>Figure 2.5 : principe de fonctionnement de la virtualisation complète</i>	21
<i>Figure 2.6 : principe de la para-virtualisation</i>	22
<i>Figure 2.7 : principe des systèmes à hyperviseur</i>	23
<i>Figure 3.1: Algorithmique classique</i>	41
<i>Figure 3.2: Algorithme à mémoire partagé</i>	42
<i>Figure 3.3 : Algorithme à agents mobiles</i>	42
<i>Figure 3.4 : algorithme distribué par passage de messages</i>	43
<i>Figure 3.5 : Schéma de l'architecture du cloud privé : Cas de panne du nœud central de supervision.</i>	47
<i>Figure 3.6 : Schéma de l'architecture du cloud privé : cas de panne d'un cluster Controller.</i>	48
<i>Figure 3.7 : schéma de la solution proposée</i>	49
<i>Figure 3.8 : Schéma d'un scénario de reprise des services du cloud Controller par un cluster Controller et réélections d'un nouveau cluster Controller de supervision.</i>	50
<i>Figure 3.9 : Schéma d'un scénario de reprise des services d'un cluster Controller par un autre cluster Controller</i>	50
<i>Figure 4.1 : Graphe normalisé de l'expression du besoin</i>	53
<i>Figure 4.2 graphe WBS du projet</i>	54
<i>Figure 4.3 : cas d'utilisation global du système</i>	56
<i>Figure 5.1 : Architecture de tests</i>	61

<i>Figure A.1 : lancement d'une machine virtuelle avec OpenNebula</i>	81
<i>Illustration 2.1 : logo de Windows Azure</i>	32
<i>Illustration 2.2 : logo Google App Engine</i>	32
<i>Illustration 2.3 : logo du vCloud de VMware</i>	33
<i>Illustration 2.4 : logo EC2 d'Amazon</i>	33
<i>Illustration 2.5 : logo d'Eucalyptus</i>	33
<i>Illustration 2.6 : logo de Open Stack</i>	34
<i>Illustration 2.7 : logo de OpenNebula</i>	34
<i>Illustration 2.8 : logo de XCP</i>	35
<i>Illustration 5.1 : Ecran principal de XenServer</i>	63
<i>Illustration 5.2 : Ecran d'accueil de XenCenter</i>	63
<i>Illustration 5.3 : interface de propriétés de XenServer</i>	64
<i>Illustration 5.4: Accès au bureau distant via XenCenter</i>	65
<i>Illustration 5.5 : Interface de supervision des performances de XenServer</i>	65
<i>Illustration 5.6 : Interface de supervision de la mémoire</i>	65
<i>Illustration 5.7 : Ecran d'accueil du centre de management Citrix Studio</i>	66
<i>Illustration 5.8 : Ecran de Citrix Studio configuré</i>	66
<i>Illustration 5.9 : Ajout d'un compte à Citrix Receiver</i>	67
<i>Illustration 5.10 : Ecran de connexion à Citrix Receiver</i>	67
<i>Illustration 5.11 : Ecran d'accueil de Citrix Studio</i>	68
<i>Illustration 5.12 : Ecran de démarrage de la machine virtuelle</i>	68
<i>Illustration 5.13 : Accession à la machine virtuelle</i>	69
<i>Illustration A.2 : Logique de l'architecture de XenOrchestra</i>	82
<i>Illustration A.3 : volumes répliqués DRBD au pool de serveurs Xen</i>	89
<i>Illustration A.4: migration du master</i>	90

INTRODUCTION GENERALE

Actuellement internet connaît une évolution sans précédent ce qui entraîne également celle des technologies basées sur lui. Et parmi ces technologies on a le Cloud Computing. La montée en force de ce dernier a fait que presque tout le monde (entreprise, particulier, individu...) utilise le Cloud Computing parfois même sans le savoir. A titre d'exemple plus de 960 million de personnes chargent au moins une photo dans le Cloud de Facebook chaque jour d'après une étude faite à ce sujet (voir les références [1] [2]) et ces personnes font ce qu'on appelle du Cloud de stockage car elles pourront avoir accès à leurs photos n'importe quand et de n'importe où tant qu'elles ont une connexion internet.

De plus les technologies de l'information et de la communication révolutionnent également nos modes de vie et de travail. Le Cloud Computing ou informatique virtuelle ou encore nuage informatique, est apparu ces dernières années comme un nouveau modèle de gestion et d'utilisation des systèmes informatiques. Le concept consiste à déporter sur des serveurs distants les traitements et stockages habituellement effectués en local afin d'y accéder sous forme de service.

En fonction des besoins, les services du Cloud Computing peuvent s'étendre du simple approvisionnement de machines virtuelles, services IAAS (Infrastructure as a service) aux services de type applications SAAS (Software as a service). Plusieurs modes de déploiement de ces services peuvent être employés selon l'institut national des normes et de la technologie des Etats-Unis (National Institute of Standard and Technology : NIST) parmi lesquels le mode public et le mode privé.

Déployés en public, les services du Cloud Computing permettent d'offrir aux utilisateurs des applications ou systèmes informatiques sous forme de service accessible via l'Internet. Ils permettent aux utilisateurs de s'affranchir des tâches de déploiement et d'administration des systèmes informatiques complexes en local, en leur offrant à ces derniers suivant un mode de paiement à l'usage. Toutefois, selon Galán et al (2009), les offres publiques du Cloud Computing sont sujettes à des contrôles de type vendor lock-in qui empêchent les utilisateurs de profiter pleinement de l'architecture. Des questions de sécurité et de réseau, soulevées par les auteurs tels que Ginovsky (2011), Subashini et Kavitha (2011), constituent également des facteurs limitant l'adoption des services publics du Cloud Computing.

En privé, le Cloud Computing permet d'optimiser et de faciliter l'utilisation et l'administration des ressources informatiques internes, généralement distribuées sur un réseau local, en offrant une haute disponibilité de ces dernières, accessibles de façon dynamique via une plateforme Web. Ce mode de déploiement apporte de nombreux avantages en termes de coût de gestion et de temps d'administration des systèmes informatiques. Si les solutions du Cloud Computing privé constituent une alternative aux services publics en termes de sécurité et de réseau, elles adoptent cependant des architectures centralisées de supervision des ressources du système et d'exposition des services aux utilisateurs (Nurmi et al 2008 ; Sempolinski et Thain 2010; Alrwais, 2011) ; ce qui est susceptible de conduire à des ruptures de service en cas de panne du nœud central du système (Naing, 2012). Il est donc nécessaire, en vue d'implémenter un système de Cloud Computing privé efficace, de concevoir une solution qui intègre la tolérance aux pannes en offrant à tous les hôtes du réseau la possibilité d'être élus pour les différentes tâches de supervision afin d'assurer la continuité des services en cas de panne d'un nœud quelconque du système. C'est dans cette optique que s'inscrit le présent projet de fin d'étude qui vise la Conception d'une solution de Cloud Computing privé basée sur un algorithme de supervision distribué. Nous avons appliqué cette solution aux services IAAS et l'avons contextualisé au cas de MATRIX TELECOMS.

Le présent mémoire rédigé, en guise de synthèse du travail effectué dans ce contexte, s'organise en cinq chapitres. Dans le premier chapitre nous allons parler du contexte et de la problématique tout en présentant l'entreprise d'accueil ; ensuite au chapitres deux et trois nous présenterons les généralités sur le Cloud Computing et les algorithmes de supervision distribués ; au chapitre quatre nous ferons l'implémentation de la solution retenue et nous présenterons les choix techniques et enfin au chapitre cinq nous allons présenter les résultats obtenus et les commenter.

CHAPITRE I : CONTEXTE ET PROBLEMATIQUE

Introduction :

Nous ne pouvons entrer dans le vif du sujet sans faire une présentation de l'entreprise, de son architecture existant, des services qu'il offre à ses clients et de ses missions (voir à la référence [3]). Puis nous n'allons pas manquer pas de présenter le contexte, la problématique et les objectifs que nous voulons atteindre.

I.1 Présentation de l'entreprise

I.1.1 Historique de Matrix Télécoms

I.1.1.1 L'origine

Au tout début était la société ICC (International Computer Center) créée le 10 mars 1995 et spécialisée dans l'ingénierie informatique ; après deux années c'est-à-dire en 1997 plus précisément le 17 novembre, ICC se lança dans la commercialisation de l'accès et des services internet et devint ICCNET (ICC NETwork). Plus tard on parlera de Groupe ICCNET. (Voir à la référence [3]).

I.1.1.2 Groupe ICCNET

Il s'agit d'un groupe de service informatique et internet qui se structurait autour de trois entreprises autonomes et complètement spécialisées dans une activité bien précise :

- **ICCSOFT** (ICC SOFTware) : entreprise du groupe spécialisée dans le développement des applications logicielles, web et hébergement des sites web.
- **RESYTAL** (REseaux, Systèmes, TéLécoms, Audits et formation) : entreprise du groupe spécialisée dans la sécurité des réseaux et systèmes informatique, l'audit de sécurité des réseaux, écriture des chartes et politiques de sécurité, télécommunication ainsi que dans la formation.
- **ICCNET SA** : spécialisée dans les services internet surtout la fourniture d'accès internet.

I.1.1.3 La naissance de matrix télécoms

Fragilisé par l'arrivée des géants aux ressources importantes comme MTN et Orange, ICCNET va en juillet 2006 fusionner avec deux autres fournisseurs d'accès à internet maintenant appelés **Douala One.com** et **CREOLINK Communications** : c'est

la naissance de **MATRIX Télécoms** avec pour administrateur directeur général (ADG) **Clovis TCHOKONTE** pour le compte cumulé de **ICCNET** et **Douala One.com**, et **Joseph MBOCK** pour le compte de **CREOLINK**.

Société anonyme de droit camerounais au capital de 950 000 000 FCFA **MATRIX Télécoms** sera géré tant bien que mal jusqu'au retrait de CREOLINK six mois seulement après la fusion soit en janvier 2007. Par la suite, Douala One.com se retirera à son tour en avril 2007 soit neuf mois après la fusion.

Après le départ de ses uniques coactionnaires, Clovis TCHOKONTE, ancien directeur de ICCNET, décide de conserver la dénomination de MATRIX Télécoms, lui permettant ainsi d'exploiter certaines licences telles que celle de la VOip, du VPN, de la vidéoconférence, le service internet et de la télésurveillance. MATRIX Télécoms verra ainsi ses activités se développer au point de se constituer aujourd'hui un capital social d'à peu près 317 000 000 FCFA.

I.1.2 Fiche signalétique, organigramme et situation géographique de matrix

I.1.2.1 Fiche de signalétique de MATRIX Télécoms

Tableau 1.1 : fiche signalétique de MATRIX Télécoms (voir à la référence [3])

Raison social	MATRIX TELECOMS
Statut juridique	Société Anonyme
Capital	317 000 000 FCFA
Effectif	90 employés au 1 ^{er} mars 2006
Registre du commerce	Registre du commerce et du crédit mobilier
Registre du commerce	C
Siège social	Immeuble MATRIX TELECOMS, Omnisport, route de Ngousso, Yaoundé
Contacts	BP : 4124 Yaoundé Tel : (+237) 222 21 26 11 E-mail : info@matrixtelecoms.com Site web : www.matrixtelecoms.com
Succursale de Yaoundé	Immeuble MATRIC TELECOMS, Omnisport, route de Ngousso

Succursale de Douala	Rond-point salle des fêtes d'Akwa
Centre des opérations techniques	Immeuble Rond-point salle des fêtes d'Akwa, Douala
Président Directeur Général	Clovis TCHOKONTE
Directeur Général Adjoint	NANA Durand
Pôles d'activité	ISP – Opérateur Virtuel de Télécommunications
Technologies utilisées	Wimax SRT, vecima, canopy, NANO Station, Mikrotik

I.1.2.2 Organigramme

Il faut savoir que MATRIX est une SA hiérarchisée découpée en plusieurs services. Le stage que nous effectuons se déroule dans le service chargé de la connectivité et des infrastructures (CSG : Connectivity Service Group) dont la mission principale est d'assurer la connectivité et la maintenance des infrastructures du réseau de MATRIX Télécom et le suivi de ses clients.

La figure suivante montre la hiérarchisation de cette structure :

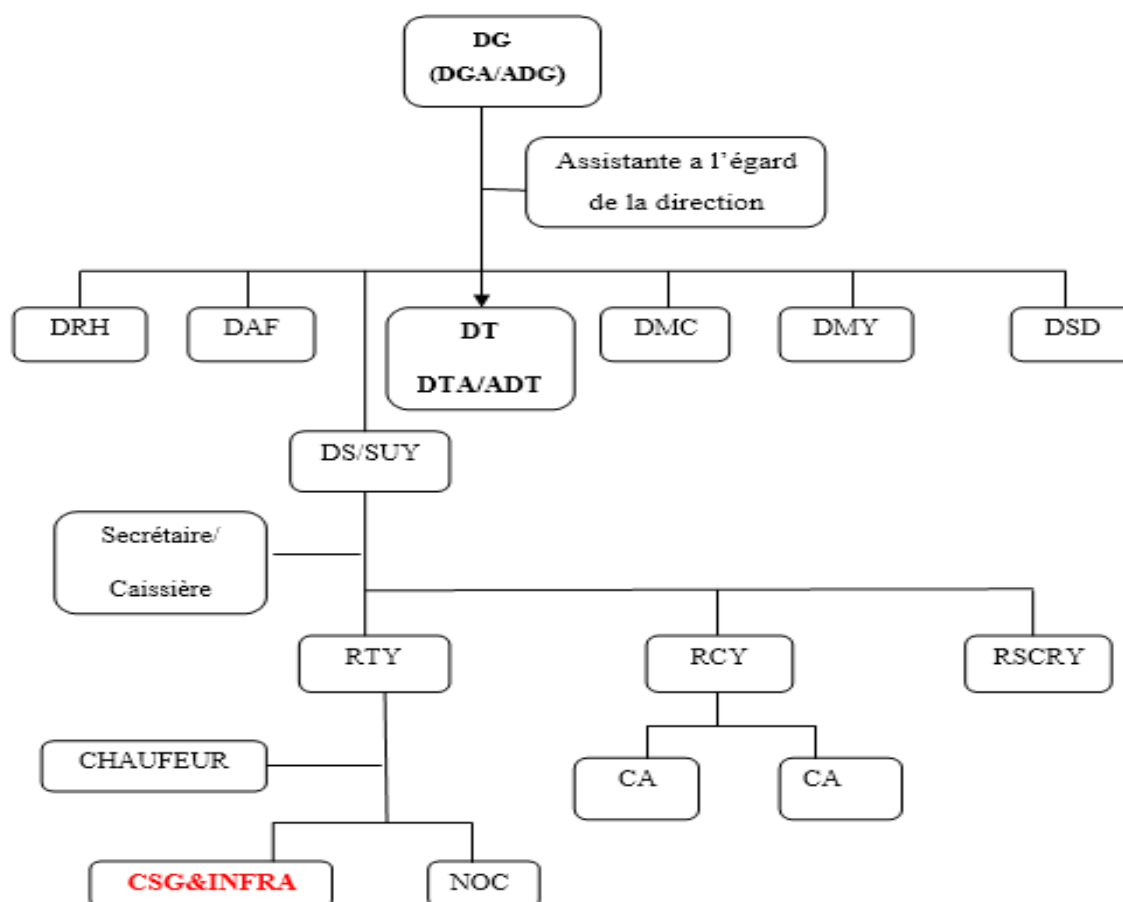


Figure 1.1 : Organigramme de MATRIX Télécoms (voir à la référence [3])

I.1.2.3 Localisation

Nous avons effectué notre stage dans la succursale de Yaoundé située à l'immeuble MATRIX TELECOMS dans le quartier Omnisport sur la route de Ngousso. Voici une illustration de sa localisation :

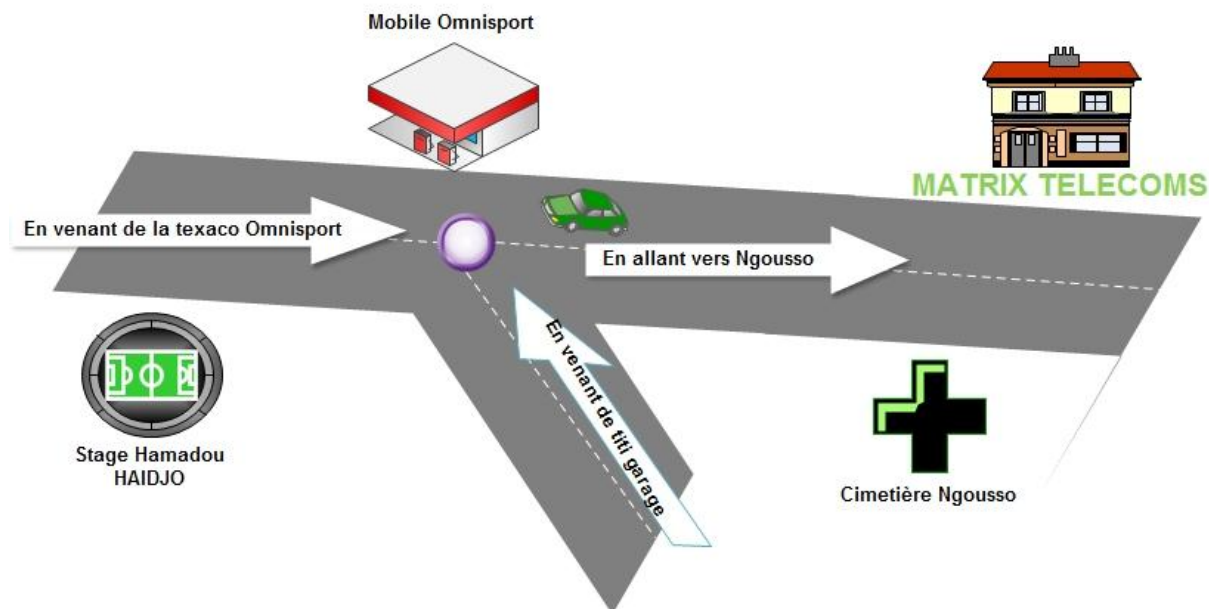


Figure 1.2 : Plan de localisation de MATRIX Télécoms

I.1.3 Services offerts, architecture et missions de MATRIX TELECOMS

I.1.3.1 Les services offerts

Matrix Télécoms est un fournisseur d'accès internet de niveau 2 qui offre des services à sa clientèle parmi lesquels :

- Accès internet pour entreprises et grand public principalement, par onde Radio (WIMAX, Canopy, vecima, SR Telecom, Nano station).
- Messagerie d'entreprise
- Hébergement des sites web
- Téléphonie sur IP
 - Appels gratuits dans les réseaux Matrix Télécoms
 - Appels à l'international
 - Appels sur les réseaux fixes/CDMA et GSM
 - Fax via le réseau internet
- Interconnexion des sites distants par VPN
 - VPN d'entreprises

- MATRIX Virtual Office
- Services vidéo
 - Vidéo conférence

I.1.3.2 Architecture

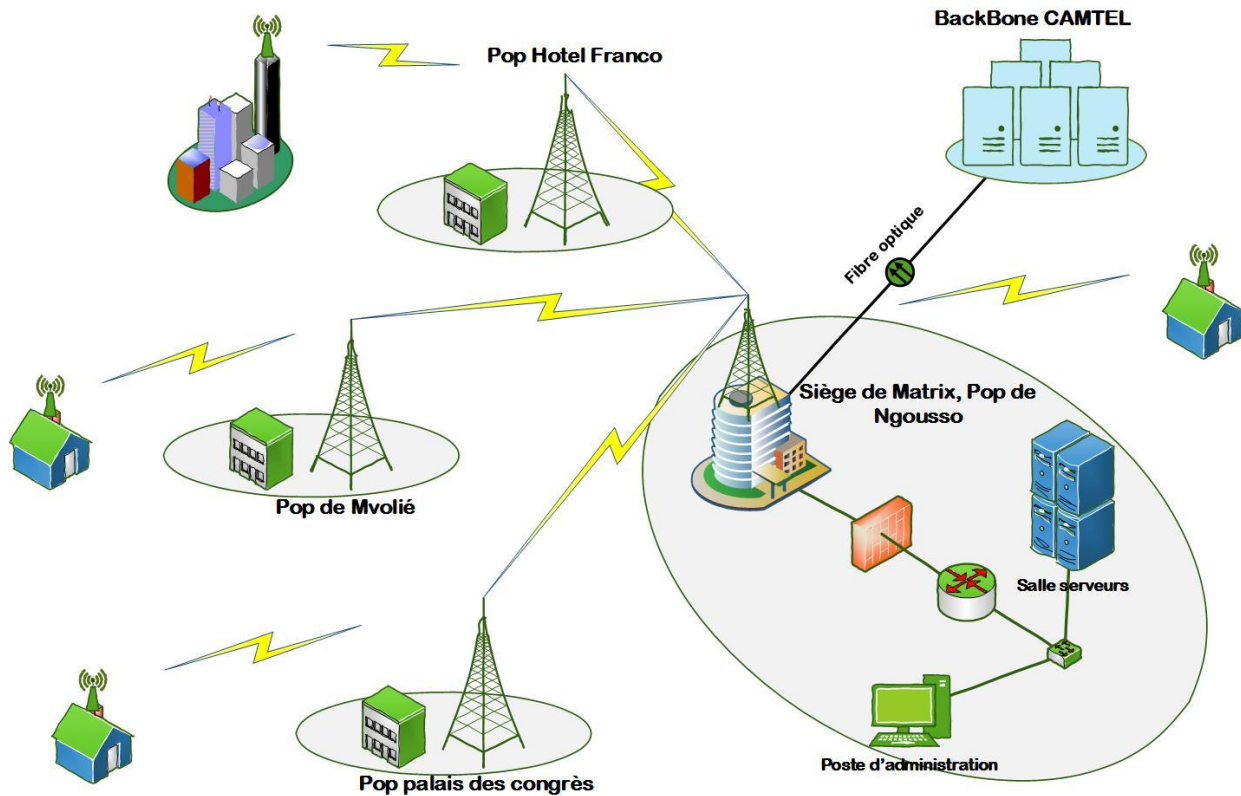


Figure 1.3 : Architecture réseau cœur de MATRIX Télécoms (voir référence [3])

I.1.3.3 Missions

La principale mission de MATRIX Télécoms est d'offrir des solutions de télécommunications de qualités avec professionnalisme et objectivité afin de répondre aux besoins spécifiques des entreprises et organisations.

Pour ce faire, elle procède par l'étude, l'installation, l'exploitation et l'entretien de toutes les infrastructures nécessaires à la fourniture des services de télécommunications sur l'ensemble du territoire national.

Le slogan de MATRIX Télécoms est « **connecting people, inspiring solution** ». Il traduit la vision de l'activité des télécoms. Par ceci, MATRIX Télécoms permet aux entreprises de communiquer entre elles et de se connecter au réseau mondial par des solutions et services IP très innovantes. Le développement de MATRIX Télécoms s'appuie sur une technologie de pointe (fibre optique) et un réseau de distribution à

l'échelle nationale pour des services internet et de téléphonie sur IP. (Voir à la référence numéro [4])

I.2 Contexte et problématique

Après les différents acquittements judiciaires, une entreprise nouvellement créée doit pour se développer avoir :

- Des ordinateurs et/ou des serveurs
- Des switches, des routeurs
- Des dizaines de centaines de mètres de câbles réseaux
- Des ingénieurs dans plusieurs domaines
- De la puissance de calculs
- De logiciels
- Plateformes web
- Etc.

Cette liste est non exhaustive et est encore bien longue. Il est important de signaler que l'installation d'un LAN et la configuration d'un certains nombres de services sont nécessaires, ce qui prend du temps et empêche l'entreprise de se lancer dans la réalisation de ses objectifs.

De plus d'après la directrice des Affaires financières de MATRIX TELECOMS Mme MONKA Freddy, la structure dépense en moyenne 1 500 000 FCFA par an pour l'achat des laptops et des logiciels destinés aux employés ainsi que pour leur maintenance.

Il est donc nécessaire de trouver une alternative pour gagner en temps, en productivité et surtout gagner en argent : c'est là qu'intervient le Cloud Computing.

Ainsi inutile d'avoir :

- Un ordinateur qui coûte cher et un système de stockage sécurisé ;
- Pas de frais de maintenance des équipements informatiques ;
- Pas de logiciels coûteux avec des licences très chères ;
- L'informatique dématérialisée fonctionne en mode web avec un accès aux services utilisant un simple navigateur web et de plus en plus avec des systèmes d'exploitation libres.

Néanmoins, on a constaté que l'informatique dématérialisée nécessitait les éléments suivants :

- Connexions Internet de bonne qualité ;
- Marché porteur pour les logiciels et les matériels ;
- Nécessaire de faire la virtualisation des applications ;
- Confiance dans la sécurité des systèmes utilisés ;
- Localisation des données.

I.3 Méthodologie

Pour résoudre cette problématique nous devons passer par plusieurs étapes dont les principales sont les suivantes :

- Mise en place d'un Cloud privé IAAS dont les services fournis seront des machines virtuelles.
- Mise en place d'un système de réplication sur un pool de serveurs
- Installation des machines virtuelles
- Mise en place d'une plateforme distribuée de supervision
- Mise en place d'un serveur de bureaux distants

A la fin ces étapes nous serons en mesure d'assurer nos objectifs.

I.4 Objectifs

L'objectif principal de ce projet consiste à la mise en place d'un Cloud d'entreprise qui devra permettre la gestion d'un ensemble de machines virtuelles réparties sur plusieurs hôtes en parlant du service IAAS notamment. Le but dans un Cloud privé est de pouvoir garantir un certain nombre de critères :

- La haute disponibilité des services (assurer la continuité) grâce à un algorithme.
- Permettre aux utilisateurs d'accéder à leur machine virtuelle via un système simple et sécurisé
- Permettre à l'administrateur de superviser les serveurs ainsi que les VMs de manière simple.

Conclusion :

Au terme de ce chapitre nous pouvons comprendre que MATRIX Télécoms est une grande entreprise qui a tout intérêt à mettre sur pied une solution de ce genre. Dans le but d'être plus compétitif, plus efficace et surtout de faire des économies. De plus les objectifs que nous devons atteindre contiennent de nombreux sous objectifs ce qui augmente la complexité du projet. Nous allons à présent entrer dans le vif du sujet et parler des concepts généraux du Cloud Computing.

CHAPITRE II : GENERALITES SUR LE CLOUD COMPUTING

Introduction :

Le Cloud Computing est un concept nouveau qui veut dire en français nuage* informatique. Ce concept a été pour la première fois prononcé par John McCarthy en 1960, et dont le principe est de transformer le matériel, les équipements et les installations informatique en des services que l'on peut offrir aux clients. En un mot on peut parler de dématérialisation. Pour mieux cerner ce qu'est le Cloud Computing il est important de parler plus en profondeur de ce nouveau concept, puis de ses technologies qu'il offre et enfin des solutions pour le mettre sur pieds.

II.1 Les concepts du Cloud Computing

II.1.1 Définition

Il existe de nombreuses approches pour définir ce qu'est le Cloud Computing. Mais celle qui est a été adoptée et qui est utilisée dans de nombreux ouvrages est celle du National Institute of Standard and Technology (NIST) et du Groupe spécialisé de l'UIT qui stipule que : « Le Cloud Computing est un modèle qui offre aux utilisateurs du réseau un accès à la demande, à un ensemble de ressources informatiques partagées et configurables, et qui peuvent être rapidement mises à la disposition du client sans interaction direct avec le prestataire de service. »

Les Services Cloud Computing ont des caractéristiques qui les distinguent des autres technologies :

- En général, les utilisateurs du Cloud Computing ne sont pas propriétaires des ressources informatiques qu'ils utilisent. Les serveurs qu'ils exploitent sont hébergés dans des data centres externes.
- Les services sont fournis selon le modèle pay-per-use* ou le modèle d'abonnement.
- Les ressources et les services fournis aux clients sont souvent virtuels et partagés par plusieurs utilisateurs.
- Les services sont fournis via l'Internet.

L'étoile (*) indique que le mot est expliqué au glossaire

Ces spécificités font de la technologie Cloud Computing une nouvelle option qui offre à ses utilisateurs la possibilité d'accès à des logiciels et à des ressources informatiques avec la flexibilité et la modularité souhaitées et à des coûts très compétitifs. L'accès à ces ressources se faisant par la suite d'une façon simple avec à l'aide d'une plateforme Web et de d'un équipement.

II.1.2 Les services du Cloud

Le Cloud Computing propose plusieurs types de services mais on compte généralement trois :

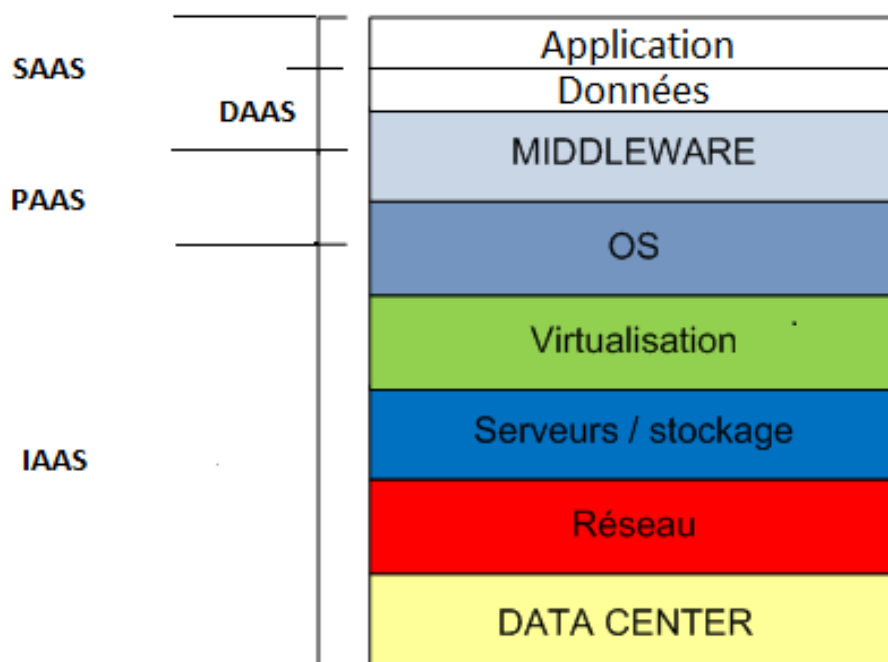


Figure 2.1: Niveaux de service Cloud (voir à la référence [5])

II.1.2.1 Infrastructure as a Service (IAAS)

L'**infrastructure en tant que service** ou, en anglais, *infrastructure as a service* (**IAAS**) est un modèle de Cloud Computing destiné aux entreprises où :

- L'entreprise gère : le middleware* des serveurs, et surtout les logiciels applicatifs (exécutables, paramétrages, l'intégration SOA*, les bases de données),
- Le fournisseur cloud gère : le matériel serveur, les systèmes d'exploitation, les couches de virtualisation, le stockage, les réseaux.

Les services IAAS du Cloud Computing permettent de mettre à la disposition des utilisateurs des ressources matérielles (réseau, stockage, systèmes d'exploitation)

L'étoile (*) indique que le mot est expliqué au glossaire

accessibles sous forme virtuelle. Ces ressources sont généralement de type serveur équivalant à des machines physiques pouvant être utilisées pour des opérations telles que l'hébergement, le test d'applications ou les travaux de recherche en réseau informatique. L'exemple le plus souvent cité est la plateforme EC2 (Elastic Compute Cloud) d'Amazon. (Voir à la référence [6])

L'intérêt consiste à faciliter le déploiement de nouvelles applications pour les utilisateurs, uniformiser les environnements de travail et accroître la sécurité des systèmes. Les ressources virtuelles des services IAAS sont généralement accessibles grâce aux protocoles de prise de contrôle à distance tels que SSH (Secure Shell), VNC (Virtual Network Computing) ou encore le protocole RDP (Remote Desktop Protocol). Les éléments constitutifs des services IAAS du Cloud Computing selon Sun et al (2011) sont :

- **Le centre d'administration du cloud** : C'est l'interface livrée aux utilisateurs, accessible via un portail Web. Il permet l'utilisation des ressources du cloud mais aussi leur administration. Le centre d'administration du cloud IAAS reçoit les requêtes de la part des utilisateurs et crée les ressources demandées à leur endroit.
- **Le centre des ressources du cloud** : Il est composé des ressources physiques utilisées pour l'exécution des services du cloud. Ces machines physiques représentent généralement les hôtes utilisées pour accueillir et exécuter les machines virtuelles de l'infrastructure. Les hôtes ainsi que les services qu'ils exécutent à travers les machines virtuelles sont tous administrés depuis le centre d'administration du cloud. Voir à la référence [7].
- **Le centre de stockage du cloud** : Il est composé des machines physiques servant d'espaces de stockages des images systèmes et fichiers de configuration des machines virtuelles du cloud.

Afin de montrer les suites d'opérations menées par un utilisateur des services IAAS du Cloud Computing, Sun et al (2011) numérotent les opérations effectuées durant le processus de 1 à 9 (voir figure II.3).

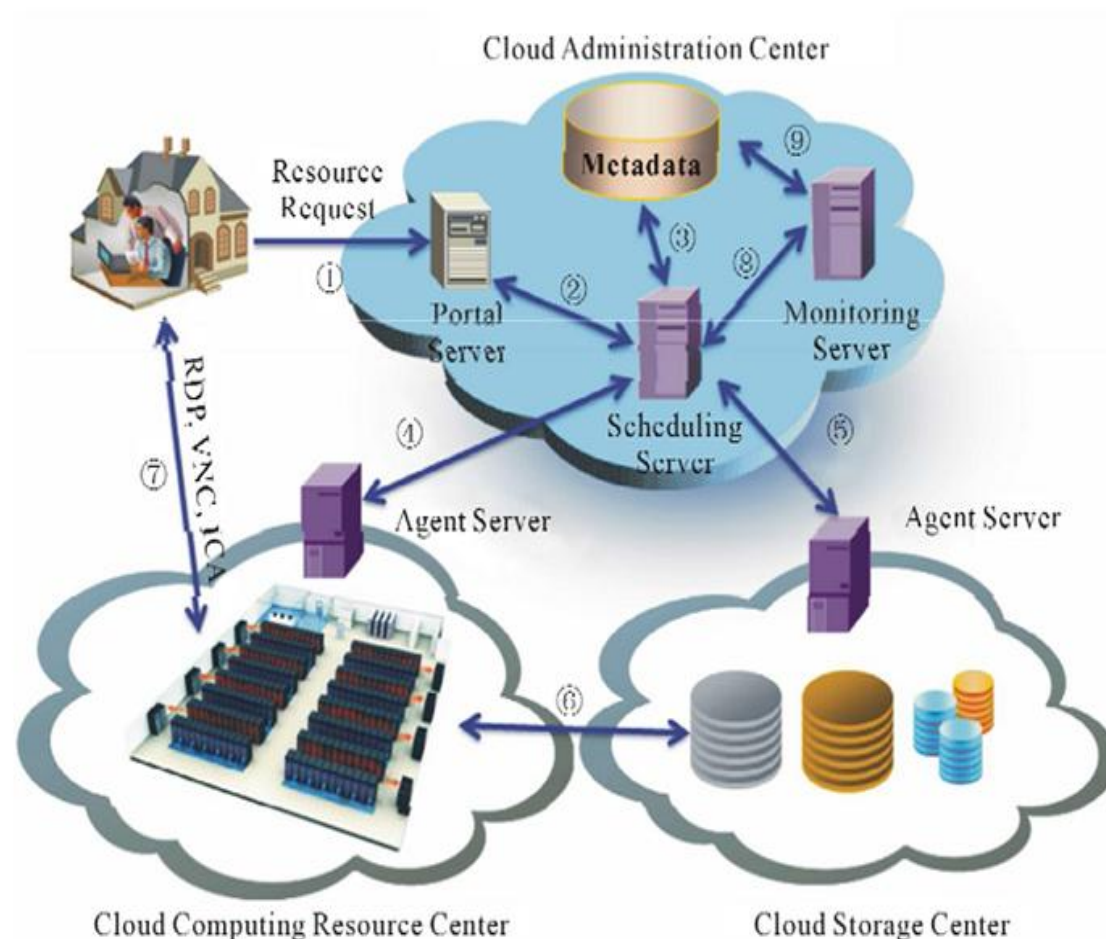


Figure 2.2 : Les éléments constitutifs des services IAAS du Cloud Computing (Sun et al, 2011). (Voir à la référence [8])

- 1.) L'utilisateur émet une requête d'utilisation de ressource à l'endroit du portail des services.
- 2.) Le portail des services (généralement une application Web) renvoie la requête vers le serveur d'orchestration*.
- 3.) Le serveur d'orchestration recherche une machine physique adéquate pour créer et exécuter la machine virtuelle demandée. Cette recherche se base sur les caractéristiques des machines physiques du réseau.
- 4.) Le serveur d'orchestration choisit une machine physique de performance optimale et envoie la commande de création de machine virtuelle.
- 5.) Le serveur d'orchestration choisit un modèle de machine virtuelle à partir du centre de stockage et l'envoie à la machine physique précédemment sélectionnée.
- 6.) Le serveur d'orchestration démarrera la machine virtuelle sur l'hôte physique si la machine est correctement chargée. Dans le cas contraire, une autre machine est sélectionnée à l'étape 4.

L'étoile (*) indique que le mot est expliqué au glossaire

7.) Si la machine virtuelle démarre correctement, l'utilisateur peut y accéder via les protocoles d'accès à distance tels que RDP, VNC, ou SSH.

Les étapes 8 et 9 concernent la mise à jour des informations du système et des services de supervision.

II.1.2.2 Platform as a service (PAAS)

Platform as a Service, souvent appelé simplement PAAS, est une catégorie de services de Cloud Computing qui fournit la plateforme et l'environnement informatique nécessaire aux développeurs pour mettre en place leurs différents services et applications sur Internet.

Les services PAAS permettent aux utilisateurs de créer des applications logicielles en utilisant les outils fournis par le fournisseur. Ils peuvent prendre la forme de fonctionnalités préconfigurées auxquelles les clients peuvent souscrire, en ne choisissant que celles qui conviennent à leurs exigences. Cela signifie que les packs PAAS peuvent aller de la simple offre "point-and-click", où le client n'a pas besoin d'avoir des connaissances particulières en hébergement, à la fourniture d'options d'infrastructure pour développement avancé. **Google App Engine, Microsoft Azure** sont des exemples de services PAAS. (Voir aux références [9] [10])

II.1.2.3 Software as a service (SAAS)

Le **logiciel en tant que service** ou *software as a service* (SAAS) est un modèle d'exploitation commerciale des logiciels dans lequel ceux-ci sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur. Les clients ne paient pas de licence d'utilisation pour une version, mais utilisent librement le service en ligne ou le plus généralement payent un abonnement.

Les services SAAS du Cloud Computing permettent de mettre à la disposition des utilisateurs des applications prêtes à l'emploi. A la différence des applications Web ordinaires, les services SAAS du Cloud Computing sont caractérisés par un haut niveau d'abstraction qui permet d'adapter l'application à un cas particulier d'usage. Il s'agit entre autres des applications de gestion de la relation clients (CRM), des applications de planifications de ressources d'entreprises (ERP) ou des applications de messagerie électronique. Parmi les solutions SAAS existantes, on peut noter comme exemples la suite bureautique Google Apps for Business en mode SAAS, les services de messagerie électronique tel que Gmail, Odoo, SAGE, etc. (Voir aux références [9] [10])

II.1.2.4 Les autres services

En dehors des trois principaux services cités en amont, le Cloud Computing offre deux autres services récemment créés à savoir :

- La communication en tant que service (CAAS : Communication as a Service) : service de communication audio/vidéo, services collaboratifs, communications unifiées, messagerie électronique, messagerie instantanée partage de donnée (web conférence). (Voir à la référence [9] [10])
- Le réseau en tant que service (NAAS : Network as a Service) : Internet managé (garantie du débit, disponibilité, etc.), réseaux virtualisés VPN couplée aux services Cloud Computing, bande passante flexible et à la demande. (Voir aux références [9] [10])
- La donnée en tant que service (DAAS : Data as a service), est une technique consistant à faire payer un abonnement pour l'accès à un dépôt de données via une interface fournie par le fournisseur¹. Comme tous les membres de la famille des "*as a Service*" (AAS), elle est basée sur le concept où le produit, les données dans ce cas, peut être fourni à l'utilisateur sur demande quelle que soit la distance entre l'utilisateur et le fournisseur des données. [11]

II.1.3 Les modèles de déploiement

Il existe quatre façons pour déployer un Cloud Computing selon la position géographique des serveurs et selon les services offerts :

II.1.3.1 Le modèle public

Il s'agit ici d'un Cloud mutualisé ouvert à tous, tels que ceux d'Amazon, Microsoft, etc. Le Cloud Computing public représente les offres du cloud livrées, par des tiers fournisseurs, aux entreprises dans un mode de **payement à l'usage**.

Il permet ainsi aux entreprises qui bénéficient de ce type de déploiement de s'affranchir des difficultés liées au déploiement des infrastructures informatiques complexes en interne. La gestion des infrastructures est ainsi confiée aux fournisseurs de services et le payement se fait à l'usage par les clients. Les services sont fournis via l'Internet et les utilisateurs y accèdent à volonté. (Voir aux références [9] [10] [12])

II.1.3.2 Le modèle privé

Ce modèle est monté pour une utilisation dédiée à un unique client. Les Clouds privés internes, gérés en interne par une entreprise pour ses besoins, les clouds privés externes, dédiés aux besoins propres d'une seule entreprise, mais dont la gestion est externalisée chez un prestataire.

Le cloud privé permet la maximisation de l'utilisation des ressources disponibles grâce aux techniques de la virtualisation employée (Dillon et al, 2010). Le cloud privé offre également la flexibilité et l'agilité dans l'exploitation des ressources informatiques disponibles.

Dans les environnements de forte utilisation des systèmes informatiques tels que les universités ou les grandes entreprises, il est également utile de disposer de l'infrastructure en privé. Dans le cas des universités par exemple, les machines virtuelles du service IAAS sont octroyées dynamiquement aux étudiants pour accomplir les tâches d'expérimentation en réseau informatique. L'université de Caroline du Nord des Etats-Unis (North Carolina State University) emploie le cloud privé IAAS à travers son projet VCL (Virtual Computing Laboratory) (Averitt et al, 2007). (Voir aux références [9] [10] [12])

II.1.3.3 Le modèle hybride

Ici on utilise une infrastructure faisant appel à la fois à des serveurs sur un Cloud public et sur un Cloud privé. Dans ce type d'architecture, les ressources critiques du client sont hébergées en local tandis que celles louées auprès des fournisseurs sont accédées à distance via l'Internet ou un réseau privé virtuel. (Voir à la référence [6])

II.1.3.4 Le modèle communautaire

Un Cloud communautaire est utilisé par plusieurs organisations qui ont des besoins communs. Ainsi cela peut porter sur l'hébergement d'une application métier très spécialisée, mais commune à de très nombreuses entreprises, qui décident de fédérer leurs efforts. (Ex : Amadeus, premier acteur mondial dans le domaine des voyages - CMed, Cloud pour les laboratoires pharmaceutiques).

Dans une telle architecture, l'administration du système peut être effectuée par l'une ou plusieurs des organisations partageant les ressources du Cloud. (Voir à la référence [10])

II.2 Les technologies du Cloud Computing

Les solutions du Cloud Computing se basent sur quelques technologies indispensables. Il s'agit de la **virtualisation** et des **technologies du Web**. La virtualisation permet de fournir les ressources nécessaires au support des applications du cloud. Elle est indispensable dans la mise en place des services IAAS en pourvoyant aux utilisateurs les machines virtuelles disponibles. Les technologies du Web permettent d'un autre côté de livrer les services aux utilisateurs à travers des portails Web. Elles permettent également d'assurer l'interopérabilité des différentes solutions existantes en fournissant des méthodes de communication de haut niveau telles que les services Web.

II.2.1 La virtualisation

La virtualisation, c'est la possibilité d'exécuter plusieurs systèmes d'exploitation ou applications sur un même matériel, chacun de ces systèmes d'exploitation ou applications n'ayant pas conscience de partager le matériel.

Le système d'exploitation ainsi virtualisé est appelé système invité (**guest system**) et le système d'exploitation servant d'accueil s'appelle le système hôte (**host system**). La machine physique utilisée est appelée **machine hôte** tandis que celle émulée au système invité est la **machine virtuelle**. Une machine virtuelle peut donc être vue comme un logiciel au même titre qu'un navigateur Web qui fournit au système d'exploitation invité un environnement similaire à celui d'une machine physique. La figure 2.4 illustre le principe de la virtualisation (voir à la référence [13])

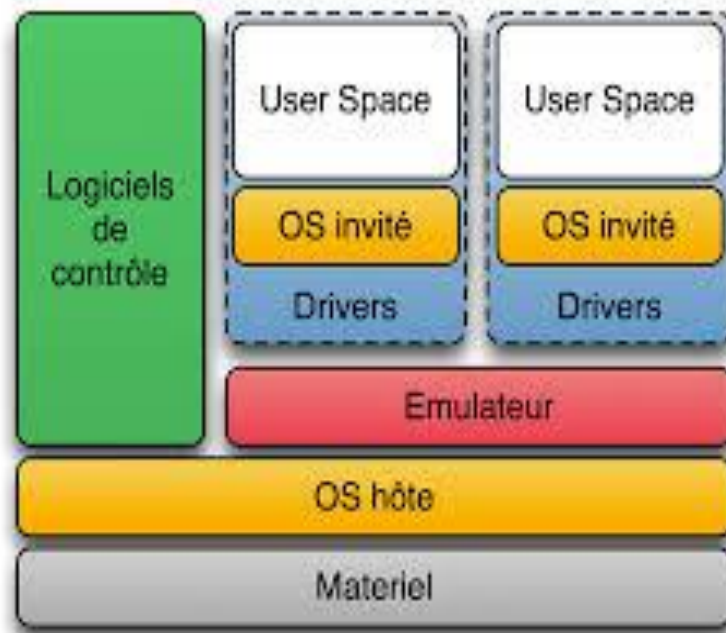


Figure 2.3 : Principe de la virtualisation (voir à la référence [14])

Ainsi comparée à un simple logiciel (ici un navigateur Web), une machine virtuelle est constituée d'un ensemble de fichiers gérés par le système hôte.

En pratique, le disque dur de la machine virtuelle est la plupart du temps géré comme un fichier volumineux pour le système hôte, alors que la mémoire vive dont le système invité dispose est réservée par le programme de la machine virtuelle.

Il existe plusieurs catégories de virtualisation utilisant chacune des technologies différentes (voir à la référence [15])

II.2.1.1 OpenVZ

OpenVZ est une technique de virtualisation de niveau système d'exploitation basée sur le noyau Linux. Cette technique de virtualisation de niveau système d'exploitation est souvent appelée **conteneurisation** et les instances sont appelées **conteneur**. Il permet à un serveur physique d'exécuter de multiples instances de systèmes d'exploitation isolés, on parle d'**isolation** ou **cloisonnement**. Voir la référence [13].

L'isolation consiste donc à diviser un système d'exploitation en plusieurs espaces ou contextes. Chaque contexte est géré par le système d'exploitation hôte comme un processus isolé dont il est théoriquement impossible d'en sortir. Avec l'isolation, l'espace noyau n'est pas différencié, il est unique, partagé entre les différents contextes.

Le partage du même noyau limite cette technologie aux environnements de mêmes types. On ne peut faire cohabiter que des distributions issues d'un même noyau. Voir la figure 2.4 :

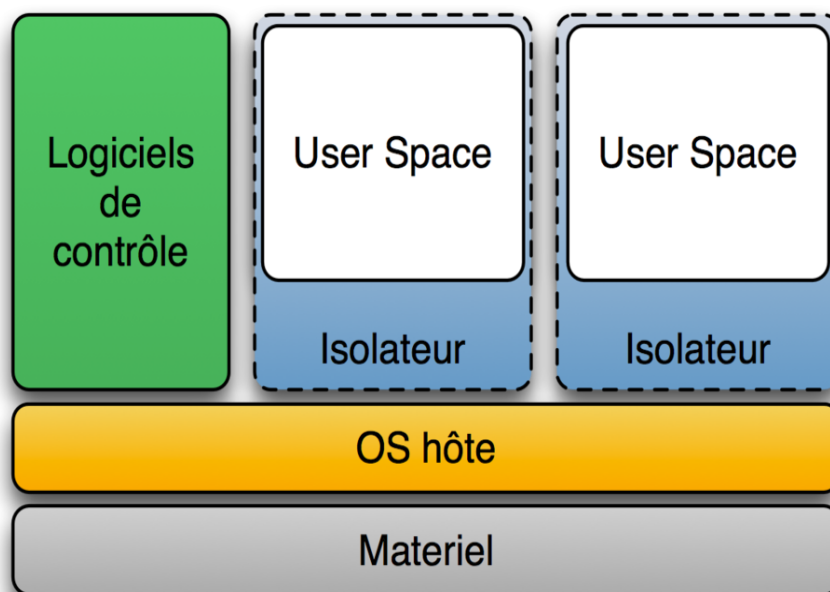


Figure 2.4 : principe de l'isolation (voir la référence [16])

OpenVZ offre moins de flexibilité dans le choix du système d'exploitation que les autres techniques de virtualisation.

II.2.1.2 La virtualisation complète

Elle s'appuie sur une conversion binaire afin d'intercepter et de virtualiser l'exécution de certaines instructions sensibles non virtualisables. Les OS invités et leurs applications sont constitués d'instructions critiques et non critiques. Sur un système basé sur un hôte, tant l'OS hôte que l'OS invité est utilisés. Une couche logicielle de virtualisation est intercalée entre les deux OS. Voir la référence [17].

Avec la virtualisation complète, les instructions non critiques s'exécutent directement sur le matériel, alors que les instructions critiques sont détectées et remplacées par des interceptions dans le VMM, afin d'être émulées* par le logiciel. Par conséquent, la virtualisation complète associe la conversion binaire et l'exécution directe. L'OS invité est entièrement découplé du matériel sous-jacent et n'a donc pas conscience de ce qui est virtualisé (voir figure 2.5).

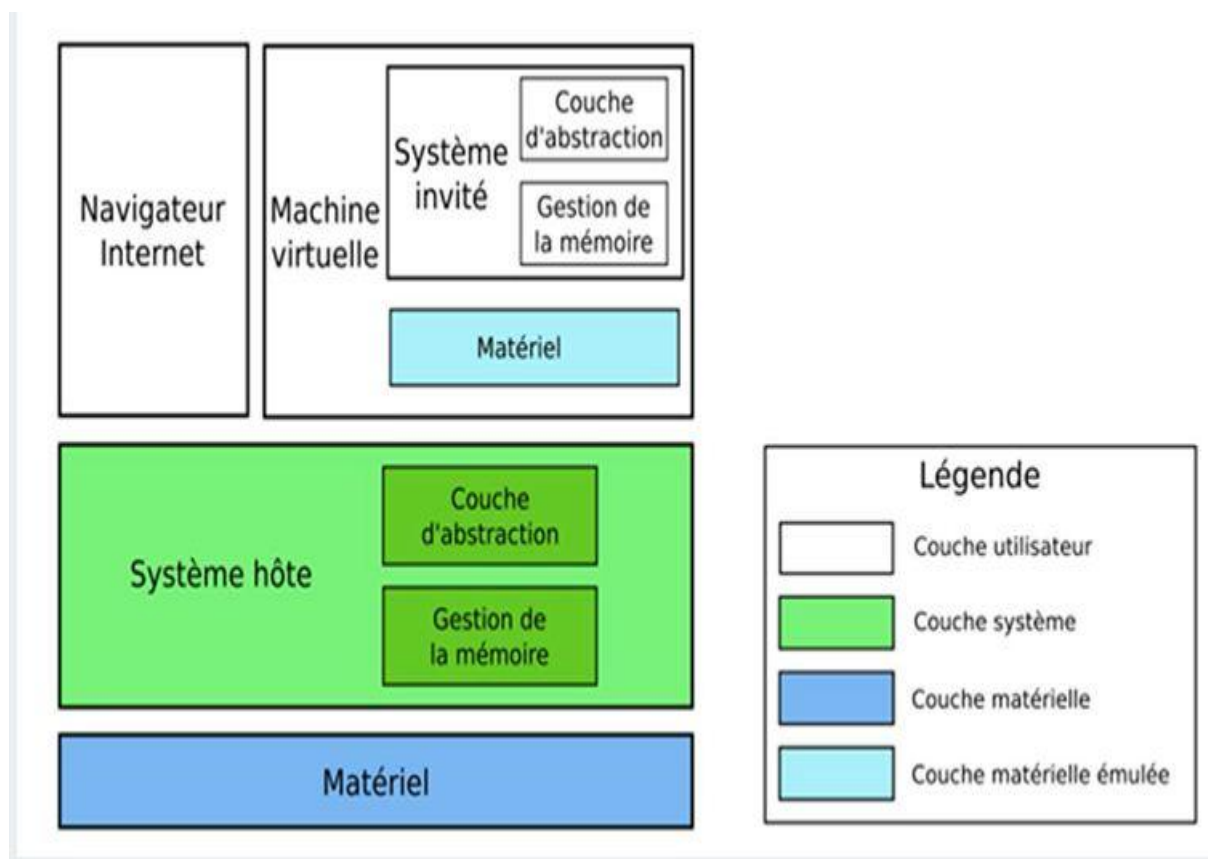


Figure 2.5 : principe de fonctionnement de la virtualisation complète [18]

Les solutions actuelles les plus connues de la virtualisation complète sont : **VMware** et **Oracle VirtualBox**. Elles permettent d'émuler le comportement d'un ordinateur réel auprès du système invité. VMware est cependant, le leader dans le domaine de la virtualisation complète grâce à la performance de ces produits. Elle est de nature commerciale mais dispose des versions gratuites offrant moins de performance. La solution VMware est de plus en plus confrontée à l'évolution d'**Oracle**

L'étoile (*) indique que le mot est expliqué au glossaire

[13] [15]

II.2.1.3 La para-virtualisation

La para-virtualisation est très proche du concept de la virtualisation complète dans le sens où c'est toujours un système d'exploitation complet qui s'exécute sur le matériel émulé par une machine virtuelle ; cette dernière s'exécutant au-dessus d'un système hôte. Toutefois, dans une solution de para-virtualisation, le système invité est modifié pour être exécuté par la machine virtuelle (figure 2.6). Les modifications effectuées visent à rendre le système invité au courant du fait qu'il s'exécute dans une machine virtuelle. De ce fait, il pourra collaborer plus étroitement avec le système hôte, en utilisant une interface spécifique (interface para-virtualisée).

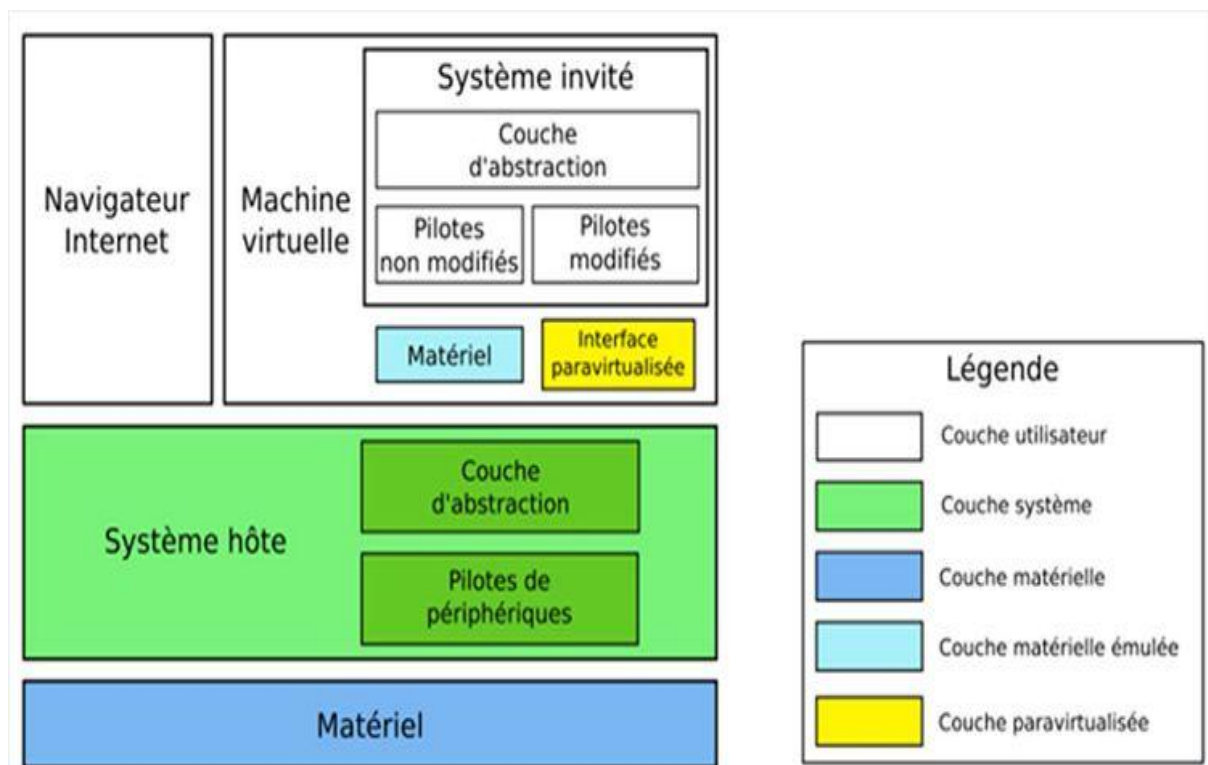


Figure 2.6 : principe de la para-virtualisation (voir à la référence [18])

La para-virtualisation permet aux VMM d'être plus simples et aux machines virtuelles fonctionnant dessus d'atteindre un niveau de performance proche du matériel réel. La solution de Microsoft pour la virtualisation dénommée Hyper-V est considérée comme un système de para-virtualisation.

II.2.1.4 Système à hyperviseur

Un **hyperviseur** est une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une même machine physique en même temps. Ici nous parlons des hyperviseurs de type 1 ou natifs qui est un logiciel qui s'exécute directement sur une plateforme matérielle ; cette plateforme est alors considérée comme **outil de contrôle** de système d'exploitation. Un système d'exploitation secondaire peut, de ce fait, être exécuté au-dessus du matériel. Voir à la référence [15].

L'utilisation d'un hyperviseur est en quelque sorte une évolution logique de la para-virtualisation. Dans les technologies précédentes, le système hôte était le seul à avoir un accès direct au matériel physique. Avec un hyperviseur, le système hôte partage cet accès avec les systèmes invités.

Au démarrage de l'ordinateur, c'est normalement le système d'exploitation qui prend la main et le contrôle du matériel. Dans le cas de l'utilisation d'un hyperviseur, c'est un système minimaliste (l'hyperviseur) qui prend le contrôle du matériel. Ensuite,

des systèmes d'exploitation complets pourront être exécutés par-dessus l'hyperviseur (voir figure 2.7).

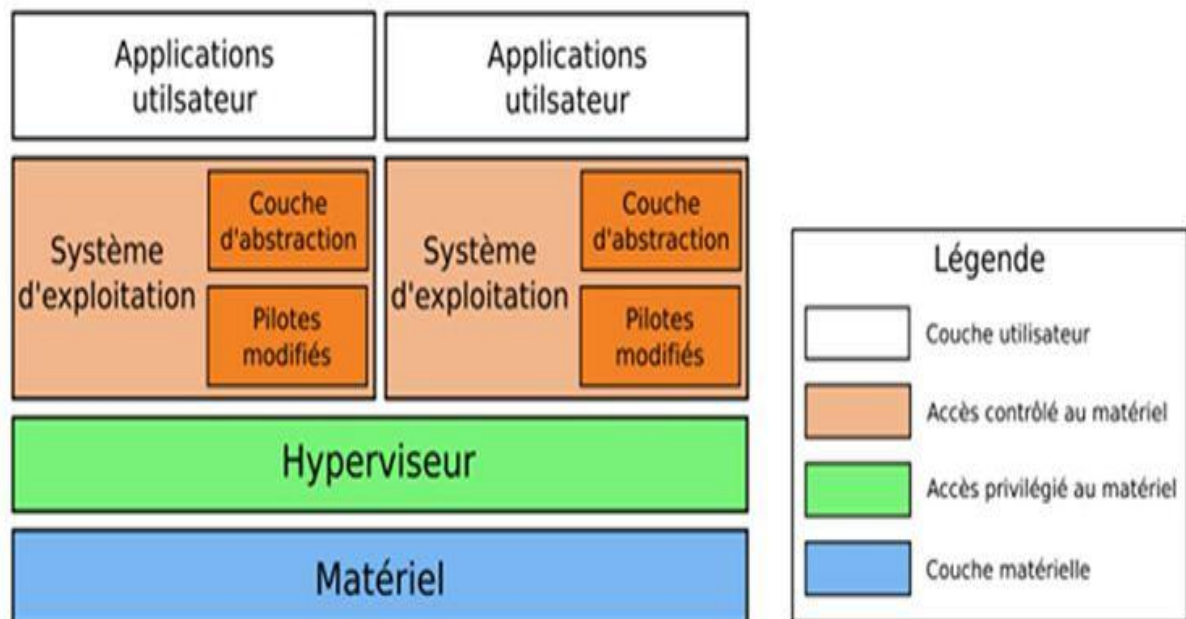


Figure 2.7 : principe des systèmes à hyperviseur (voir la référence [18])

Dans une architecture à hyperviseur, les systèmes d'exploitation exécutés ont presque un accès direct au matériel physique. Le rôle de l'hyperviseur est donc de s'assurer qu'ils n'accèdent qu'aux ressources qui leur sont autorisées sans perturber le fonctionnement des autres systèmes. Xen et hyper-V sont les solutions majeures dans ce domaine. Voir la référence [14]

II.2.1.5 Synthèse

Après avoir vu les principales solutions de la virtualisation, il est important pour nous de faire une synthèse pour mieux voir les avantages et les inconvénients de ces solutions. Le tableau ci-dessous présente un récapitulatif :

Tableau 2.1 : Synthèse des techniques de virtualisation (voir la référence [13])

Types Eléments	Isolation (OpenVZ)	Para- virtualisation	Virtualisation complète
Conscient de la virtualisation	Non	Oui	Non
Performance	++	+++	+

Simplicité	+++	+	++
Exemples de solutions	OpenVZ	Xen, Hyper V, KVM	VirtualBox, VMware

Lorsqu'on regarde ce tableau, on remarque que la virtualisation basée sur la technologie de para-virtualisation est la plus performante, mais est aussi la plus complexe à mettre en place. Les solutions pour faire de la para-virtualisation sont nombreuses on peut citer entre autres Xen, Hyper V, KVM. Le tableau qui suit fait une comparaison des solutions d'hyperviseurs les plus utilisées, cette comparaison est basée sur plusieurs caractéristiques.

Tableau 2.2 : Comparaison des hyperviseurs

Caractéristiques techniques des hyperviseurs	VMware ESX/ESXi						Microsoft Hyper V	Citrix Xen Server	Promox VE3.2
	PME/PMI			Moyennes et grandes entreprises					
Edition	Hyperviseur	Essentials	Essentials Plus	vSphere Standard	vSphere Enterprise	vSphere Enterprise Plus	Hyper V	Xen Server	VE3.2
Tarification	Gratuit	580 €	4321 €	1311 €	3597 €	4375 €	Prix d'une licence Windows serveur (2008 R2 mini)	Offre de base gratuite	Gratuit
Nombres d’hôtes maximum gérés pour une licence	1	3	3	3	3	3	1	Illimité	Illimité
Haute disponibilité	Non	Non	Oui	Oui	Oui	Oui	Oui	Oui	
DRS Répartition de charges	Non	Non	Non	Non	Oui	Oui	Oui Network Load Balancing	/	/
Gestion centralisé	vCenter						System Center	XenCenter	/
Capacité de stockage d'installation minimum	200 Mo						5 Go mini	1 Go mini	/
Interface d'administration	vSphere client / vSphere Web Access						Server manager, RSAT	Xen Center	Interface WEB

Independence vis à vis de l'os	Indépendant du système d'exploitation générique						Repose sur Windows server 2008 et 2012	Repose sur Linux	/
Pilotes	Optimisation avec les fournisseurs de matériels						Pilotes Windows	Pilotes Linux	
Support d'OS	73 Os différents (Windows Linux...)						Garanti Windows	/	/
Ajout à chaud	Non	Non	Non	Non	Oui	Oui	Oui	/	/
Thin Disk Provisioning*	Oui						Oui	Limité	/
SnapShot des VM	Oui			Oui			Oui	Oui mais machine éteintes	Oui
Conversion P2V Physic to Virtual	VMWare Converter			VMWare Converter			Oui (Disk2VHD)	Xen Converter	Oui Ghost disk

Ce tableau nous montre donc les avantages et les inconvénients des différents hyperviseurs les plus utilisés dans le monde de la virtualisation. Il fait une comparaison simple et claire en se basant sous plusieurs critères notamment les prix. On peut donc constater que la solution de **Citrix Xen Server** est un très bon compromis entre faire des économies et avoir une bonne performance (voir à la référence [19]). Mais quels sont les solutions concernant le Cloud lui-même ?

L'étoile (*) indique que le mot est expliqué au glossaire

II.2.2 Les services web

Un Service Web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit d'un ensemble de fonctionnalités exposées sur Internet ou sur un Intranet, pour des applications ou machines, sans intervention humaine, et en **temps réel***.

Les solutions de Cloud Computing se basent sur les services Web à la fois pour l'exposition des fonctionnalités du système sur Internet en respectant les standards tels que le WSDL ou l'UDDI, et aussi pour l'interaction des différents composants internes du système en se basant sur les **protocoles de communication** tels que le XML-RPC, le SOAP, le REST, le JSON ou encore le RFB. Voir la référence [13].

II.2.2.1 XML-RPC

XML-RPC est un protocole RPC (Remote Procedure Call), une spécification simple et un ensemble de codes qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau.

XML-RPC permet d'appeler une fonction sur un serveur distant à partir de n'importe quel système (Windows, Mac OS X, GNU/Linux) et avec n'importe quel langage de programmation. Le serveur est lui-même sur n'importe quel système et est programmé dans n'importe quel langage.

Cela permet de fournir un service web utilisable par tout le monde sans restriction de système ou de langage. Les processus d'invocation à distance utilisent le protocole HTTP pour le transfert des données et la norme XML pour la structuration des données.

XML-RPC est conçu pour permettre à des structures de données complexes d'être transmises, exécutées et renvoyées très facilement. XML-RPC est une alternative aux Services Web WS-*, dont SOAP.

II.2.2.2 SOAP

SOAP (ancien acronyme de *Simple Object Access Protocol*) est un protocole de RPC orienté objet bâti sur XML.

Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur.

L'étoile (*) indique que le mot est expliqué au glossaire

Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP.

Le protocole SOAP est composé de deux parties :

- Une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement,
- Un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

SOAP est utilisée notamment dans le cadre d'architectures de type **SOA** (*Service Oriented Architecture*) pour les Services Web WS-^{*}.

Avantages

- Utiliser SOAP via HTTP facilite la communication et évite les problèmes de proxys et pare-feu par rapport à des technologies plus anciennes
- SOAP est
 - Assez ouvert pour s'adapter à différents protocoles de transport.
 - Indépendant de la plate-forme.
 - Indépendant du langage.
 - Extensible.

Inconvénients

- En raison du nombre d'informations qu'impose le format XML, SOAP peut alourdir considérablement les échanges par rapport à des middlewares* comme CORBA, ce qui n'est pas forcément un handicap quand les volumes de données transités par SOAP sont faibles par rapport au volume total de données échangées.
- SOAP décrit la manière dont les applications doivent communiquer entre elles, certains considèrent que le couplage reste fort entre le serveur et ses clients. Une modification de l'API implique ainsi une évolution côté client, contrairement à une architecture orientée ressources telle que REST.

L'étoile (*) indique que le mot est expliqué au glossaire

II.2.2.3 REST

REST (Representational State Transfer) est un style d'architecture pour les systèmes hypermédia distribués, créé par Roy Fielding en 2000.

REST repose sur le même principe de fonctionnement que les Web Services. Elle s'appuie sur les protocoles Internet, dont HTTP, pour véhiculer des messages décrits au format XML. Cependant, la principale différence qui réside entre REST et les autres protocoles est qu'il ne s'agit pas d'un standard mais d'un style architectural qui se veut simple pour l'échange de données entre applications (Costello, 2007).

Avantages

La thèse de Roy Fielding précise les avantages de ce style architectural par rapport à d'autres styles d'architectures d'applications Web. Entre autres :

- L'application est plus simple à entretenir, car elle désolidarise la partie client de la partie serveur. La nature hypermédia de l'application permet d'accéder aux états suivants de l'application par inspection de la ressource courante.
- L'absence de gestion d'état du client sur le serveur conduit à une plus grande indépendance entre le client et le serveur. Elle permet également de ne pas avoir à maintenir une connexion permanente entre le client et le serveur. Le serveur peut ainsi répondre à d'autres requêtes venant d'autres clients sans saturer l'ensemble de ses ports de communication. Cela devient essentiel dans un système distribué.
- L'absence de gestion d'état du client sur le serveur permet également une répartition des requêtes sur plusieurs serveurs : une session client n'est pas à maintenir sur un serveur en particulier (via une *sticky* session* d'un *loadbalancer*), ou à propager sur tous les serveurs (avec des problématiques de fraîcheur de session). Cela permet aussi une meilleure évolutivité et tolérance aux pannes (un serveur peut être ajouté facilement pour augmenter la capacité de traitement, ou pour en remplacer un autre).
- Dans un contexte Web :
 - L'utilisation du protocole HTTP permet de tirer parti de son enveloppe et ses en-têtes ;

- L'utilisation d'URL comme représentant d'une ressource permet d'avoir un système universel d'identification des éléments de l'application,
- La nature auto-descriptive du message permet la mise en place de serveurs cache : les informations nécessaires sur la fraîcheur, la péremption de la ressource sont contenues dans le message lui-même.

Inconvénients

Le principal inconvénient de REST est la nécessité pour le client de conserver localement toutes les données nécessaires au bon déroulement d'une requête, ce qui induit une consommation en bande passante réseau plus grande. Notamment dans l'univers des appareils mobiles, chaque aller-retour de connexion est consommateur de bande passante. La latence de la connexion rend également l'interaction moins fluide.

II.2.2.4 JSON

JSON ou JavaScript Object Notation, est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Il a été Créé par Douglas Crockford entre 2002 et 2005.

Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.

Un document JSON ne comprend que deux types d'éléments structurels :

- Des ensembles de paires nom/valeur ;
- Des listes ordonnées de valeurs.

Ces mêmes éléments représentent trois types de données :

- Des objets ;
- Des tableaux ;
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.

Ces types de données sont suffisamment génériques et abstraits pour, d'une part, pouvoir être représentés dans n'importe quel langage de programmation, d'autre part, pouvoir représenter n'importe quelle donnée concrète.

Le principal avantage de JSON est qu'il est simple à mettre en œuvre par un développeur tout en étant complet.

Au rang des avantages, on peut également citer :

- Peu verbeux, ce qui le rend lisible aussi bien par un humain que par une machine ;
- Facile à apprendre, car sa syntaxe est réduite et non extensible (bien qu'il souffre de quelques limitations) ;
- Ses types de données sont connus et simples à décrire.
- Le JSON est plus facile à parser* que le XML.

II.2.2.5 Le RFB

Remote Frame Buffer (RFB) est un protocole simple pour l'accès à distance aux interfaces utilisateur graphiques. Car il fonctionne au niveau du **frame buffer***, il est applicable à tous les systèmes de fenêtrage et applications, y compris Microsoft Windows, Mac OS X et le système X Windows. RFB est le protocole utilisé dans le Virtual Network Computing (VNC) et ses dérivés.

Ce protocole à plusieurs avantages on peut citer par exemple la souplesse dans l'utilisation, facilité de mise en place via VNC et la prise en charge de multiples encodages.

Il a aussi des inconvénients donc le plus important reste certainement le fait que le transfert de données du presse-papier ne prenne en compte que le jeu de caractères Latin-1.

II.2.3 Le web 2.0

L'expression fondamentale « **Web 2.0** » désigne l'ensemble des techniques, des fonctionnalités et des usages qui ont suivi la forme originelle du web, *www* ou *World Wide Web*, caractérisée par plus de simplicité et d'interactivité. Elle concerne en particulier les interfaces et les échanges permettant aux internautes ayant peu de connaissances techniques de s'approprier de nouvelles fonctionnalités du web. Les internautes peuvent d'une part contribuer à l'échange d'informations et interagir (partager, échanger, etc.) de façon simple, à la fois au niveau du contenu et de la structure des pages, et d'autre part entre eux, créant notamment le Web Social. L'internaute devient, grâce aux outils mis à sa disposition, une personne active sur la toile.

L'étoile (*) indique que le mot est expliqué au glossaire

II.3 Les solutions du Cloud Computing

Dans le monde du Cloud Computing plusieurs plateformes s'offrent à nous pour le mettre en place et parmi elles on trouve des solutions propriétaires et des solutions open source.

II.3.1 Les solutions propriétaires

II.3.1.1 La plateforme Azure de Microsoft



Illustration 2.1 : logo de Windows Azure (voir la référence [20])

Appelé Windows Azure jusqu'en 2014, Microsoft Azure est la plateforme applicative en nuage de Microsoft. Il s'agit d'une offre d'hébergement (applications et données) et de services (workflow*, stockage et synchronisation des données, bus de messages, contacts...). Un ensemble d'API permet d'utiliser et d'accéder à cette plateforme et aux services associés. Un environnement d'exécution (le « Live Operating Environment ») permet une intégration étroite avec les principaux systèmes d'exploitation existant (Windows, Mac OS et Windows Phone). La plateforme Microsoft Azure correspond aux offres d'informatique en nuage publics et privé de type PAAS et IAAS de Microsoft.

II.3.1.2 Google App Engine



Illustration 2.2 : logo Google App Engine (voir la référence [21])

Google App Engine est une plateforme de conception et d'hébergement d'applications web basée sur les serveurs de Google. Cette plateforme est automatiquement basée sur le trafic entrant. L'équilibrage de charge, les micro-services, les bases de données SQL et NoSQL, memcached*, le fractionnement du trafic,

L'étoile (*) indique que le mot est expliqué au glossaire

l'exploration forestière, la recherche, le versioning*, et le balayage de sécurité sont pris en charge en mode natif et sont hautement personnalisable.

II.3.1.3 VMware vCloud Director



Illustration 2.3 : logo du vCloud de VMware (voir la référence [22])

vCloud est un projet de Cloud Computing mené par VMware. Il a pour but de permettre à ses clients de migrer leur travail, à leur demande, à partir de leur stockage interne des hyperviseurs VMware vers un stockage à distance (des hyperviseurs VMware également). Le but du projet est de fournir la puissance du Cloud Computing avec la flexibilité permise par la virtualisation.

II.3.1.4 La plateforme EC2 d'Amazon



Illustration 2.4 : logo EC2 d'Amazon (voir la référence [23])

Amazon Elastic Compute Cloud (Amazon EC2) est un service Web qui fournit une capacité de calcul redimensionnable dans le cloud. Destiné aux développeurs, il est conçu pour faciliter l'accès aux ressources de Cloud Computing à l'échelle du Web.

II.3.2 Les solutions open source

II.3.2.1 Eucalyptus



Illustration 2.5 : logo d'Eucalyptus (voir la référence [24])

Eucalyptus est un logiciel médiateur (*middleware*) open-source pour l'implémentation du Cloud Computing (ou informatique en nuage) sur une grappe de serveurs. Son nom fait référence à l'acronyme anglais « Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems » qui peut se traduire en

L'étoile (*) indique que le mot est expliqué au glossaire

« Utilitaire d'Architecture informatique élastique pour relier vos programmes à des systèmes fonctionnels ». Eucalyptus est compatible avec Amazon Web Services. Il est intégré dans la distribution Gnu/Linux Ubuntu 9.04 en tant qu'outils de « Cloud Computing ». Eucalyptus peut s'installer facilement sur la majorité des distributions GNU/Linux : Debian, CentOS...

II.3.2.2 Open Stack



Illustration 2.6 : logo de Open Stack (voir à la référence [25])

OpenStack est un ensemble de logiciels open source permettant de déployer des infrastructures de Cloud Computing (infrastructure en tant que service). La technologie possède une architecture modulaire composée de plusieurs projets corrélés (Nova, Swift, Glance...) qui permettent de contrôler les différentes ressources des machines virtuelles telles que la puissance de calcul, le stockage ou encore le réseau inhérents au centre de données sollicité.

II.3.2.3 OpenNebula



Illustration 2.7 : logo de OpenNebula (voir à la référence [26])

OpenNebula est une plateforme de Cloud Computing pour la gestion des Datacenter hétérogènes et distribués. Cette plateforme gère l'infrastructure virtuelle du Datacenter de construction privée, publique et hybride implémentant le service IAAS. OpenNebula est un logiciel gratuit et open-source.

II.3.2.4 Xen Cloud Platform (XCP)



Illustration 2.8 : logo de XCP (voir à la référence [27])

La Xen Cloud Platform ou XCP est une porte clé pour les services de virtualisation et de Cloud Computing open source offrant la possibilité de déployer des solutions de cloud privé et hybride. Cette plateforme a été créée par l'entreprise **Xen.org** aujourd'hui **Citrix.org** et intègre plusieurs outils notamment :

- **L'hyperviseur Xen** : qui fournit le faible niveau de la virtualisation.
- **XAPI** : le toolstack XCP qui est utilisé en interne pour gérer l'hyperviseur.
- **xe** : l'outil de ligne de commande pour configurer et exploiter la plateforme XCP.
- **xsconsole** : une interface utilisateur à base de texte pour effectuer des tâches et des configurations de base.
- **XenCenter** : une interface web pour la supervision
- **XenDesktop** : une plateforme web pour la connexion à distance aux VM (serveur de bureaux distants)
- **Xen Receiver** : client pour l'accès à distance aux machines virtuelles.

XCP intègre également un serveur de VM à savoir XenServer qui est la distribution commerciale de l'hyperviseur Xen créée par l'entreprise Citrix* pour XCP. XenServer permet ainsi d'avoir une interface graphique (application Web) pour la gestion des VMs qu'on appelle XenCenter.

Une des fonctionnalités clef de Xen Cloud Platform (XCP) est l'interopérabilité entre des clouds d'entreprise et les plateformes comme Amazon EC2, OpenNebula, OpenStack, Apache CloudStack. (Voir les références numéro [28])

II.3.2.5 Conséquences du Cloud Computing

Pour les fournisseurs, le développement du Cloud Computing entraîne le développement des centres de données ou Datacenter*. Les fournisseurs de service doivent augmenter leurs infrastructures (serveurs, bande passante, surface au sol...) pour faire face aux besoins croissants des clients. Les modèles proposés par ces prestataires doivent également évoluer, le cloud privé transformant les modèles économiques qui prévalaient jusqu'ici.

À titre d'exemple, **Salesforce.com**, pionnier dans le domaine du nuage informatique, gère les données de 54000 entreprises, et leurs 1,5 million d'employés, avec seulement 1000 serveurs (mars 2009). Un déplacement des effectifs informatiques vers les opérateurs de clouds est une conséquence logique de cette tendance.

Pour les utilisateurs, particuliers et entreprises, la location de services associée au Cloud Computing permet généralement de réaliser des économies à court terme. Mais le coût total à moyen et long terme peut se révéler, au bout de quelques années, supérieur au coût d'une application hébergée en interne. Cela dépend du mode d'utilisation (fréquence, nombre d'utilisateurs...) et de la durée de vie de l'application. Un calcul comparatif s'impose avant de faire son choix. Ce calcul ne doit pas se limiter aux coûts directs, mais doit aussi intégrer l'ensemble des coûts cachés que le cloud va permettre d'économiser ainsi que l'impact des avantages du cloud sur le business de l'entreprise (productivité accrue, recentrage métier...). La tâche n'est pas forcément facilitée par les modes de facturation proposés qui sont parfois peu « lisibles », et dépendent de plusieurs paramètres : l'utilisation des fonctions (volumétrie), le coût de production ou de mise à disposition, incluant les évolutions, le degré de complexité, et enfin le tarif locatif du service. Les durées d'engagement peuvent varier d'un prestataire à l'autre mais restent pour la plupart autour de deux ou trois ans. (Voir la référence [10])

II.3.2.6 Avantages du Cloud Computing

Le *Cloud Computing* peut permettre d'effectuer des économies, notamment grâce à la mutualisation des services sur un grand nombre de clients. Certains analystes indiquent que 20 à 25% d'économies pourraient être réalisées par les gouvernements sur leur budget informatique s'ils migraient vers le *Cloud Computing*. Comme pour la virtualisation, l'informatique dans le nuage peut être aussi intéressante pour le client grâce à son évolutivité. En effet, le coût est fonction de la durée de l'utilisation du service

L'étoile (*) indique que le mot est expliqué au glossaire

rendu et ne nécessite aucun investissement préalable (homme ou machine). L'élasticité du nuage permet de fournir des services évolutifs et peut permettre de supporter des montées en charge. Inversement, le fournisseur a la maîtrise sur les investissements, est maître des tarifs et du catalogue des offres, et peut se rémunérer d'autant plus facilement que les clients sont captifs.

La maintenance, la sécurisation et les évolutions des services étant à la charge exclusive du prestataire, donc c'est généralement le cœur de métier, celles-ci ont tendance à être mieux réalisées et plus rapidement que lorsque sous la responsabilité du client (principalement lorsque celui-ci n'est pas une organisation à vocation informatique). (Voir la référence [10])

II.3.2.7 Inconvénients du Cloud Computing

Plusieurs catégories d'inconvénients existent :

- L'utilisation des réseaux publics, dans le cas du cloud public, entraîne des risques liés à la sécurité du cloud. En effet, la connexion entre les postes et les serveurs applicatifs passe par le réseau internet, et sont exposés à des risques supplémentaires de cyberattaques, et de violation de confidentialité. Le risque existe pour les particuliers, mais aussi pour les grandes et moyennes entreprises, qui ont depuis longtemps protégé leurs serveurs et leurs applications des attaques venues de l'extérieur grâce à des réseaux internes cloisonnés.
- Le client d'un service de Cloud Computing devient très dépendant de la qualité du réseau pour accéder à ce service (Bande passante). Aucun fournisseur de service cloud ne peut garantir une disponibilité de 100%. Par exemple, des défaillances sur les services cloud sont référencées par l'International Working Group of Cloud Resiliency.
- Les entreprises perdent la maîtrise de l'implantation de leurs données. De ce fait, les interfaces inter-applicatives (qui peuvent être volumineuses) deviennent beaucoup plus complexes à mettre en œuvre que sur une architecture hébergée en interne.
- Les entreprises n'ont plus de garanties (autres que contractuelles) de l'utilisation qui est faite de leurs données, puisqu'elles les confient à des tiers.

- Des questions juridiques peuvent se poser, notamment par l'absence de localisation précise des données du Cloud Computing. Les lois en vigueur s'appliquent, mais pour quel serveur, quel data center, et surtout quel pays ?
- Tout comme les logiciels installés localement, les services de Cloud Computing sont utilisables pour lancer des attaques (craquage de mots de passe, déni de service...). En 2009, par exemple, un cheval de Troie a utilisé illégalement un service du Cloud public d'Amazon pour infecter des ordinateurs.
- Du fait que l'on ne peut pas toujours exporter les données d'un service cloud, la réversibilité (ou les coûts de sortie associés) n'est pas toujours prise en compte dans le cadre du projet. Le client se trouve souvent « piégé » par son prestataire et c'est seulement lorsqu'il y a des problèmes (changement des termes du contrat ou des conditions générales d'utilisation, augmentation du prix du service, besoin d'accéder à ses données en local, etc.) qu'il se rend compte de l'enfermement propriétaire (vendor lock-in) dans lequel il se trouve.

L'ONG Greenpeace dénonce aussi, dans son rapport 2010 sur l'impact écologique du secteur informatique, les impacts négatifs du nuage informatique. Un de ceux-ci réside en la consommation élevée d'énergie des fermes de serveurs qui supportent les services du nuage informatique. À titre d'illustration, si l'ensemble des fermes de serveur du nuage informatique constituaient un pays, celui-ci viendrait au quatrième rang mondial des plus grands consommateurs d'énergie. (Voir les références [9] [10])

Conclusion :

Au terme de ce chapitre nous comprenons que le Cloud Computing est en fait de la dématérialisation de matériel. Qu'il est un concept nouveau et est en plein essors. On a pu voir que le Cloud Computing intègre trois principaux services à savoir l'IAAS, le PAAS et le SAAS ainsi que quatre modèles de déploiement notamment le cloud privé, public, hybride et communautaire. Nous avons également vu que ses principales technologies sont la virtualisation avec les hyperviseurs, la para-virtualisation, ... et les services web avec XML-RPC, SOAP, ... Plusieurs solutions existent pour faire du Cloud Computing et parmi ces solutions on a les solutions propriétaires et open source. Comme toutes choses le cloud présente des avantages et des inconvénients. A présent

que nous connaissons ce qu'est le Cloud Computing, nous devons le rendre tolérant aux pannes en distribuant les tâches et pouvoir le superviser.

CHAPITRE III : ALGORITHME DE SUPERVISION DISTRIBUE

Introduction :

Dans ce chapitre nous allons présenter notre solution pour superviser le réseau et notre algorithme de distribution des tâches. Mais avant d'y arriver il est important de comprendre certaines notions notamment les notions de système distribué, algorithme distribué, réplication des données.

III.1 Notion de système distribué

Andrew Tanenbaum définit le système distribué comme « un ensemble d'ordinateurs indépendants qui apparaissent à l'utilisateur comme un seul système homogène ». Cette définition est très simple et cache toute la complexité qui se trouve derrière cette notion. Ainsi si nous devons entrer en profondeur et utiliser des termes plus techniques, on peut dire que :

Un système distribué est un ensemble composé d'éléments reliés par un système de communication (réseau de communication); les éléments ont des fonctions de traitement (processeurs), de stockage (mémoire). Les différents éléments du système ne fonctionnent pas indépendamment mais collaborent à une ou plusieurs tâches communes. La conséquence est qu'une partie au moins de l'état global du système est partagée entre plusieurs éléments. L'histoire des systèmes distribués est étroitement liée à l'évolution des outils de communication. Ainsi, l'apparition vers le milieu des années 70 du réseau Ethernet, réseau local à haut débit (sic) utilisant un réseau à diffusion, marque une étape importante : premiers postes de travail individuels dotés de facilités graphiques, organisation client/serveur pour l'accès aux ressources partagées. Les premiers systèmes répartis utilisant les réseaux locaux, qui apparaissent à la fin des années 70, sont réalisés en interconnectant plusieurs systèmes homogènes (notamment des systèmes Unix). La plupart de ces systèmes étendent simplement le système de fichiers pour offrir un accès transparent aux fichiers locaux ou distants ; quelques-uns permettent la création et l'exécution de processus à distance. Ce n'est qu'au début des années 80 qu'apparaissent les systèmes distribués intégrés conçus au départ comme répartis. Ces systèmes donnent alors l'impression aux usagers qu'ils disposent d'un

système d'exploitation aussi souple que les systèmes à temps partagé mais avec de nouvelles possibilités.

L'essor des systèmes distribués a entraîné le développement de nouvelles applications conçues pour s'y exécuter : les **applications distribuées***. Pour le programmeur d'applications distribuées, il se pose alors un problème nouveau : celui de la communication. La gestion des communications est une charge pour le programmeur du fait des imperfections de la transmission des messages (perte par exemple) et des défaillances possibles des sites (nœuds, éléments) du réseau de communication. Le développement de protocoles de communication fiable permet alors de décharger le programmeur de la lourde tâche du traitement des erreurs de transmission et des défaillances susceptibles de survenir pendant une communication.

Dans un tel système, tous les nœuds ou sites exécutent au moins un processus du système distribué. Un des principes de l'informatique distribuée est de briser la symétrie de ces processus en octroyant à l'un d'entre eux le privilège de supervision des autres. C'est le rôle des algorithmes distribués. (Voir les références [29] [30] [31])

III.2 Algorithmes distribués

Dans l'algorithmique classique, nous avons un **processus*** qui exécute un **algorithme*** sur une seule donnée et ayant une complexité égale aux : **nombres d'opération / nombres instructions** comme le montre la figure. Voir la référence numéro [32]



Figure 3.1: Algorithmique classique (voir référence [32])

En ce qui concerne les algorithmes distribués, il existe trois types de modèles :

III.2.1 Mémoire partagé

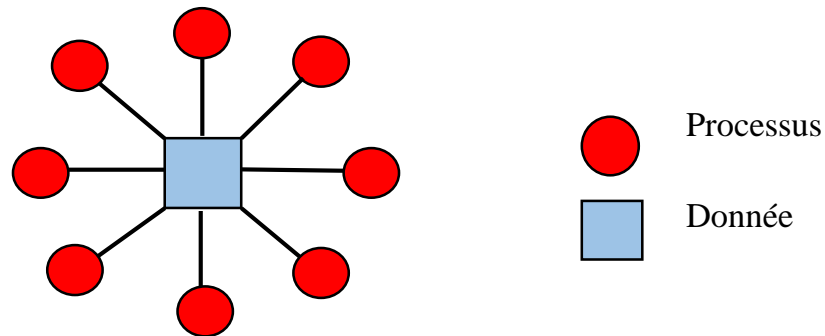


Figure 3.2: Algorithme à mémoire partagé (voir référence [32])

Comme le montre cette figure 3.2, nous pouvons voir que dans ce modèle nous avons :

- Une donnée
- n processus qui exécutent le même algorithme et peuvent lire/écrire dans la donnée
- complexité = nombre d'écriture/lecture

III.2.2 Agents mobiles

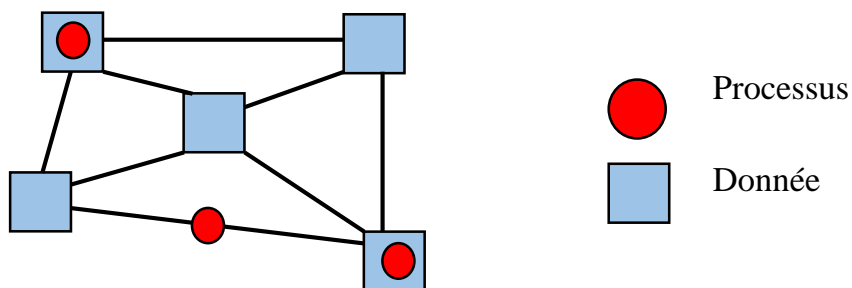


Figure 3.3 : Algorithme à agents mobiles (voir référence [32])

Ici nous avons :

- n données
- k processus qui exécutent le même algorithme et peuvent se déplacer dans le réseau.
- complexité = nombre de déplacements

III.2.3 Passage de message

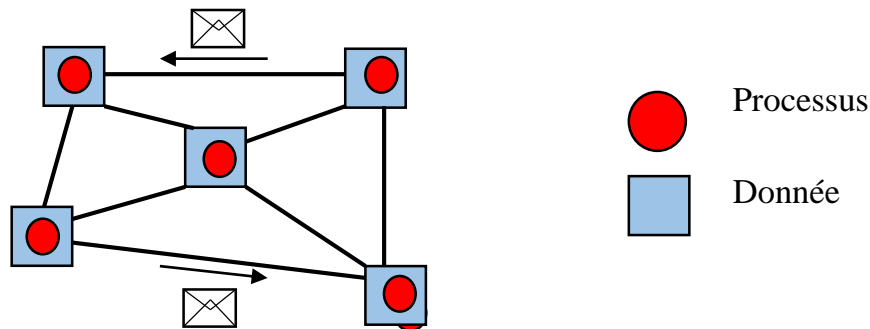


Figure 3.4 : algorithme distribué par passage de messages (voir référence [32])

La figure 3.4 nous explique qu'on a :

- n données
- n processus qui exécutent le même algorithme et peuvent envoyer des messages à leur voisins
- complexité = nombre de messages

Dans ce type de modèle, tous les processus exécutent le “même” algorithme (l'état initial peut différer.) ; Chaque processus connaît l'information locale (pas d'information globale). Ils doivent communiquer pour l'échange des informations et à chaque étape de l'algorithme, un nœud peut : envoyer des messages aux voisins, recevoir des messages des voisins, et faire des calculs locaux.

III.3 Principe d'élection dans un système distribué

Etant donné un réseau de nœuds initialement dans l'état Inconnu, l'algorithme termine avec les conditions suivantes :

- **Election** : Un nœud et un seul termine dans l'état ELU.
- **Variante** (terminaison explicite) Tous les autres nœuds savent que l'algorithme est terminé (et qu'ils sont non-ELU).

III.4 Caractérisation des systèmes distribués

Un système distribué est caractérisé par :

- Les processus qui sont des programmes en cours d'exécution doté d'une structure de contrôle non déterministe car ils attendent plusieurs événements possibles.
- Les voies de communication :

- Anneau : structure en boucle, ici tout processus possède deux voisins et il ne peut communiquer qu'avec ces deux voisins.
- Etoile : structure centralisée, ici un processus en particulier peut communiquer avec tous les autres qui, réciproquement ne communiquent qu'avec le processus particulier. Ce maillage présente l'inconvénient de particulariser un site et de rendre ainsi fâcheuse une panne de ce dernier.
- Arbre : structure hiérarchisée, ici la panne de tout processus non terminal déconnecte le système.
- Maillage complet : structure fortement connexe, ici tout processus peut communiquer avec les autres.

Les autres caractéristiques d'un système distribué sont entre autres la distribution des données et la distribution du contrôle :

- La distribution des données : afin d'assurer la reprise des services de supervision en cas de panne d'un nœud ou site, une donnée x du système doit être recopiée sur les autres sites. C'est le principe de duplication des données. L'autre principe en termes de distribution des données est le partitionnement, qui consiste à recopier sur chaque site une partie de l'information. [32]
- La distribution du contrôle : dans un système de supervision distribué, le principe de privilège tournant permet de pallier aux problèmes de panne générale. Un site particulier possède le privilège de superviser les autres. En cas de panne de ce dernier, un autre site est immédiatement élu pour continuer les tâches de supervision des autres. Le privilège passe ainsi de site en site pour assurer la continuité des services de supervision dans le réseau. [32]

III.5 La réplication des données

C'est une technique qui consiste à copier les données de références sur plusieurs sites : un site primaire (maître) et un ou plusieurs sites secondaires (esclave) :

Sur le maître, l'application se fait en lecture/écriture et plusieurs opérations peuvent s'y effectuer notamment les insertions, les updates, les sélections, les suppressions. Par contre sur l'esclave, l'application se fait en lecture uniquement la seule

opération qu'on peut effectuer est la sélection ; les mises à jour sont gérées par un protocole.

La réplication des données comporte de nombreux avantages on peut citer entre autres l'équilibrage des charges entre les sites, un meilleur temps de réponse, une réduction du temps de transfert des données, et surtout une forte tolérance aux pannes. Mais elle comporte aussi des inconvénients : les mises à jour sont plus coûteuses car il y a un échange élevé de messages inter-sites, on doit faire un contrôle de cohérence entre la donnée de référence et la donnée répliquée. Il existe deux types de répliquations des données : la réplication synchrone et la réplication asynchrone. [33]

III.5.1 La réplication des données synchrone

Une réplication synchrone fonctionne de la façon suivante : une transaction n'est pas considérée validée (pas de COMMIT) tant que tous les serveurs n'ont pas validé cette transaction. Il y a plusieurs intérêts à ça :

- Il n'y a pas de risque de perte de données au moment d'une bascule du maître ;
- Le résultat d'une requête est cohérent quel que soit le serveur interrogé dans le cadre d'un cluster* en répartition de charge.

Le souci majeur de ce type de réplication est les performances (ou plutôt les contre-performances). En effet, il faut bien comprendre que pour chaque transaction modifiant des données, il ne suffit pas qu'un serveur enregistre cette information. Il faut que tous l'aient enregistré. Cela va ajouter une surcharge plus ou moins importante suivant la solution choisie. Des solutions asynchrones existent. [34] [35]

III.5.2 La réplication des données asynchrone

Les solutions asynchrones sont généralement plus performantes. Le serveur qui fait l'écriture renvoie au client le fait que l'enregistrement s'est bien passé et se charge un peu après d'envoyer l'information sur les données modifiées aux autres serveurs. En cas d'arrêt brutal (crash) du serveur maître, les données peuvent ne pas avoir été complètement synchronisées avec les serveurs esclaves. Cela occasionne une perte de données, généralement faible mais toujours très désagréable.

Problème peut être pire, la répartition de charge. Dans ce cadre, il est tout à fait possible que la même requête exécutée sur deux serveurs différents ne renvoie pas les mêmes résultats suivant le serveur sur laquelle elle est exécutée. Tout dépend évidemment du délai dans la synchronisation avec les autres serveurs. [35]

L'étoile (*) indique que le mot est expliqué au glossaire

III.5.3 Les différents types de répicateurs

Il existe plusieurs technologies de répliquions qui font aussi bien de la répliquion synchrone, qu'asynchrone. On peut citer entre autre :

- **DRBD : Distributed Replicated Block Device** ou périphérique en mode bloc répliqué et distribué est une architecture de stockage distribuée pour GNU/Linux, permettant la répliquion de périphériques de bloc (disques, partitions, volumes logiques, etc...) entre des serveurs. Très utilisé dans la répliquion synchrone. Voir annexe 4 pour les configurations.
- **SRDF**, abréviation de **Symmetrix Remote Data Facility**, est un produit de la société **EMC Corporation** permettant la répliquion de données informatiques entre 2 baies de disques en utilisant le mécanisme miroir Maître/Esclave. Les disques maîtres (nommé R1) sont en accès RW (Read-Write) et les disques esclaves (nommé R2) sont en WD (Write Disable).
- **Global Mirror** est une technologie IBM qui fournit de la répliquion des données sur de longues distances entre les deux sites pour permettre la reprise des activités après un sinistre. Global Mirror permet de faire de la répliquion asynchrone avec une durée entre 3 à 5 secondes.
- **IDR : Informatica Data Replication** est l'un des leaders dans le monde de la répliquion des données. Il fonctionne aussi bien en mode synchrone qu'en mode asynchrone.

Et cette liste est non exhaustive car en effet il existe encore plusieurs algorithmes de répliquion synchrone et/ou asynchrone.

III.6 Présentation de notre solution

Toutes les solutions de cloud que nous avons étudiées en amont sont pour la plupart constituées d'un nœud central de supervision appelé **Cloud Controller (CLC)**. Le problème lié à ces solutions est l'arrêt des services livrés aux utilisateurs lors d'une panne au niveau du cloud Controller.

Notre solution est donc basée sur un système qui va permettre une reprise facile du service en cas de panne du CLC. En effet les clients qui veulent accéder à leurs machines virtuelles doivent pouvoir le faire de n'importe où, et même quand le serveur principal tombe en panne.

Le scénario est simple lorsqu'une VM est créée, elle est faite sur le serveur primaire. Après cette création, une instance de cette VM est créée ou répliquée sur les autres serveurs du pool. Pour assurer une disponibilité permanente et une reprise facile des services, un basculement manuel ou automatique doit être configuré sur le pool de serveurs. En reprenant l'architecture d'Eucalyptus qui adopte une représentation modulaire sous forme d'arbre des différents hôtes du réseau, on obtient à la figure 3.5 le schéma de l'architecture en cas de panne du nœud central du système.

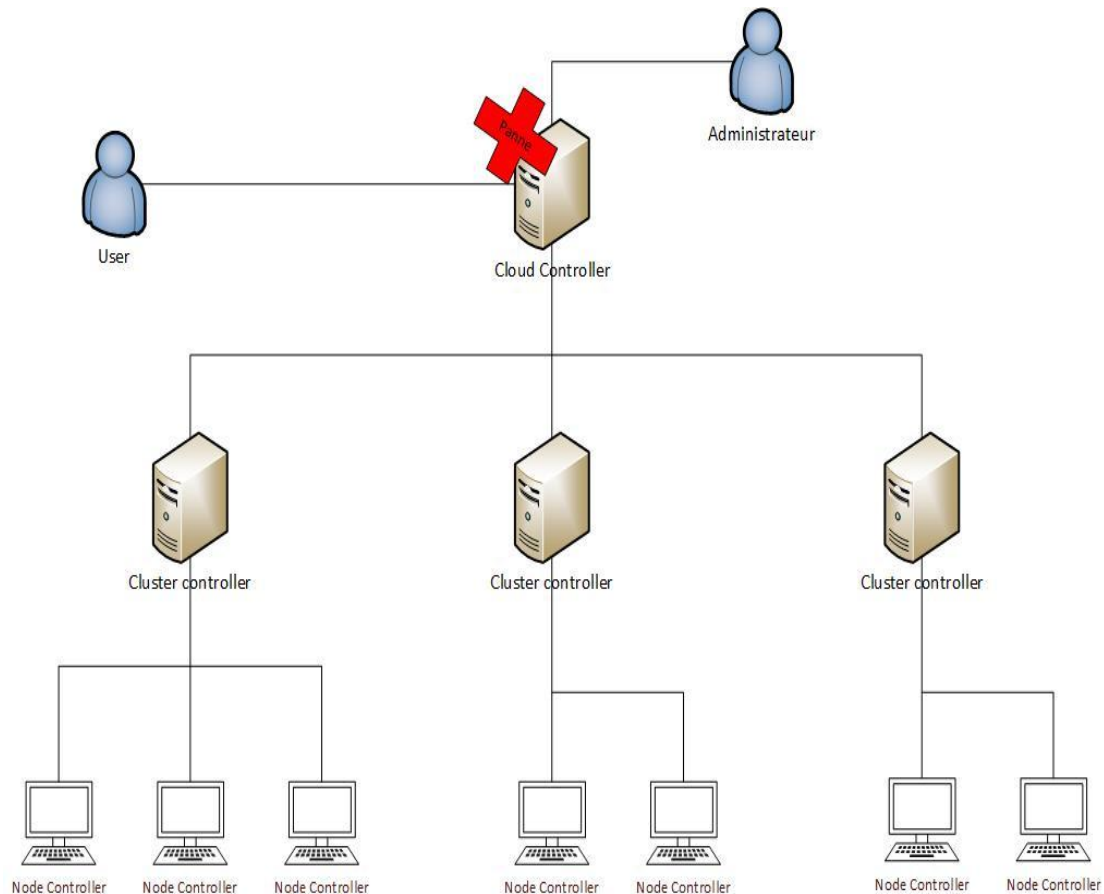


Figure 3.5 : Schéma de l'architecture du cloud privé : Cas de panne du nœud central de supervision.

Comme on peut le voir la panne du nœud central de supervision coupe ainsi les services d'administration et de supervision disponibles au niveau du cloud Controller. Bien que les différents nœuds du réseau soient toujours fonctionnels, il est impossible d'accéder à ces derniers via les différents portails et interfaces d'administration présents au niveau du CLC. Aussi, la tâche de supervision assurée par le cloud Controller est ainsi arrêtée. Le même problème s'observe partiellement en cas de panne d'un cluster Controller qui priverait le cloud Controller de l'accès aux

informations et services disponibles au niveau des différents nœuds de ce cluster (Figure 3.6).

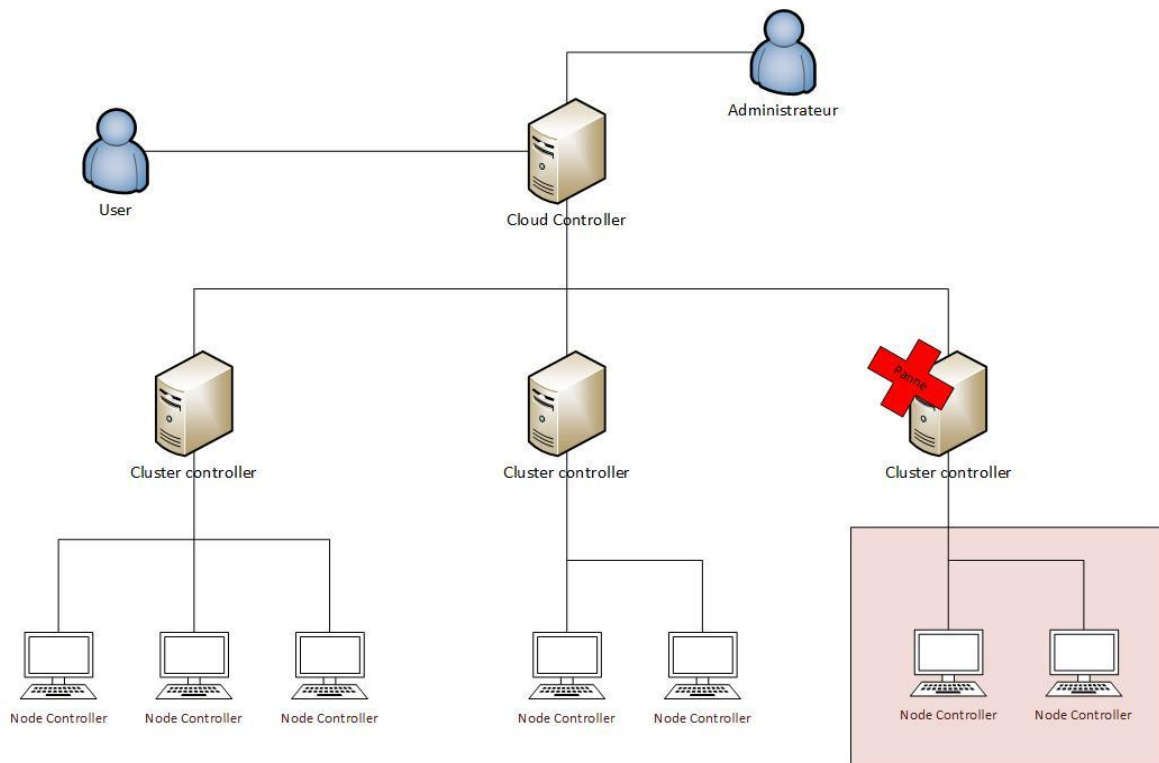


Figure 3.6 : Schéma de l'architecture du cloud privé : cas de panne d'un cluster Controller.

La solution à proposer devra introduire un fonctionnement distribué permettant à la fois la reprise des tâches et des fonctions assurées par le cloud Controller en cas de l'arrêt de ce dernier. Une approche de solution que nous avons trouvée à ce problème consiste à considérer le maillage en arbre des différents composants du système afin de déléguer aux clusters Controller la charge de faire des supervisions mutuelles et surtout celle de superviser le CLC comme le montre la figure 3.7 :

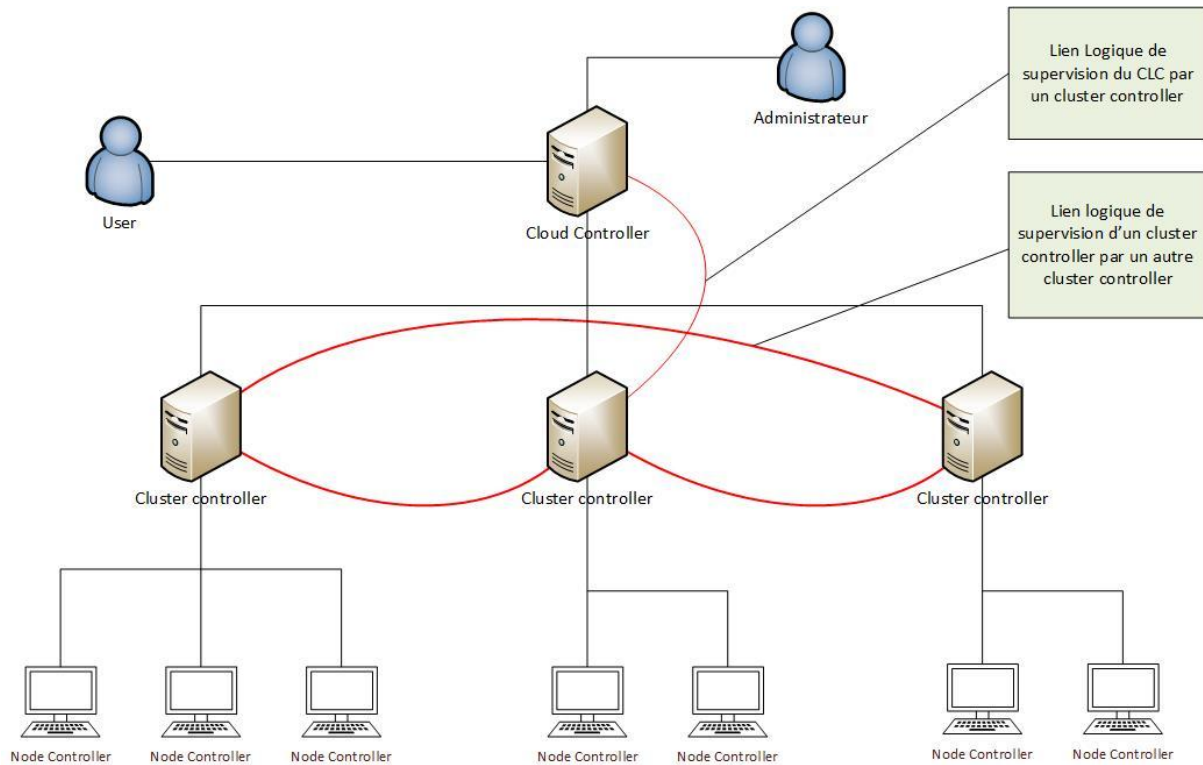


Figure 3.7 : schéma de la solution proposée

Sur cette figure on peut voir qu'un maillage logique complet a été créé entre les clusters controllers et qu'un lien logique a également été créé entre le cloud Controller et un cluster Controller. De cette manière, une panne localisée au niveau d'un cluster Controller ferait que la charge de ses nodes controllers passerait automatiquement à un autre cluster Controller. De même, lorsque le cloud Controller tomberait en panne, le cluster Controller en charge de sa supervision prendrait automatiquement sa place par changement d'état. Ce processus enclencherait immédiatement l'élection d'un nouveau cluster Controller pour la supervision du nouveau cloud Controller.

Au niveau du cloud Controller, il est donc question de s'assurer qu'un cluster Controller (CC) au moins soit élu et soit capable de reprendre les services du CLC en cas de panne de ce dernier (voir figure 3.8).

Sur cette figure on peut voir que la panne du cloud Controller entraine un changement d'état du cluster Controller chargé de le superviser et ce dernier devient ainsi le nouveau cloud Controller et un nouveau lien symbolique est créé avec un autre CC. De même lorsqu'un CC tombe en panne l'un des autres CCs prend en charge tous les nodes controllers du CC en panne comme le montre la figure 3.9 :

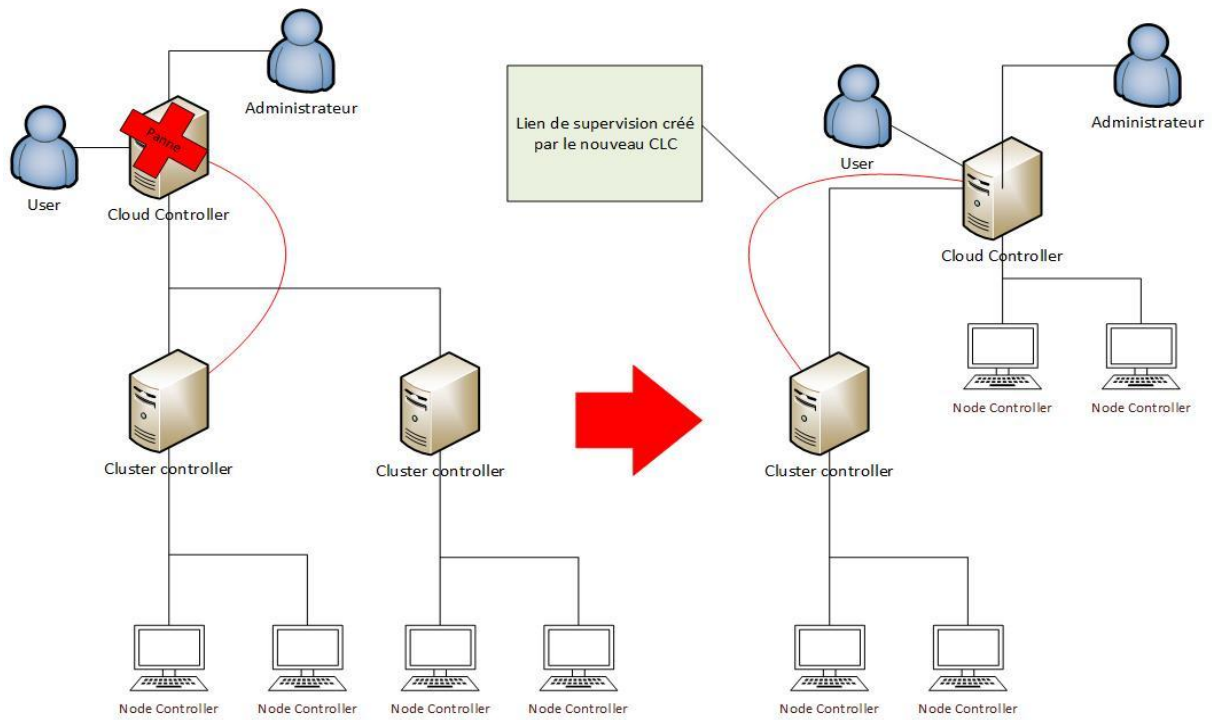


Figure 3.8 : Schéma d'un scénario de reprise des services du cloud Controller par un cluster Controller et réélections d'un nouveau cluster Controller de supervision.

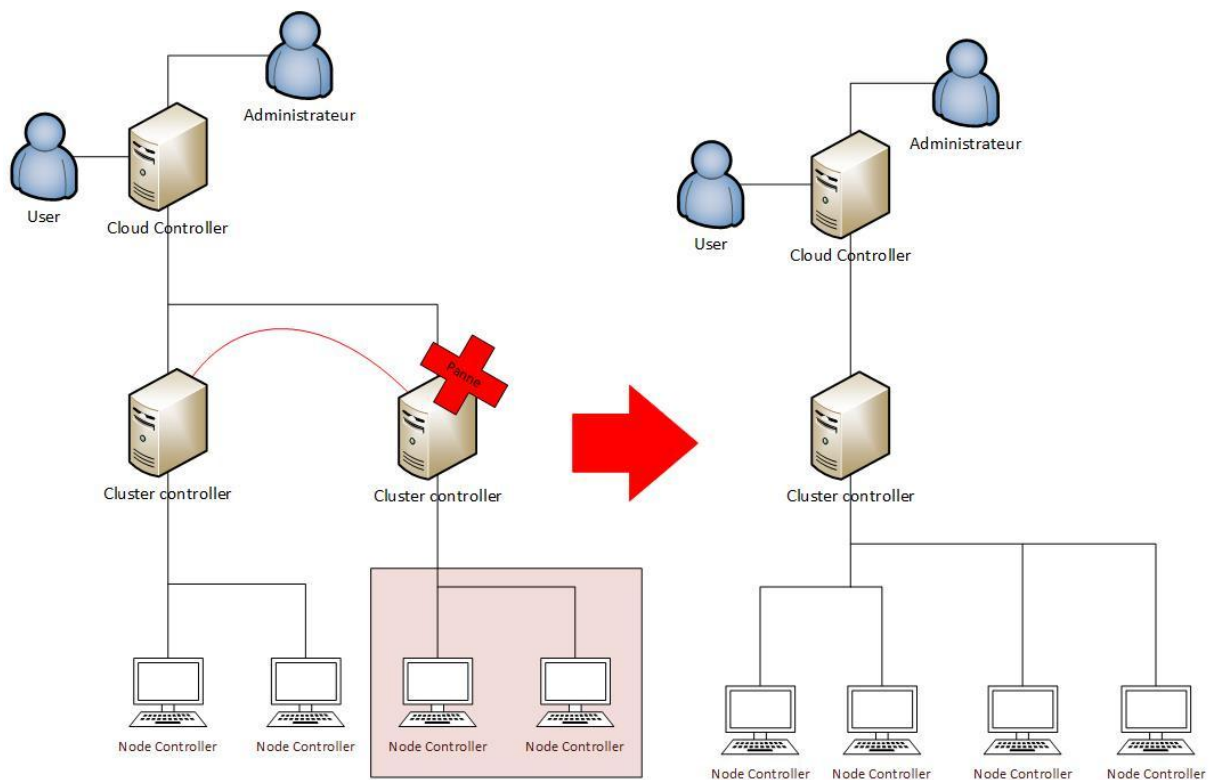


Figure 3.9 : Schéma d'un scénario de reprise des services d'un cluster Controller par un autre cluster Controller

Conclusion :

En définitive dans ce chapitre nous avons parlé des systèmes et des algorithmes distribués, des principes d'élections et les caractéristiques de ce genre de système. Nous avons également présenté et explication la réplication des données dans les systèmes distribués. Enfin nous avons également montré par des figures les différents scénarios possibles de reprise de services en cas de panne d'un nœud du système. Dans le chapitre suivant nous ferons une étude et une planification du projet.

CHAPITRE IV : IMPLEMENTATION DU PROJET ET CHOIX TECHNIQUES

Introduction :

La réalisation d'un projet exige un certain d'étapes indispensable pour qu'elle soit menée à bien. Dans ce chapitre nous faire allons donc faire l'analyse fonctionnelle des besoins, l'élaboration de la structure du projet, le diagramme des bêtes à cornes, l'élaboration de la charte du projet, le diagramme de GANTT et nous ferons également une analyse financière du projet. Enfin nous présenterons nos choix techniques pour sa réalisation.

IV.1 Etude et planification projet

IV.1.1 Analyse du besoin

IV.1.1.1 Analyse fonctionnelle du besoin

L'analyse fonctionnelle est une méthode de travail qui permet de définir les besoins d'un produit en termes de fonctions, au prix le plus juste.

Cette méthode utilise des outils qui permettent d'identifier des fonctions d'usage et d'estime. Elle permet de répondre à trois questions : A qui ? Sur quoi ? Pourquoi faire ? En formulant cette phrase : le produit rend service à qui ? En agissant sur quoi ? Pour satisfaire quel besoin ?

Grace aux réponses apportées à ces questions nous pouvons ressortir à la figure 4.1 le graphe normalisé de l'expression du besoin (diagramme de bête à cornes). Voir aux références numéro [36] [37] et [38].

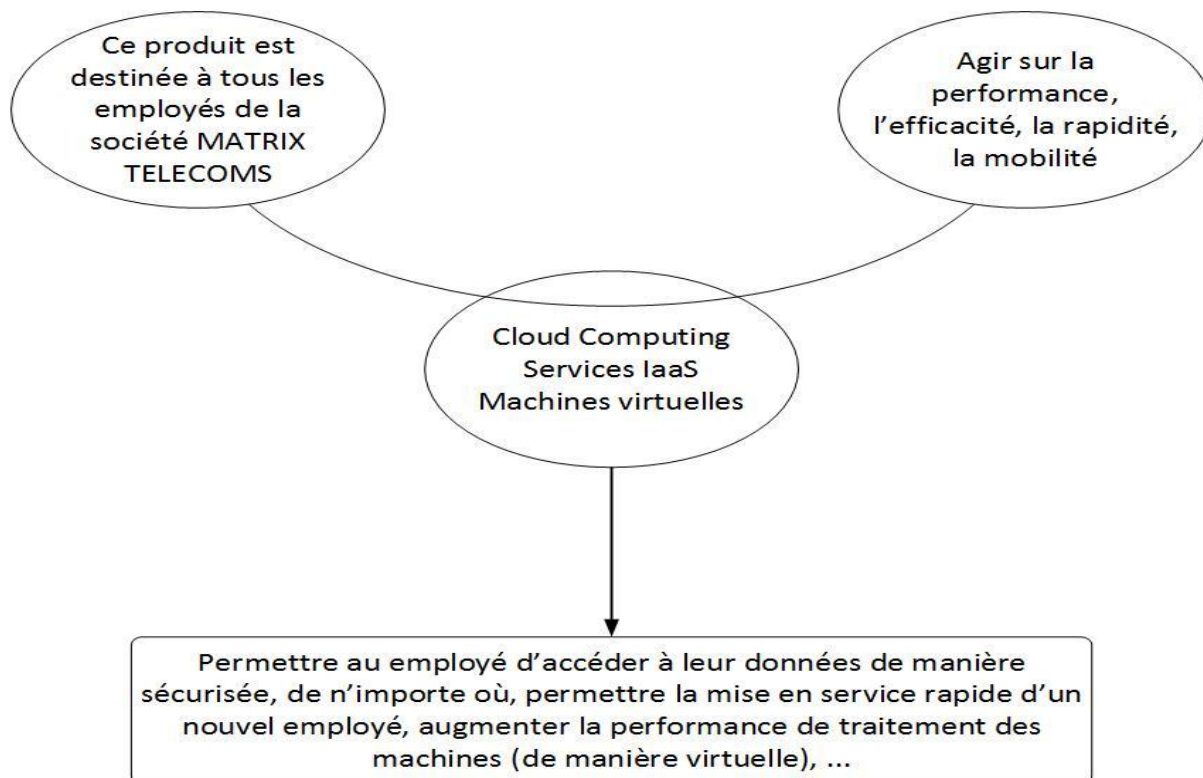


Figure 4.1 : Graphe normalisé de l'expression du besoin

Ainsi la prestation présentée dans le tableau 4.1 nous permet de préciser la valeur des critères à atteindre.

Tableau 4.1 : Valeur des critères à atteindre

La prestation précise la valeur des critères à atteindre	
Pourquoi le produit existe ?	Parce que l'obtention des bénéfices pour l'entreprise implique d'être plus performant
Qu'est ce qui pourrait faire évoluer le produit ?	Que les besoins des employés changent
Qu'est ce qui pourrait faire disparaître le produit ?	Que l'obtention des laptops très puissants devienne accessible et facile, que la capacité de stockage basique passe du Giga au Téra,...

IV.1.1.2 Elaboration de la structure de découpage du projet (WBS)

Le Project Managment Institut définit le WBS (**Work Breakdown Structure**) ou structure de découpage du projet comme une décomposition hiérarchique, axée sur les tâches et activités, du travail que l'équipe de projet doit exécuter pour atteindre les

objectifs du projet et produire les livrables voulus. La figure 4.1 nous montre le graphe WBS du projet.

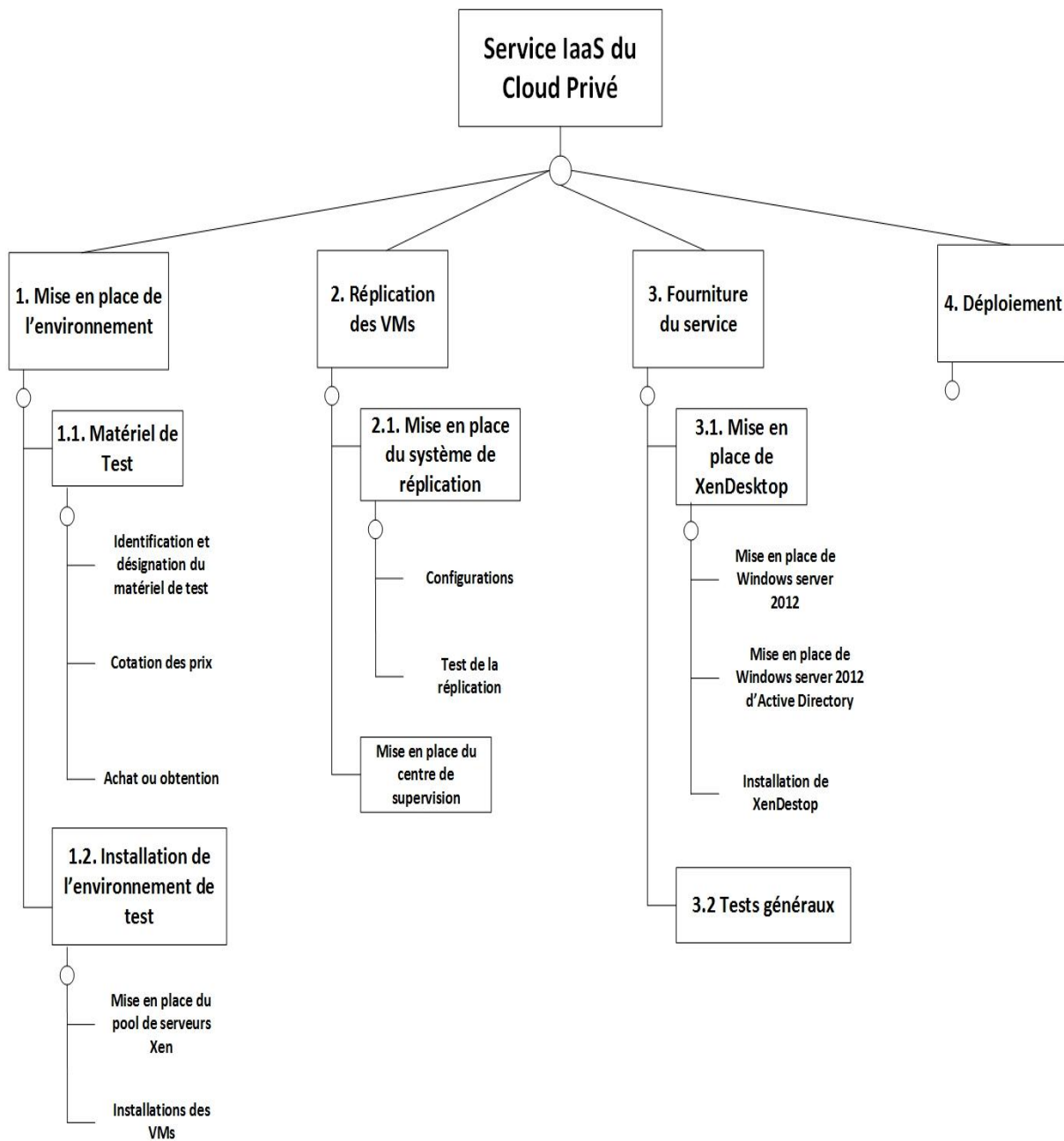


Figure 4.2 graphe WBS du projet

IV.1.2 Elaboration de la charte du projet

IV.1.2.1 Contexte et enjeux

Les différents outils technologiques utilisés offrent au personnel des collectivités une grande ouverture vers l'extérieur. Cette ouverture peut apporter des améliorations de performances importantes si l'utilisation de ces outils technologiques est faite à bon escient et selon certaines règles.

A l'inverse, une mauvaise utilisation de ces outils peut avoir des conséquences extrêmement graves. En effet, ils augmentent les risques d'atteinte à la confidentialité, de mise en jeu de la responsabilité, d'atteinte à l'intégrité et à la sécurité des fichiers de données personnelles (virus, intrusions sur le réseau interne, vols de données).

De plus, mal utilisés, les outils informatiques peuvent aussi être une source de perte de productivité et de coûts additionnels.

L'application des nouvelles technologies informatiques et de communication permettent de préserver le système d'information, le bon fonctionnement des services et les droits et libertés de chacun. Les chartes sont trop souvent considérées comme un moyen de contrôle du travail des agents. Elles doivent être expliquées au personnel.

IV.1.2.2 L'objectif

La présente charte informatique est un code de déontologie formalisant les règles légales et de sécurité relatives à l'utilisation de tout système d'information et de communication au sein de l'entreprise.

Tout manquement, selon sa gravité, est susceptible d'entraîner pour l'utilisateur des sanctions disciplinaires, et ce sans exclusion d'éventuelles actions pénales ou civiles à son encontre.

L'utilisateur pourra, en outre, voir ses droits d'accès aux ressources et système d'information et de communication suspendus ou supprimés, partiellement ou totalement.

IV.1.2.3 Le champ d'application

La présente charte s'applique à l'ensemble du personnel tous statuts confondus de MATRIX TELECOMS, ainsi qu'au personnel temporaire. Elle s'applique également à tout prestataire extérieur ayant accès aux données et aux outils informatiques de la collectivité.

IV.1.2.4 Les règles générales d'utilisations des VMs

Le personnel de MATRIX Télécoms devra respecter les règles suivantes pour l'accès aux services du Cloud privé :

Un laptop sera mis à la disposition de chaque utilisateur et tous le nécessaire y seront préinstallés pour lui permettre d'accéder à sa machine virtuelle via le Cloud. L'utilisateur ne devra en aucun cas ou moment installer quoi que ce soit dans ce laptop.

une VM puissante et contenant tous son environnement de travail sera mise à sa disposition. Son environnement sera installé selon sa fonction au sein de l'entreprise. Un serveur de Tests virtuel sera également mis à sa disposition pour d'éventuels tests.

L'utilisateur pourra faire une demande en bonne et due forme pour l'obtention d'une VMs personnalisée.

Il sera responsable de tous dommages causé à sa VM et devra assurer une sécurité minimal notamment avoir un mot de passe fort pour les accès.

IV.1.3 Diagramme de cas d'utilisation

Dans le cas d'utilisation global on a deux acteurs principaux qui interagissent avec le système à savoir l'administrateur et l'employé de MATRIX comme le montre la figure 4.3 qui montre les différents cas d'utilisation du système :

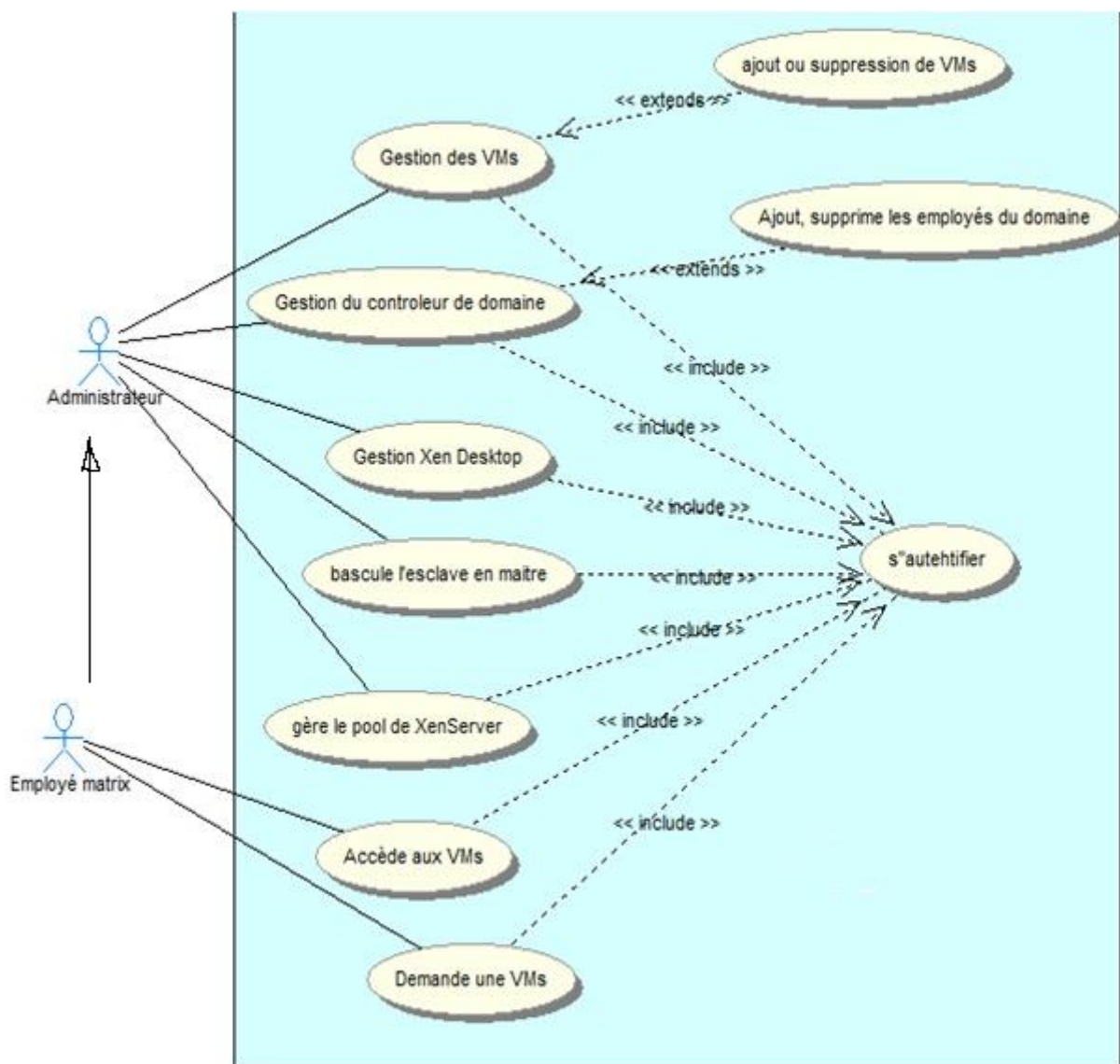


Figure 4.3 : cas d'utilisation global du système

IV.1.4 Estimation des ressources matérielles et financière

Le tableau 4.2 suivant montre une estimation financière et détaille de l'ensemble des matériels minimum nécessaires et les plus importants pour le déploiement en grandeur nature dans le cadre de l'entreprise :

Tableau 4.2 : estimation financière des ressources matériels

Désignations du matériel	Caractéristiques	Quantité minimale	Prix sur le marché mondial à l'unité	Existant en entreprise	observations
Routeur mikrotik	N'importe quel type ferait l'affaire	1	Ente 59\$ et 600\$	Oui	Prix d'Amazone et il varie selon le pays et la performance du routeur
Serveurs	Cœurs : 96 RAM : 128 Go Stockage : 2x30To 3 cartes réseaux	2	Entre 7860\$ et 11659\$	oui	Prix d'Amazone et varie selon le pays et la marque
Une bonne connexion internet	1Mbps/Poste	//	//	Oui	Varie selon le FAI
Xen Desktop	Un DVD Xen Desktop	1		Non	Prix du site de Citrix et la licence est valable pour un poste

IV.2 Choix technique

IV.2.1 La solution de virtualisation employée

Nous avons étudié en amont plusieurs solutions de virtualisation, et notre choix s'est porté sur l'hyperviseur XEN pour plusieurs raisons :

- **Administration** : Xen sépare l'exécution de l'hyperviseur de l'administration du système d'exploitation, la pile d'administration, les pilotes pour les équipements et les systèmes invités (guest OS).
- **Choix du système d'exploitation** : prend charge n'importe quel type de système d'exploitation.
- **Sécurité** : chaque système invité est complètement isolé. Le code hyperviseur est réduit au minimum pour éviter tout domaine d'attaque. L'hyperviseur ne peut attaquer les invités et vice-versa.
- **Performance** : Xen optimise l'usage des CPU, de la mémoire et du réseau grâce à la para-virtualisation et au concept de pass-through.

De plus XEN est solution open source proposée par la société CITRIX. Cette dernière propose également un ensemble d'outils qui fonctionne parfaitement bien avec l'hyperviseur XEN mais aussi avec plusieurs autres hyperviseurs. Ces outils permettent de superviser et d'administrer le système et aussi de servir les VMs. Et ils constituent notre portail web des services IAAS (voir la référence N° [39]).

IV.2.2 Le portail web des services IAAS

L'accès aux services IAAS se fait grâce à un ensemble d'outils qui sont proposés par la société CITRIX et qui sont des solutions standards pour assurer la sécurité et la convivialité des services IAAS on a:

- **XenCenter** [39] [40]: est une console de gestion qui permet de réaliser les opérations de gestion et de maintenance sur les hôtes d'un cluster. Cet outil qui est natif à XenServer et constitue notre centre de supervision, il est téléchargeable via un navigateur en tapant juste l'adresse IP du serveur Xen dans la barre URL. XenCenter ne fonctionne que sous OS Windows, mais il existe un autre centre de supervision pour les administrateurs qui préfèrent les OS Linux à savoir Orchestra une documentation est présentée en annexe 3.

Suivre la référence N°1 et N°2 pour le téléchargement, l'installation et l'utilisation de XenCenter (installation des VMs).

- **XenDesktop et Xen Receiver** [41] : XenDesktop de Citrix permet de mettre à disposition des bureaux (virtuels) pour une liste d'utilisateurs prédéfinis. Cette technologie permet d'économiser de l'argent, car seul le serveur doit être performant (là où tournent les bureaux virtuels utilisés par les utilisateurs autorisés). Les clients ne reçoivent qu'une image (une série de pixels) du bureau utilisé à distance. Il est important de noter que XenDesktop utilise le protocole RFB (VNC) et ne s'installe que sous l'OS Windows server 2012 R2. Le client permettant d'utiliser un bureau (virtuel ou non) à distance est nommé "Citrix Receiver". Ce programme peut être téléchargé depuis le serveur XenDesktop que vous allez installer.

Suivre la référence N°3 pour le téléchargement et l'installation de XenDesktop.

Ces outils ainsi cités constituent notre portail web pour l'accès et la supervision services IAAS.

IV.2.3 Synthèse des outils utilisés

Le tableau 4.3 qui suit nous montre la synthèse des outils logiciels qui ont été implémentés pour la réalisation de ce projet.

Tableau 4.3 : Synthèse des outils utilisés pour l'implémentation

N°	Désignations	Choix	Versions	Lieux d'installations
1	Systèmes d'exploitation	Windows Server	2012 R2	1 serveur
		Windows 7	7	Machine administrateur
		Mikrotik OS	5.18	Routeur Mikrotik
2	Solution de virtualisation	XenServer	6.5.0	2 serveurs

3	Méthode de supervision des machines	XenCenter		6.5	Machine administrateur
4	Protocole d'accès aux machines	VNC	Client : Xen Receiver	3.9.1	Machine cliente
			Serveur : Xen Desktop	7.9	Serveur Windows Server 2012
5	Protocole de communication	XML-RPC (org.apache.xmlrpc)		3.1.3	//
6	Réplication des VMs	DRBD (drbd-utils)		8.4.3-2	Serveurs XenServer
7	Accès à distance aux XenServer	Putty		0.6	Machine administrateur
8	Partage de fichiers entre les nœuds du réseau	WinSCP		5.7.6	Machine administrateur

Conclusion :

A ce niveau, nous savons exactement ce qu'il y a faire pour implémenter notre projet et comment y arriver. En effet dans ce chapitre nous avons fait une étude détaillée du projet, nous avons également fait une étude financière du matériel et enfin nous avons présenté notre choix technique. A présent nous allons passer aux simulations et aux commentaires.

CHAPITRE V : RESULTATS ET COMMENTAIRES

Introduction

L'implémentation de notre solution a consisté à mettre en place une plateforme pour le déploiement des services IAAS avec tolérance aux pannes. Afin de montrer les performances de notre solution et sa capacité à tolérer les pannes dans un environnement de cloud privé IAAS, nous l'avons déployée sur un ensemble machines physiques reliées dans un réseau local LAN (Local Area Network). Dans ce chapitre qui est le dernier de ce document, nous allons présenter notre architecture de tests ainsi que les détails des ressources utilisées, les résultats obtenus lors de l'implémentation du projet ainsi que quelques configurations indispensables pour la réplication. Enfin nous ferons des commentaires.

V.1 Résultats

V.1.1 Architecture de tests

Avant le déploiement en grandeur nature, des tests sont indispensables ; pour ce faire nous avons opté pour une architecture client/serveur comme le montre la figure 5.1 dont les configurations IP et softwares sont décrits dans le tableau 5.1

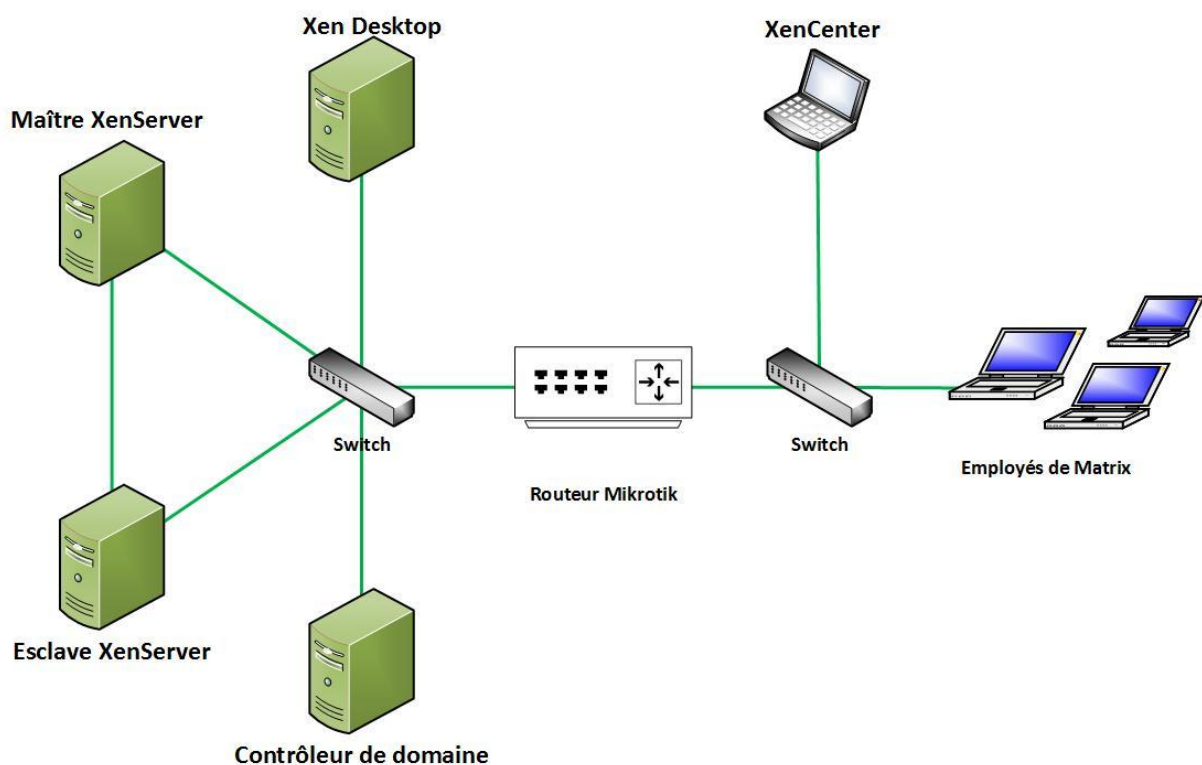


Figure 5.1 : Architecture de tests

Tableau 5.1 : récapitulatif des équipements et de leurs caractéristiques

désignation	caractéristiques	Configurations réseaux	Softwares	commentaires
Maître XenServer	Processeur : Core i5 RAM : 6Go DD : 2 Disques Durs de 500Go 2 cartes réseaux	eth0 IP : 10.0.0.1/24 eth1 IP : 10.0.1.1	OS : XenServer 6.5.0	Il s'agit d'un desktop HP qui est le serveur maitre des VMs
Esclave XenServer	Processeur : Core i5 RAM : 6Go DD : 2 Disques Durs de 500Go 2 cartes réseaux	eth0 IP : 10.0.0.1/24 eth1 IP : 10.0.1.1/24	OS : XenServer 6.5.0	Il s'agit d'un desktop HP qui est le serveur esclave des VMs
Contrôleur de domaine	Processeur : Core i3 RAM : 2Go DD : 1 Disque dur de 350Go	IP : 10.0.0.3 Masque : 255.255.255.0 Passerelle : 10.0.0.254	OS : Windows Server 2012 R2 Active Directory DNS Autorité de certification	Desktop HP qui est notre Serveur Active directory et DNS, avec notre autorité de certification locale
Xen Desktop	Processeur : Core i3 RAM : 2Go DD : 1 Disque dur de 350Go	IP : 10.0.0.4 Masque : 255.255.255.0 Passerelle : 10.0.0.254	OS : Windows Server 2012 R2 Xen Desktop	Desktop HP qui est notre serveur de bureaux distants aux VMs
Routeur Mikrotik	processeur : Core i3 RAM : 1Go DD : 1 Disque dur de 350Go 2 cartes réseaux	IP : WAN ; 10.0.0.254 IP : LAN ; 192.168.0.254 Masques : 255.255.255.0	MIKROTIK 6.5 OS	Desktop HP qui est notre routeur mikrotik
XenCenter	Processeur : dual core RAM : 3Go DD : 1 Disque dur de 160Go	DHCP	OS : Windows 7 Citrix Receiver	Laptop HP qui est notre machine administrateur et cliente
Deux Switchs	Switch TP Link simple 8 ports	//	//	Il nous permet l'interconnexion des serveurs

V.1.2 Quelques interfaces

V.1.2.1 XenServer

Après l'installation des XenServer on peut voir son écran d'accueil qui s'affiche comme le montre l'illustration 5.1 ; sur cet écran est présenté toutes les caractéristiques de notre serveur.

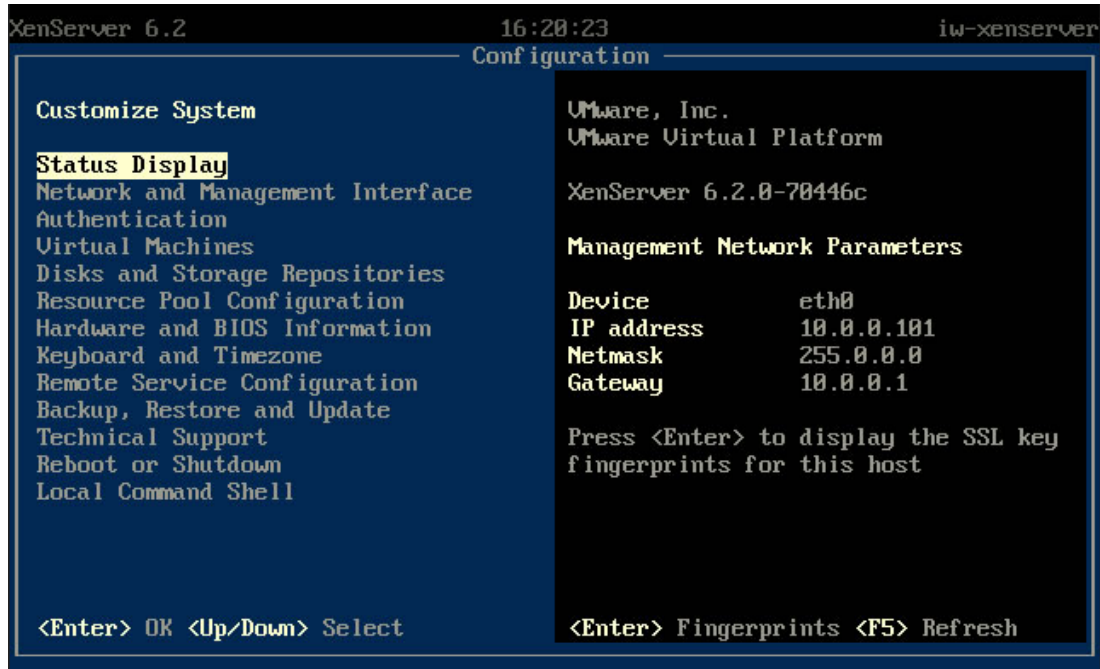


Illustration 5.1 : Ecran principal de XenServer

V.1.2.2 Centre de supervision

L'illustration 5.2 suivante nous montre l'écran d'accueil de XenCenter.

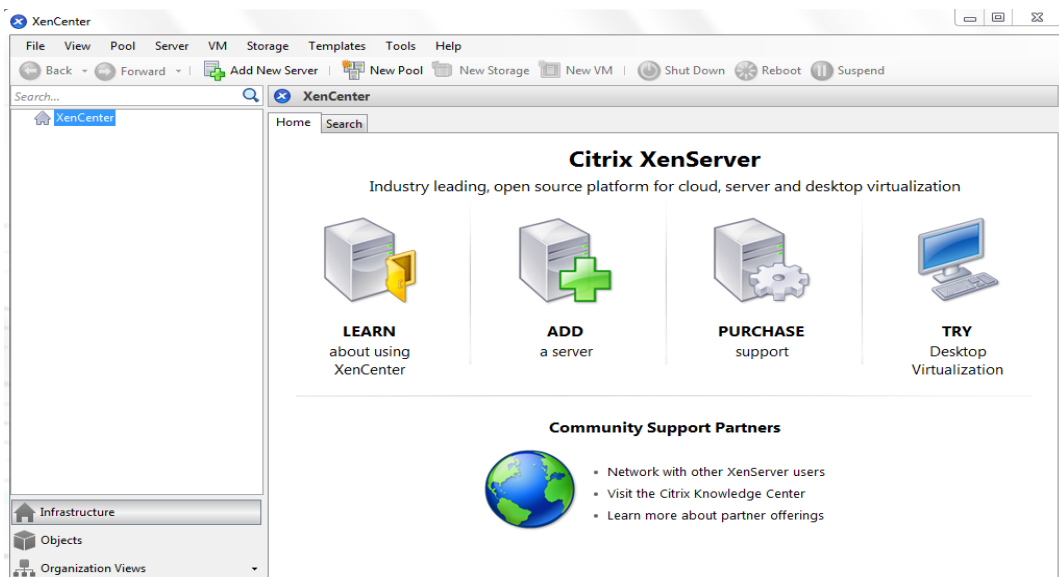


Illustration 5.2 : Ecran d'accueil de XenCenter

Sur l'illustration 5.2 on peut voir qu'on a la possibilité d'en apprendre plus sur XenCenter, ajouter des serveurs XenServer, avoir accès aux supports Citrix et enfin tester la virtualisation en ligne.

Sur cet interface on peut voir sur le volet de gauche notre pool de serveurs ainsi que les VMs qui y sont installées, on peut également voir les propriétés du serveur, gérer la mémoire, le stockage, le réseau, avoir un aperçu visuel des serveurs et des VMs, gérer les performances, les utilisateurs et faire des recherches.

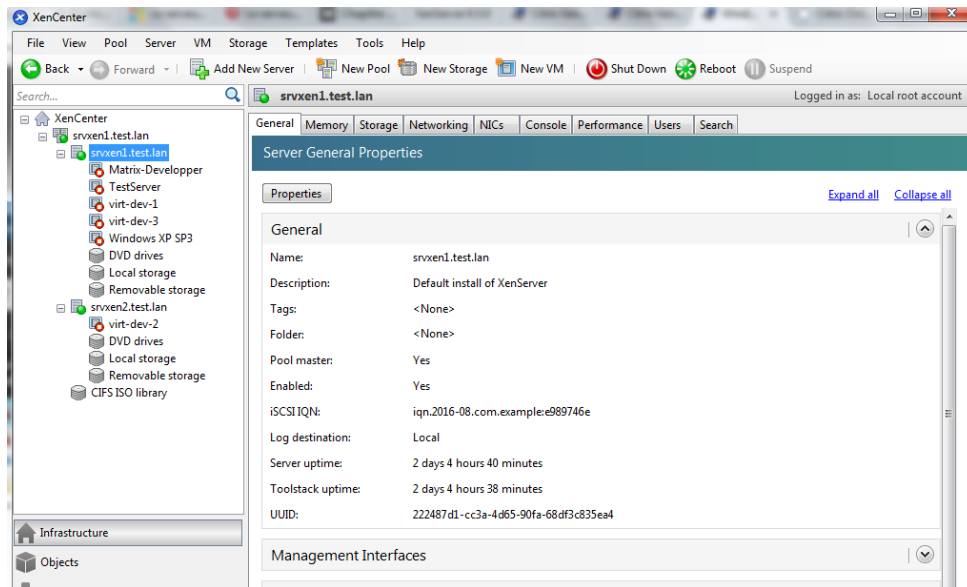


Illustration 5.3 : interface de propriétés de XenServer

On peut voir sur cet écran que en cliquant sur l'onglet « Console » on peut avoir accès au bureau distant de la VM sélectionnée.

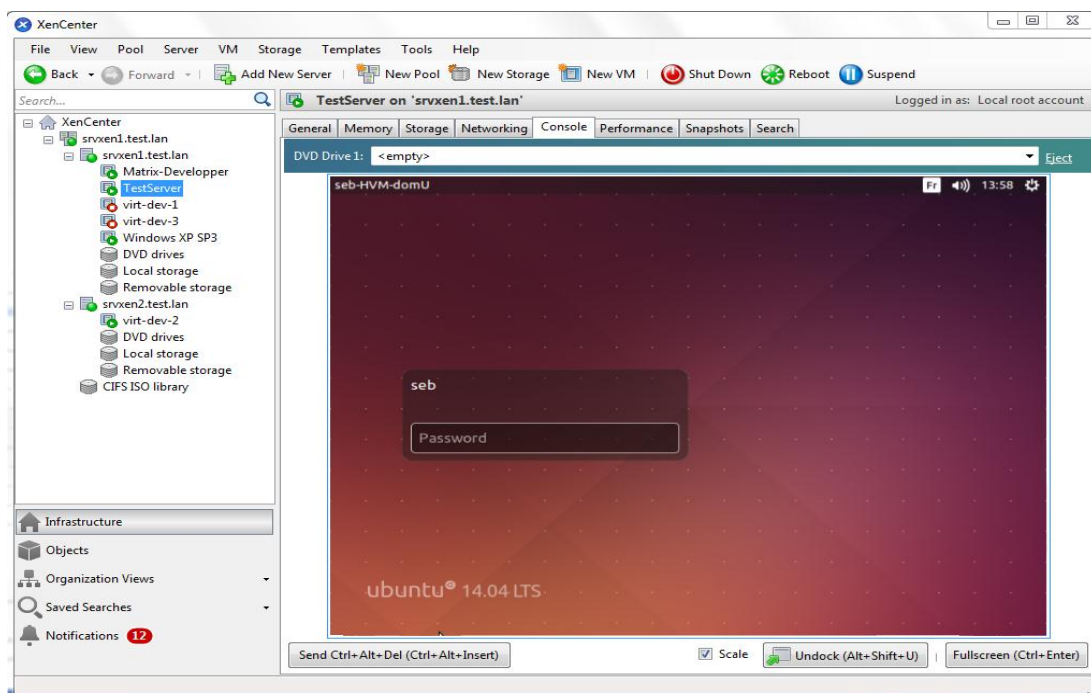


Illustration 5.4: Accès au bureau distant via XenCenter

L'écran suivant nous montre les performances CPU du « server1.test.lan »

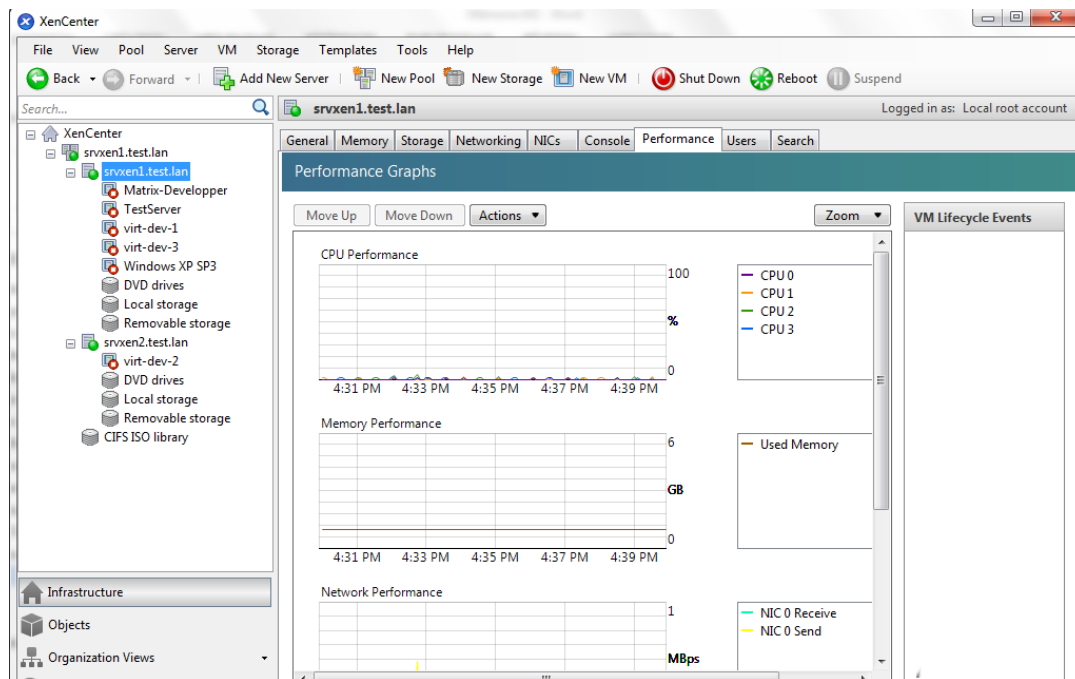


Illustration 5.5 : Interface de supervision des performances de XenServer

L'écran qui suit nous montre les performances mémoire en termes de RAM du serveur ainsi que de toutes les VMs tournant sur lui.

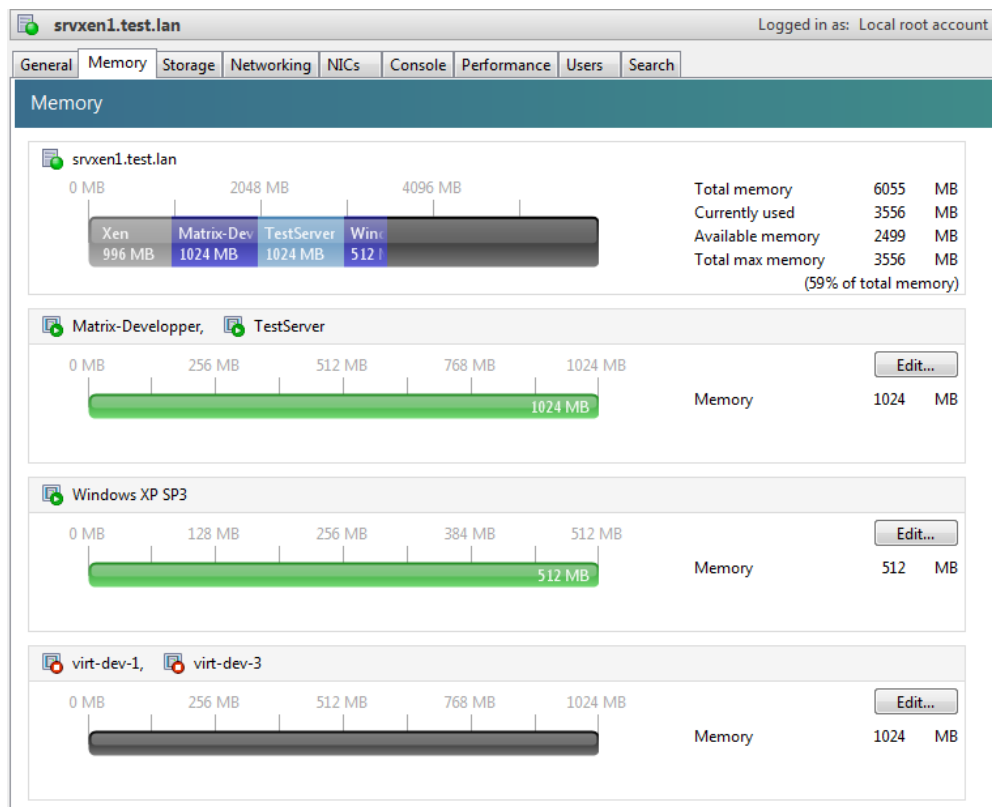


Illustration 5.6 : Interface de supervision de la mémoire

V.1.2.3 Xen Desktop et Xen receiver

Xen Desktop étant le serveur, son interface de management est Citrix Studio : il a une interface très agréable et conviviale ; les illustrations 5.7 et 5.8 représentent respectivement la page d'accueil avant et après les configurations de Citrix Studio.

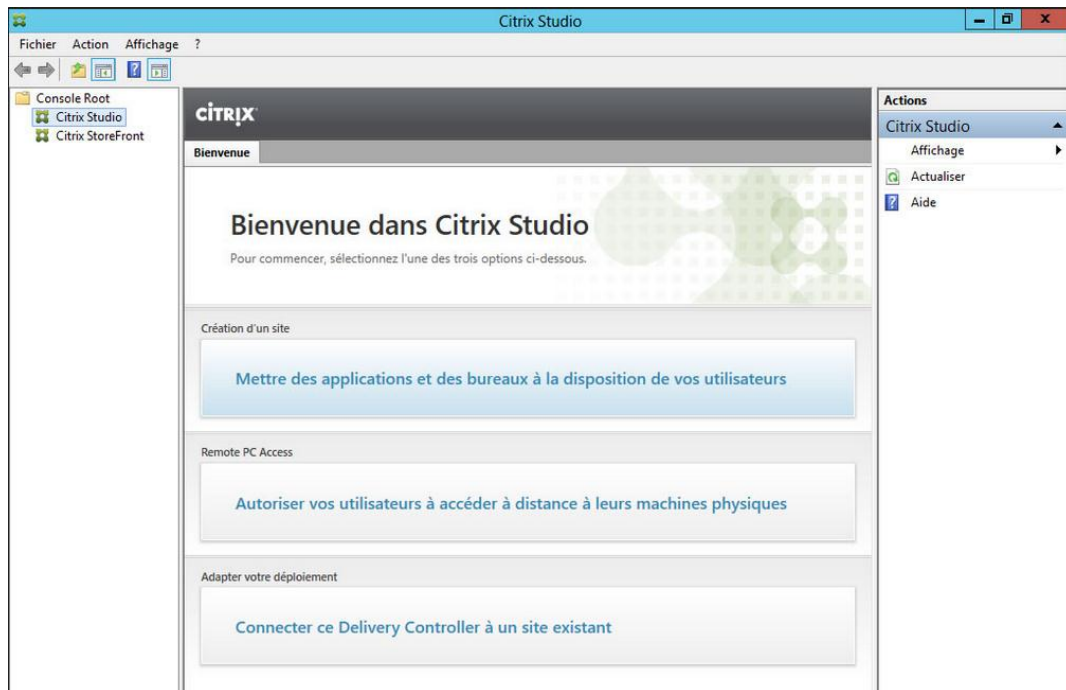


Illustration 5.7 : Ecran d'accueil du centre de management Citrix Studio

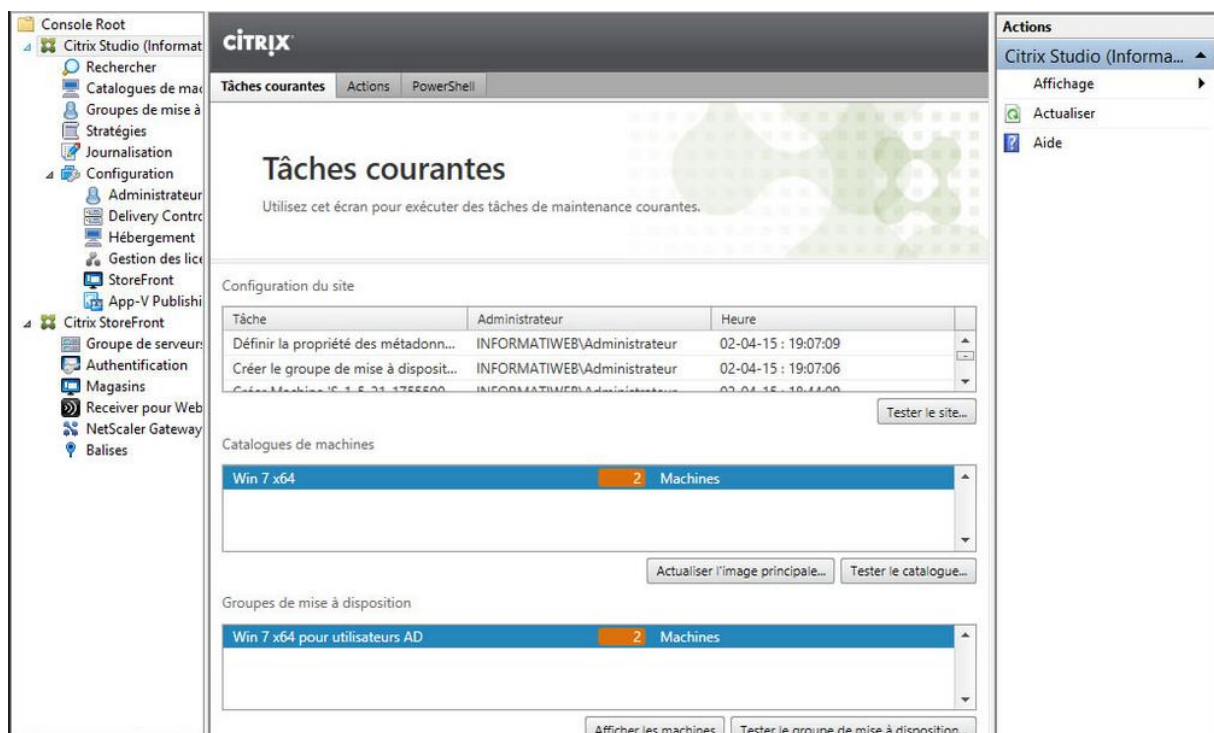


Illustration 5.8 : Ecran de Citrix Studio configuré

Pour accéder aux machines virtuelles, la machine cliente doit avoir Citrix Receiver installé (pour l'installation et les configurations voir à la référence [41]). Ainsi il pourra ajouter l'adresse du serveur Xen Desktop, se connecter de manière sécurisée et enfin avoir accès à sa VM comme le montre les illustrations suivantes :

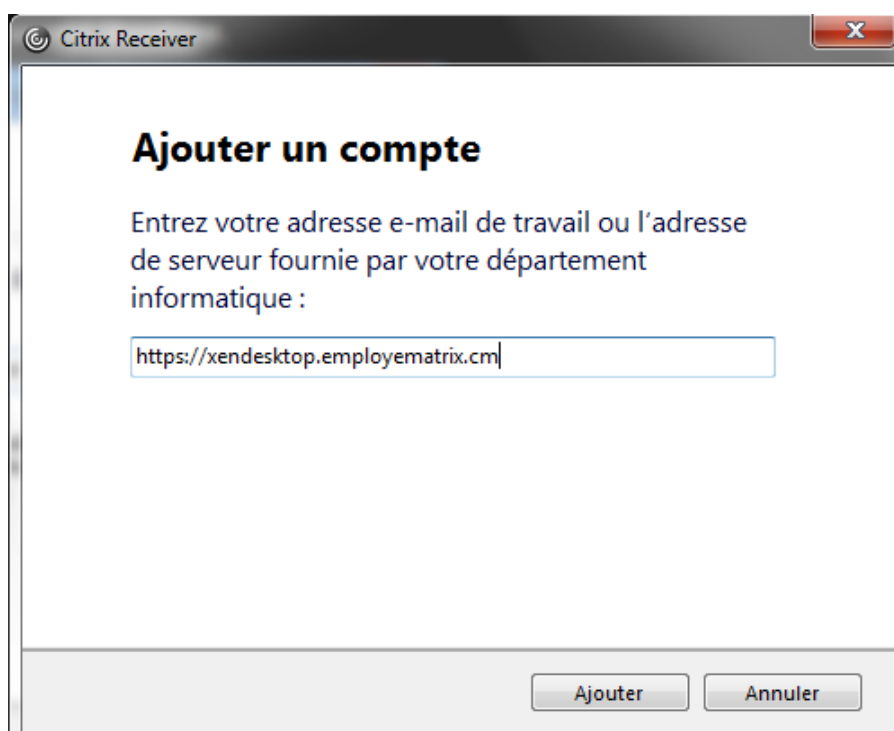


Illustration 5.9 : Ajout d'un compte à Citrix Receiver

Sur cet écran on s'authentifie en entrant : nom du domaine\nom d'utilisateur dans le domaine et ensuite on entre son mot passe.



Illustration 5.10 : Ecran de connexion à Citrix Receiver

L'écran d'accueil de Citrix Receiver s'affiche enfin et pour accéder aux bureaux publiés (ceux que vous pouvez utiliser), cliquez sur le +.

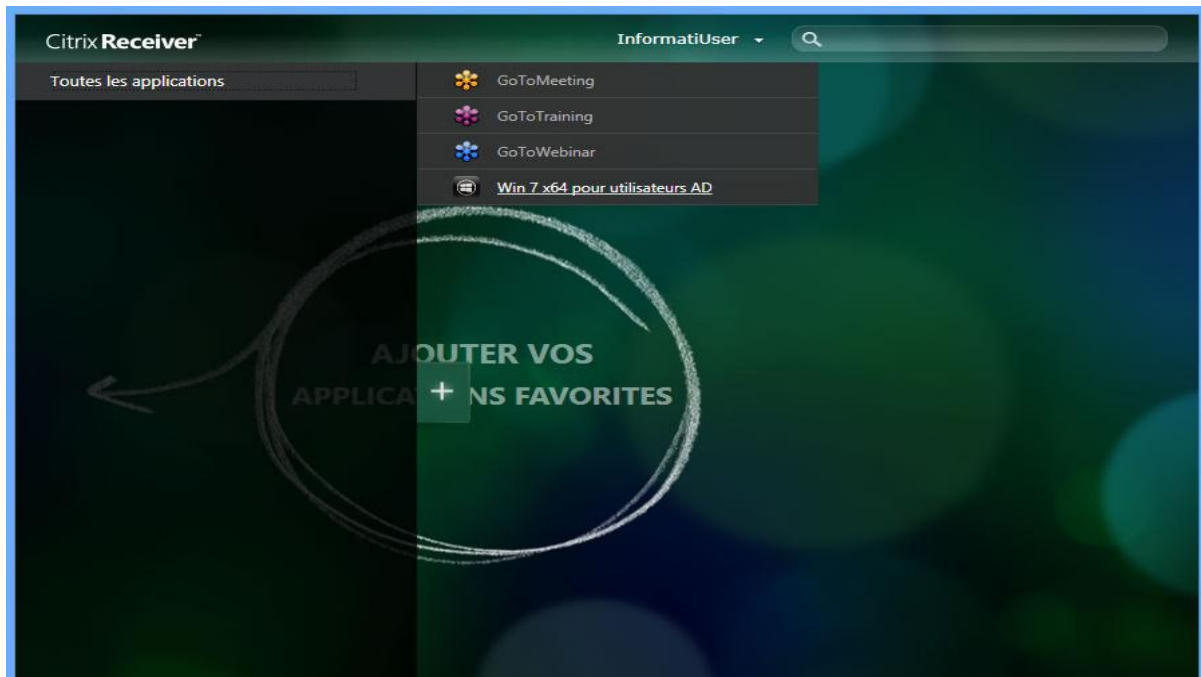


Illustration 5.11 : Ecran d'accueil de Citrix Studio

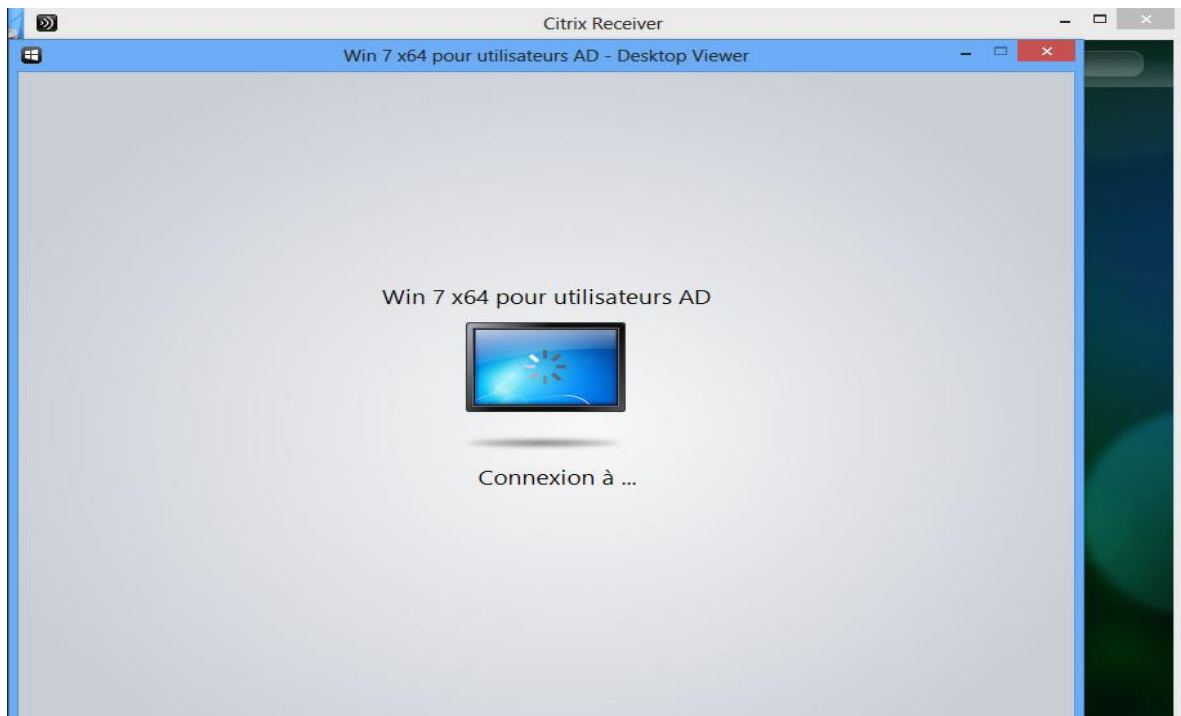


Illustration 5.12 : Ecran de démarrage de la machine virtuelle

Sur les illustrations 5.12 et 5.13 on peut voir que notre VM qui tourne sous Windows 7 démarre et s'affiche parfaitement bien. Ainsi on peut travailler exactement comme si on le faisait sur une machine physique.

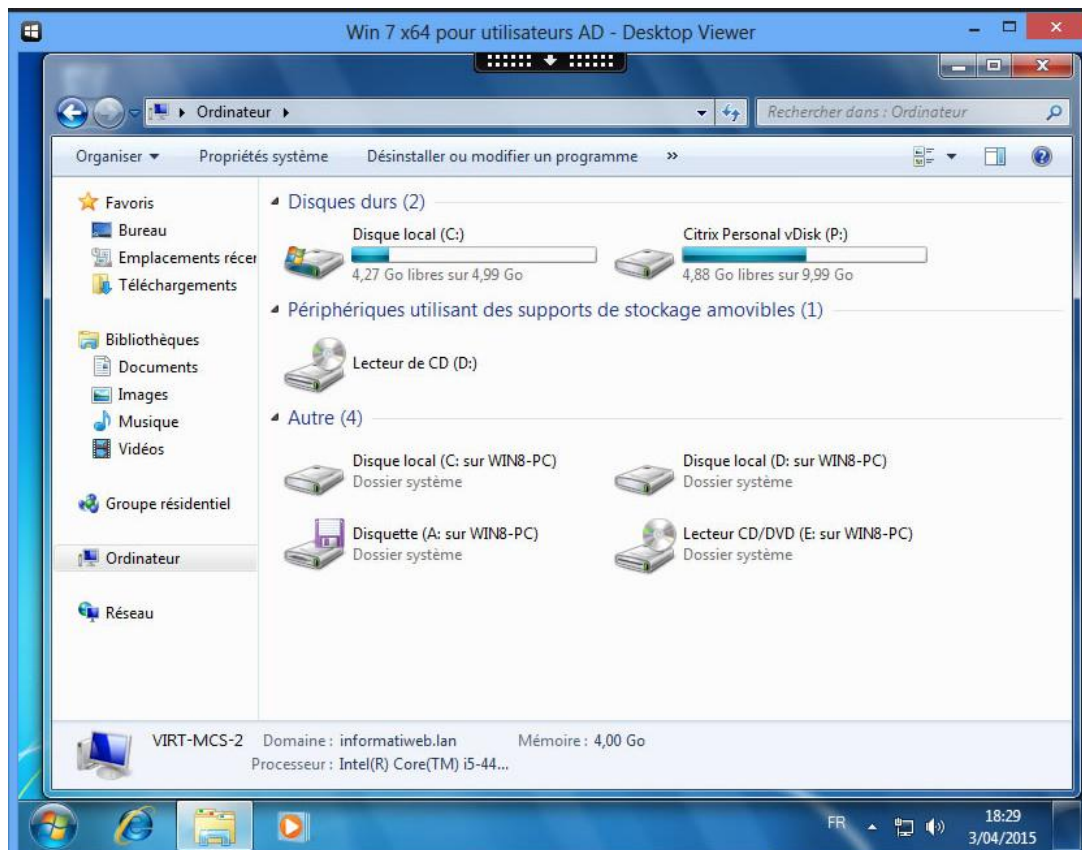


Illustration 5.13 : Accession à la machine virtuelle

V.2 Commentaires

La conception de notre système et les tests effectués ont confirmé sa capacité à assurer la continuité des services dans un environnement de cloud privé. En termes de fonctionnalités, la solution permet de gérer un ensemble de machines physiques et virtuelles à travers une interface d'administration CLI mais aussi à travers un portail Web qui offre une utilisation beaucoup plus flexible du système. La solution est sécurisée car la connexion aux VMs ne se fait que via le protocole HTTPS de plus les machines clientes ayant accès aux serveurs doivent être dans un contrôleur de domaine.

Comme nous l'avons vu en amont la réplication est indispensable pour assurer une disponibilité constante des services surtout en cas de crash de l'un des serveurs.

Les serveurs XenServer doivent avoir impérativement la même architecture et doivent être dotés d'au moins deux disques durs distincts ainsi que de deux cartes réseaux pour nous permettre de faire la réplication.

Le scénario est simple ; les machines virtuelles sont installés sur le serveur XenServer Maitre (elles peuvent également être installées sur le serveur esclave) et une instance de chaque VM est répliquée sur le serveur esclave qui est en attente. Mais en attente de quoi ? En effet lorsque le serveur Maitre tombe en panne un basculement

manuel ou automatique est fait pour changer le serveur esclave en maitre ainsi les services ne sont pas arrêtés et les données ne sont pas perdues. Ce basculement permet bien attendu de faire les maintenances nécessaires sur le serveur maitre tombé en panne.

Le choix de l'outil de réplication s'est porté sur DRBD car il permet de faire de la distribution de tâches et de la réplication synchrone, il est un standard dans le monde de la réplication et n'est relativement pas très difficile à mettre en place.

Vous trouverez en annexe 4 l'ensemble des configurations nécessaires pour la mise en place de DRBD sur deux serveur XenServer en architecture Maitre/Esclave grâce à LVM.

Conclusion :

A cette étape, nous tendons vers la fin du projet qui doit se terminer par un déploiement dans les serveurs de MATRIX TELECOMS. En effet ce chapitre nous a permis de mettre du concret dans toutes les notions abstraites que nous avons étudié jusque-là et ainsi éclairer quelques points d'ombres. Pour se faire, nous avons présenté notre architecture de test avec les caractéristiques de tous les éléments le constituant puis nous avons également présenté quelques interfaces principales de la plateforme mise en place et en nous avons fait des commentaires. Nous devons donc à présent conclure notre travail tout en donnant les perspectives.

CONCLUSION ET PERSPECTIVES

L'objet de ce projet était de faire l'étude et la mise en place d'une solution de cloud Computing privé basée sur un algorithme de supervision distribuée et appliquée aux services IAAS dans le contexte de l'entreprise MATRIX TELECOMS. La motivation d'une telle solution a été nourrie par le désagrément qu'occasionnent les pertes de données, de temps, ... due à des problèmes techniques du matériel informatique des employés de la structure.

Bien que le cloud Computing soit une technologie nouvelle dans le monde et encore plus au Cameroun, il existe de nombreuses solutions pour le mettre en place. Celle que nous avons choisie intègre un très grand nombre de fonctionnalités telles que le démarrage, l'arrêt et la migration des machines virtuelles via deux interfaces : une interface CLI et une interface web. Le système permet également l'accès à distance aux machines virtuelles et hôtes du réseau en employant le protocole VNC. La solution supporte la tolérance aux pannes en faisant la réplication des données dans une architecture maître/esclave. Elle offre également une portabilité intéressante qui permet son intégration facile dans les applications mobiles. Nous l'avons surtout employé pour tester notre système et permettre un déploiement rapide de la solution sur un ensemble d'hôtes du réseau local. Mais elle

Concernant les questions de l'utilisation de notre solution pour le cloud privé, les fonctionnalités actuelles permettraient son emploi en privé dans un réseau local pour les opérations comme le test d'applications en offrant de façon dynamique des machines virtuelles à différents utilisateurs. Le protocole d'accès à distance VNC utilisé permettrait d'accéder à ces machines pour le travail. En termes de sécurité, l'accès au système est subordonné à une authentification par un couple identifiant et mot de passe ainsi qu'à un contrôleur de domaine Active Directory.

De plus la solution peut être transformée en un produit fournisseur de revenu à l'entreprise car la migration au cloud public est très facile et n'exige pas un gros financement. D'après une étude effectuée, on estime que qu'en 2020, 840 millions de personnes dans le monde utiliseront des solutions « Cloud Computing », en faisant économiser un minimum de 210 milliards annuels aux entreprises [7].

BIBLIOGRAPHIE

Ouvrages

- [7] K. Michelin et S. D'Anna, «CE QU'IL FAUT SAVOIR SUR LE CLOUD COMPUTING,» avril, 2012.
- [8] P. CODO, «Chapitre 1 : Les concepts du cloud computing,» [En ligne]. Available: <http://www.institut-numerique.org/chapitre-1-les-concepts-du-cloud-computing-51c0279ca534a>. [Accès le 14 mai 2016].
- [10] N. HENNION, Introduction aux technologies cloud, 2011.
- [11] Wikipédia, «Data as a service,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Data_as_a_service. [Accès le 10 09 2016].
- [29] B. Yahia, Introduction aux Systèmes Distribués, Université M'hamed Bougara de Boumerdès.
- [30] C. Morin, «Architectures et systèmes distribués tolérants aux fautes,» L'Université de Rennes 1, 5 mars 1998.
- [31] B. Lélia, Algorithmique répartie, 2015.
- [32] A. labourel, Algorithmique Distribuée Problèmes et Algorithmes Fondamentaux, Marseille: Aix-Marseille Université, 2014.
- [33] S. Gançarski et H. Naacke, Réplication des bases de données, Université Paris 6, 2006.
- [36] J.-C. JACQUIOT, L'analyse fonctionnelle, Paris: jean-claude.jacquot@case-france.com, 2010.

Articles

- [1] L. Ronfaut, «Plus d'un Français sur trois se connecte à Facebook chaque jour,» *LE FIGARO.fr*, 03 03 2016.
- [34] G. LELARGE, «Centre de compétences PostgreSQL,» *magazine GNU/Linux Magazine France, numéro 131 (Octobre 2010)*, n° 1131, Octobre 2010.

- [35] G. Lelarge, «Mise en place de la réplication avec PostgreSQL 9.0 - 1/2,» *magazine GNU/Linux Magazine France*, n° %1131 , Octobre 2010.
- [43] COSIT, «consultation et services technologiques,» [En ligne]. Available: <http://www.cositafrique.net/nos-services/cloud-computing>. [Accès le 15 mai 2016].
- [44] «Introduction à la réplication de bases de données,» *Linux Magazine France HS* , n° %118, Février 2014.

Thèse et mémoire

- [3] R. NOUBIAP NOMSI, «Implémentation d'une solution IDS/IPS,» Institut Supérieur du Sahel, Maroua, 2015.
- [13] C. Patrice Paterne, «CONCEPTION D'UNE SOLUTION DE CLOUD COMPUTING PRIVE : APPLICATION AUX SERVICES IAAS,» ABOMEY, 2012.

Articles du web (webographie)

- [2] «Chiffres facebook 2015,» 2015. [En ligne]. Available: www.blogdumoderateur.com/chiffres-facebook/. [Accès le 10 mai 2016].
- [4] MATRIX. [En ligne]. Available: www.matrixtelecoms.com. [Accès le 14 avril 2016].
- [5] wikipédia, «Infrastructure as a service,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Infrastructure_as_a_service. [Accès le 14 mai 2016].
- [6] S. MAAREF, «Cloud Computing en Afrique situation et perspectives,» avril, 2012.
- [9] M. JUGANARU-MATHIEU, Cloud Computing, St Etienne, 2014-2015.
- [12] I. Center, «Infrastructure de Cloud privée en tant que service,» [En ligne]. Available: <http://www.intel.fr/content/www/fr/fr/it-management/intel-it-managers.html>. [Accès le 20 avril 2016].

- [14] Ubuntu-fr, «virtualisation,» [En ligne]. Available: <https://doc.ubuntu-fr.org/virtualisation>. [Accès le 14 mai 2016].
- [15] F. Santy, «La Virtualisation,» Université libre de Bruxelles, Bruxelles, 2009-2010.
- [16] Linux.org. [En ligne]. Available: <http://linuxfr.org/news/la-folie-docker>. [Accès le 14 mai 2016].
- [17] iTPro.fr, «La virtualisation complète,» [En ligne]. Available: file:///D:/M%C3%A9moire/M%C3%A9moire/Xen/La%20virtualisation%20compl%C3%A8te%20_%20iTPro.fr.html. [Accès le 15 mai 2016].
- [18] P. CODO, «Chapitre 2 : Les technologies du cloud computing,» [En ligne]. Available: <http://www.institut-numerique.org/chapitre-2-les-technologies-du-cloud-computing-51c0279ca9f6f>. [Accès le 14 mai 2016].
- [19] VMWare, «Synthèse virtualisation serveur,» [En ligne]. Available: <http://store.vmware.com/store/vmwde/home>. [Accès le 13 juin 2016].
- [20] techweez.com, «What Drives Azure? Virtualized Networks According to Microsoft,» [En ligne]. Available: <http://www.techweez.com/2013/11/09/azure-microsoft-cloud-platform-undergone-multiple-changes-past-3-years/>. [Accès le 14 juin 2016].
- [21] highscalability.com. [En ligne]. Available: <http://highscalability.com/blog/2011/9/7/what-google-app-engine-price-changes-say-about-the-future-of.html>. [Accès le 14 juin 2016].
- [22] VMware, «Attains VMware IaaS Powered Status in the VMware vCloud Air Network,» [En ligne]. Available: <https://www.blueshift.net/attains-vmware-iaas-powered-status-in-the-vmware-vcloud-air-network/>. [Accès le 14 juin 2016].
- [23] ArcsysSoftware. [En ligne]. Available: <http://www.arcsys-software.fr/partenaires/partenaires-technologiques/>. [Accès le 14 juin 2016].
- [24] clker.com, «Eucalyptus Logo Awh image,» [En ligne]. Available: <http://www.clker.com/clipart-122025.html>. [Accès le 14 juin 2016].

- [25] OpenStack, «The OpenStack Logo,» [En ligne]. Available: <https://www.openstack.org/brand/openstack-logo/logo-download/>. [Accès le 14 juin 2016].
- [26] OpenNebula, «Virtualisation with Ceph and OpenNebula,» [En ligne]. Available: <http://mricharl.github.io/opennebula-talk/#/start>. [Accès le 14 juin 2016].
- [27] pinterest, «Pinterest • Le catalogue d'idées, Explorez Architecture Arm, Open Source et plus encore !,» [En ligne]. Available: <https://www.pinterest.com/pin/183943966000494633/>. [Accès le 14 juin 2016].
- [28] P. Scoffoni, «Xen Cloud Platform, la solution open source de cloud computing de Xen.org,» 29 août 2009. [En ligne]. Available: <https://philippe.scoffoni.net/category/cloud-computing/>. [Accès le 14 juin 2016].
- [37] «WikiMeca,» [En ligne]. Available: http://wikimeca.org/index.php?title=Exemple_trait%C3%A9:_Store_SOMFY. [Accès le 22 08 2016].
- [38] ac-grenoble.fr, «PRESENTATION DE L'ANALYSE FONCTIONNELLE».
- [39] L. EPPE, «Citrix XenServer - Installation, configuration, mises à jour et utilisation,» 30 décembre 2014. [En ligne]. Available: <http://www.informatiweb-pro.net/virtualisation/1-citrix/10--citrix-xenserver-installation-configuration-mises-a-jour-et-utilisation.html>. [Accès le 11 mai 2016].
- [40] xenserver.org, «XENCENTER DEVELOPMENT,» [En ligne]. Available: <http://xenserver.org/partners/developing-products-for-xenserver/21-xencenter-development/88-xc-dev-home.html>. [Accès le 08 Août 2016].
- [41] L. EPPE, «Citrix XenDesktop - Installation, configuration et mise à disposition de bureaux via MCS,» 13 mai 2015. [En ligne]. Available: <http://www.informatiweb-pro.net/virtualisation/1-citrix/13--citrix-xendesktop-installation-configuration-et-mise-a-disposition-de-bureaux-via-mcs.html>. [Accès le 06 août 2016].

GLOSSAIRE

Applications distribuées : c'est une application qui fonctionne dans un système distribué et qui permet de distribuer les tâches entre les différents nœuds de ce système.

Citrix : est une entreprise multinationale américaine qui propose des produits de collaboration, de virtualisation et de mise en réseau pour faciliter le travail mobile et l'adoption des services cloud. (www.citrix.com)

Cluster : grappe en français. Architecture de groupes d'ordinateurs, utilisée pour former de gros serveurs. Chaque machine est un nœud du cluster, l'ensemble est considéré comme une seule et unique machine. Utilisée pour le décisionnel, le transactionnel et le datawarehouse. (Source jargon Informatique).

Emulation : consiste à substituer un élément de matériel informatique tel un terminal informatique, un ordinateur ou une console de jeux par un logiciel. La définition du terme *émuler* est « chercher à imiter ». Il faut voir dans l'émulation une imitation du comportement physique d'un matériel par un logiciel, et ne pas la confondre avec la simulation, laquelle vise à imiter un modèle abstrait.

Frame buffer : Partie de la mémoire gérée comme étant une représentation directe des pixels à l'écran. Écrire dans cette mémoire revient à modifier l'affichage. (Source jargon Informatique).

KVM : commutateur permettant de partager un ensemble clavier, écran, souris, entre plusieurs machines, sans devoir redémarrer celles-ci à chaque changement. Surtout utilisé dans les salles machine. (Source jargon Informatique).

Le nuage (anglais *cloud*) : est un ensemble de matériel, de raccordements réseau et de logiciels qui fournit des services sophistiqués que les individus et les collectivités peuvent exploiter à volonté depuis n'importe où dans le monde.

Memcached : est un système d'usage général servant à gérer la mémoire cache distribuée. Il est souvent utilisé pour augmenter la vitesse de réponse des sites web créés à partir de bases de données. Il gère les données et les objets en RAM de façon à réduire le nombre de fois qu'une même donnée stockée dans un périphérique externe est lue. Il tourne sous Unix, Windows et MacOS et est distribué selon les termes d'une licence libre dite permissive.

Middleware : (ou logiciel médiateur en français ou encore intergiciel), littéralement « élément du milieu », l'ensemble des couches réseau et services logiciel,

qui permettent le dialogue entre différents composants d'une application répartie. Ce dialogue se base sur un protocole applicatif commun, défini par l'API du middleware. Le **Gartner Group** définit le middleware comme une interface de communication universelle entre processus. Il représente véritablement la clef de voûte de toute application client-serveur.

Orchestration : fonctionnalité qui permet d'optimiser l'efficacité et la réactivité du Datacenter.

Parser : se dit des fichiers sur lequel on veut faire de l'analyse syntaxique. Il peut être un programme informatique.

Pay-per-use : paiement à l'utilisation

Pointer-et-cliquer (de l'anglais « *point-and-click* ») : est une des actions qu'un utilisateur peut effectuer sur une interface utilisateur graphique avec pointeur. L'utilisateur déplace le pointeur avec un dispositif de pointage (souris, manette de jeu, pavé tactile) sur un emplacement particulier de l'écran d'ordinateur (pointer), puis il appuie sur un bouton du dispositif (ou équivalent) pour déclencher l'action (cliquer).

SOA : L'architecture orientée services est une forme d'architecture de médiation qui est un modèle d'interaction applicative qui met en œuvre des services.

Sticky : se dit d'un attribut de fichier, qui permet d'empêcher l'effacement d'un fichier temporaire et d'accélérer l'accès au fichier.

Temps réel : ce terme signifie que les services peuvent être invoqués à tout moment avec une réponse instantanée de la part des applications.

Versioning : la gestion de versions (en anglais *version control* ou *revision control*) consiste à maintenir l'ensemble des versions d'un ou plusieurs fichiers (généralement en texte). Essentiellement utilisée dans le domaine de la création de logiciels, elle concerne surtout la gestion des codes source.

ANNEXE 1 : Explication du fonctionnement d'Eucalyptus & OpenNebula

2.1 Fonctionnement d'Eucalyptus

Eucalyptus est un outil open source issu d'un projet de recherche de l'université de Californie. Il est développé en C, Java, Python et est disponible sous deux licences. Une licence GPL gratuite supportant les hyperviseurs Xen et KVM* et une licence commerciale offrant des fonctionnalités avancées telles que le support de VMware. Il permet de construire aussi bien les solutions privées du cloud Computing que les solutions publiques. Son grand avantage est qu'il est intégré dans les distributions Ubuntu et Debian. Eucalyptus offre des interfaces compatibles avec les services EC2 d'Amazon. Ce qui lui confère la possibilité d'être employé pour les solutions hybrides de cloud Computing. L'architecture d'Eucalyptus est constituée de quatre composants principaux (Nurmi et al, 2008 ; Alrwais, 2011).

- Le contrôleur de nœud (Node Controller NC) : contrôle l'exécution, et l'arrêt des machines virtuelles présentes sur le nœud où il est exécuté.
- Le contrôleur de cluster (cluster Controller CC) : collecte les informations sur les différents nœuds d'un cluster et planifie l'exécution des machines virtuelles sur chaque nœud.
- Le contrôleur de stockage (Warlus) : c'est le composant qui gère l'accès au service de stockage. Il est souvent intégré au contrôleur du cloud (CLC).
- Le contrôleur de cloud (CLC): C'est le point d'entrée (Front end) des utilisateurs et administrateurs du système. Il collecte des informations sur les nœuds et planifie leur exécution au travers des contrôleurs de clusters (CCs). Il expose les services du cloud à travers une application Web mais également à travers des interfaces compatibles EC2.

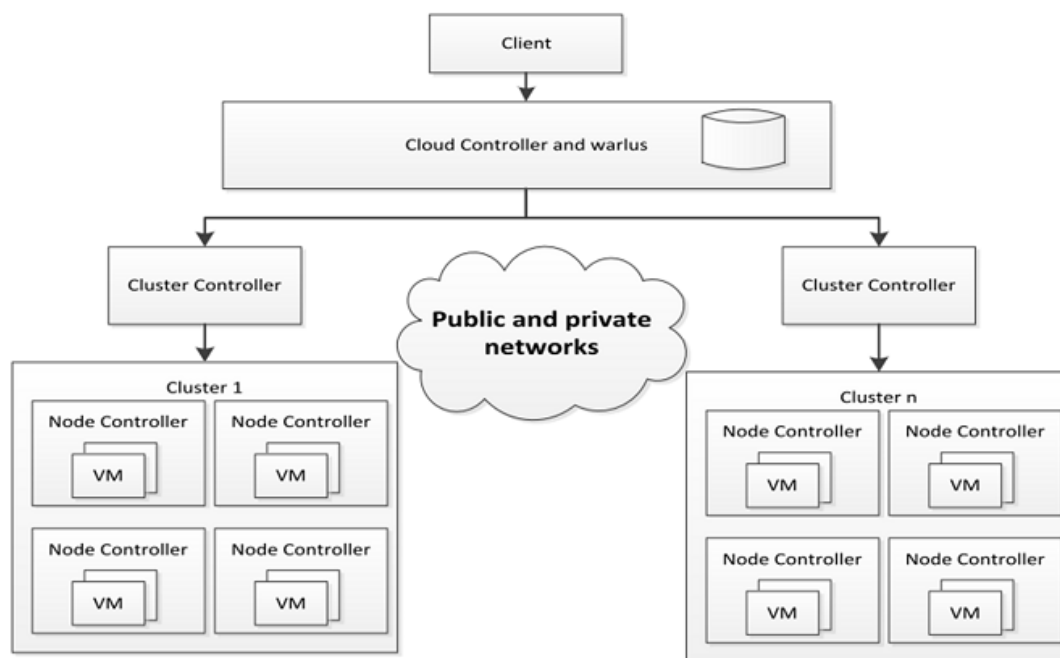


Figure A.1: Architecture d'Eucalyptus (Alrwais, 2011; Naing, 2012).

Fonctionnement

Lorsqu'il est installé, le système Eucalyptus inclut sur le nœud frontal un utilitaire appelé euca2ools pour fonctionner. Ce dernier fournit les éléments nécessaires pour l'authentification et l'interaction d'un utilisateur avec le système. Les étapes de déroulement d'un scénario de création de machine virtuelle se présentent à la figure:

1. L'utilisateur fait une requête de machine virtuelle à travers le nœud central à l'aide de l'outil euca2ools.
2. Un modèle d'image virtuelle est envoyé sur le nœud d'exécution.
3. L'image est ensuite ajustée à une taille adéquate selon les informations fournies.
4. Le nœud d'exécution crée un pont virtuel afin de fournir une interface réseau virtuelle avec une adresse MAC à la machine virtuelle.
5. Une requête DHCP est effectuée à l'endroit du nœud frontal pour obtenir une adresse IP pour la machine.
6. La machine est créée et démarrée dans le gestionnaire des machines virtuelles.
7. L'utilisateur peut ensuite se connecter via SSH, ou autres protocoles, à la machine virtuelle.

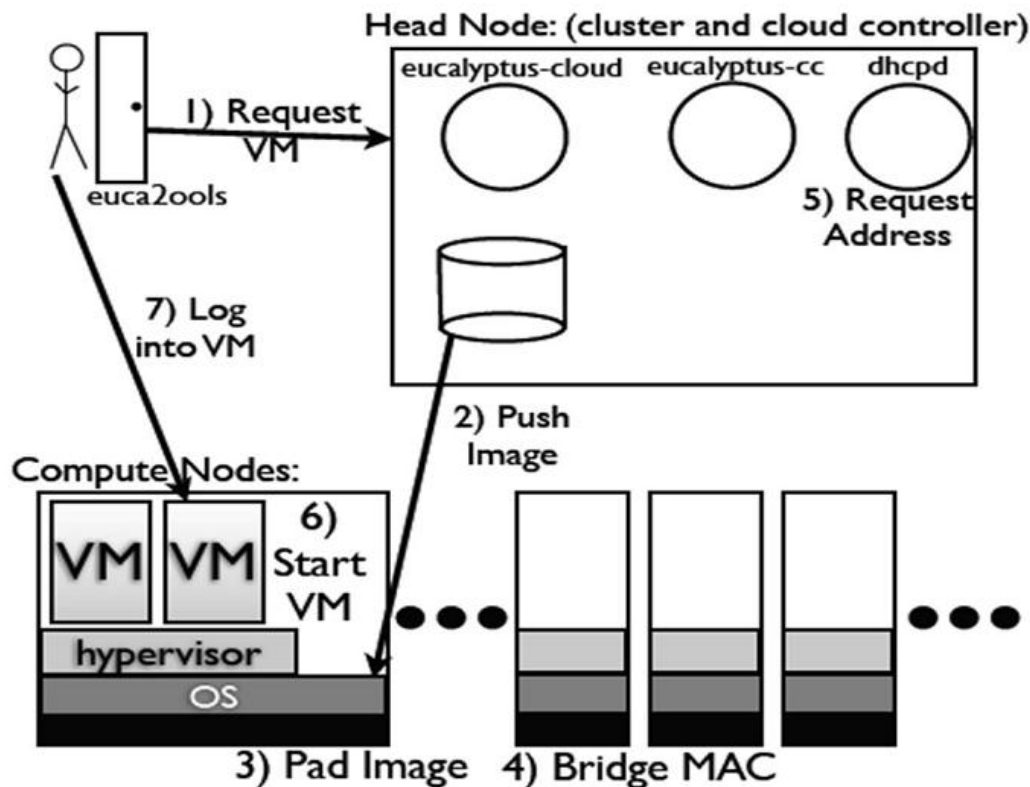


Figure A.2 : Utilisation de machine virtuelle avec Eucalyptus (Sempolinski et Thain, 2010).

2.2 Fonctionnement d'OpenNebula

OpenNebula se base sur un compte utilisateur *oneadmin* créé et configuré sur tous les hôtes du réseau pour fonctionner. La configuration de cet utilisateur concerne l'attribution des privilèges au compte *oneadmin* pour certaines commandes telles que les commandes de gestion des machines virtuelles qui ne sont accessibles par défaut qu'au compte **root** des systèmes GNU/Linux employés. Les requêtes envoyées par le Cloud Controller, représenté par le processus *oned* (*oned process*), localisé sur le nœud frontal d'OpenNebula à l'endroit des différents nœuds d'exécution sont traitées et les commandes nécessaires à la réalisation des opérations sont effectuées en employant le compte *oneadmin* de l'hôte indexé. OpenNebula emploie par défaut le système de partage de fichiers NFS (Network File System) pour l'accès aux différents fichiers des machines virtuelles localisés sur le nœud frontal (front-end) par les différents nœuds d'exécution.

La solution de cloud privé d'OpenNebula offre également la possibilité de migrer les machines virtuelles d'un hôte à un autre en employant le protocole de transfert SSH (Secure Shell) à travers les commandes *scp* (secure copy). Une exigence de l'emploi de ce protocole concerne l'utilisation des techniques d'échange de clés publiques RSA afin de contourner les processus d'authentications nécessaires à chaque déploiement de

machines virtuelles sur un nœud. Ainsi, les comptes oneadmin disponibles sur les hôtes, et également sur le front-end doivent être configurés pour des accès directs, par SSH, sans authentification par mots de passe.

Le lancement d'une machine virtuelle avec OpenNebula par un utilisateur se déroule en huit (08) étapes (voir figure) :

1. L'utilisateur accède au nœud frontal par SSH.
2. Il utilise les commandes onevm (commandes du processus oned d'OpenNebula sur le front-end) pour effectuer les requêtes d'utilisation d'une machine virtuelle.
3. Un modèle de la machine virtuelle est copié et ajusté à une taille précise dans le répertoire NFS du nœud frontal.
4. Le processus oned du nœud frontal utilise SSH pour se connecter à un nœud d'exécution.
5. Le nœud d'exécution crée un pont virtuel afin de fournir une interface réseau virtuelle avec une adresse MAC à la machine virtuelle.
6. Les fichiers nécessaires sont ensuite envoyés sur le nœud d'exécution par NFS ou SSH.
7. La machine virtuelle est démarrée par le gestionnaire des machines virtuelles (VMM).
8. L'utilisateur peut ensuite y accéder par SSH.

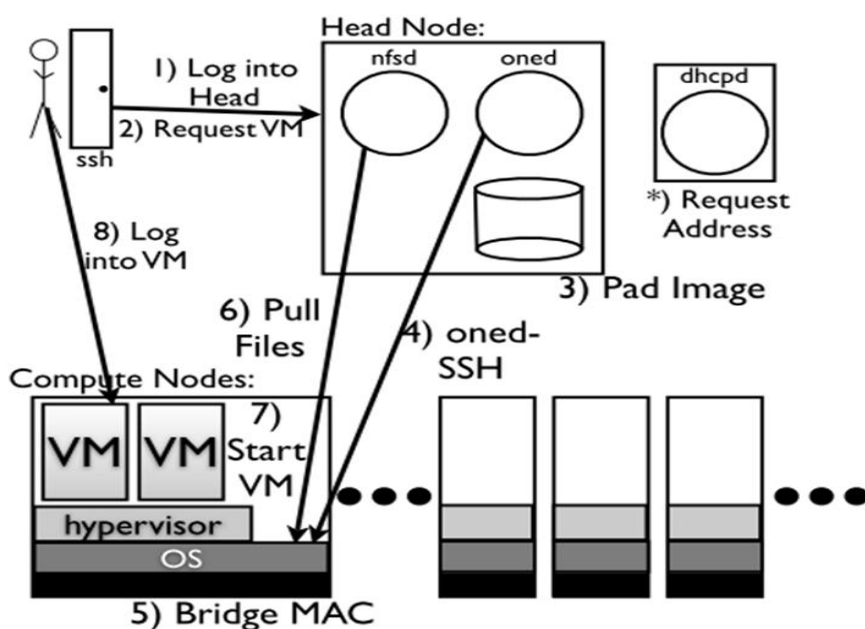


Figure A.1 : lancement d'une machine virtuelle avec OpenNebula [13]

ANNEXE 3 : Documentation de Xen orchestra

Afin de replacer un peu le contexte, sachez pour commencer que XenOrchestra est un produit français! En effet, ce projet est maintenu par la société **Vates**, qui est une entreprise française spécialisée dans les solutions Open Source. Cette société travaille sur le projet **Xen Orchestra** en collaboration avec le projet Xen, mais pas seulement car elle propose d'autres solutions. Cette entreprise ne vend pas à proprement parler de solutions logicielles, mais le support, la formation et le consulting qui les accompagnent.

Dans la logique, XenServer est là pour gérer l'hébergement brut de vos machines virtuelles. Mais par défaut, le seul et unique moyen (hors la CLI XE) d'administrer vos machines virtuelles hébergées par vos XenServer, c'est d'utiliser ce que Citrix appelle le **XenCenter**. Et là, en ce qui me concerne, c'est une grosse FAILLE !

Et oui, cela oblige l'admin à utiliser Windows car XenCenter ne fonctionnant que sous OS Windows. Pour information, le projet **OpenXenManager** tente de porter XenCenter sur les distributions GNU/Linux mais ce n'est pas encore stable et cela ne propose «pour le moment» que «quelques» fonctionnalités de bases proposées par XenCenter.

XenOrchestra qui, comme vous l'aurez compris, est une **Web GUI** (ou interface Web) pour XenServer et XAPI.

Il s'agit donc d'une interface Web plutôt sympathique d'un point de vue design et ergonomie qui permet la gestion simplifiée de ses machines virtuelles (ou domU) réparties sur un ensemble d'hyperviseur (ou dom0).

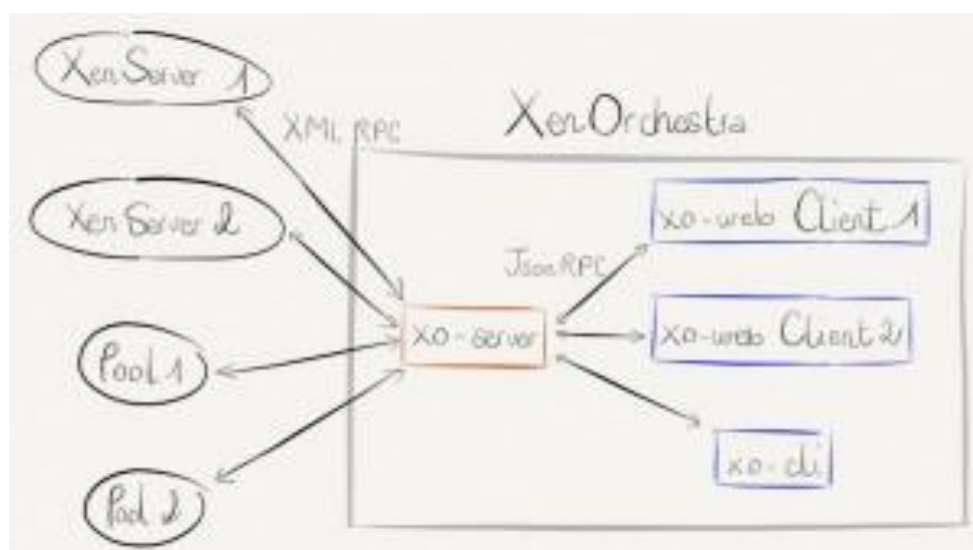


Illustration A.2 : Logique de l'architecture de XenOrchestra [42]

C'est donc cette interface Web qui va à terme remplacer l'utilisation de XenCenter et nous permettre de nous séparer de Windows. Attention toutefois, si vous n'êtes pas totalement à l'aise avec la CLI de XenServer, XenOrchestra ne permet pas un remplacement total de XenCenter. En effet, certaines fonctionnalités « cruciales » telle que l'import ou l'export des machines virtuelles ne sont pas (encore ?) intégrées. (Voir la référence N°2 pour plus d'informations).

ANNEXE 4 : Mise en place de la réplication DRBD entre 2 XenServer

Configuration LVM

Pour notre démo, le volume raid est /dev/sdb, 500GB. A adapter à votre cas

- Commande à lancer sur les 2 XenServer

```
ssh root@"ip xenservers"  
modifier les filtres LVM  
vi /etc/lvm/lvm.conf  
commenter la ligne  
#filter = [ "r|/dev/xvd.|", "r|/dev/VG_Xen.*/*|"]  
remplacer par  
filter = ["a|/dev/drbd*|", "a|sda.*|", "r|.*/"]  
Redémarrer le serveur  
pvcreate /dev/sdb  
vgcreate xenserver /dev/sdb  
Avant de créer les LVM, il faut prendre en compte que le volume groupe ne fait pas  
500 Go:  
La commande  
vgs  
affiche  
xenserver 1 0 0 wz--n- 437G 437G  
créer les volumes LVM en conséquence  
lvcreate -n drbd1 -L 200G xenserver  
lvcreate -n drbd2 -L 200G xenserver
```

Configuration du DRBD

Configurer le gestionnaire de paquets yum

```
vi /etc/yum.repos.d/CentOS-Base.repo  
dans la section [base] changer enabled=0  
enabled=1  
vi /etc/yum.repos.d/Citrix.repo  
changer enabled=1  
enabled=0  
cd /etc/pki/rpm-gpg/  
wget --no-check-certificate https://fedoraproject.org/static/217521F6.txt  
mv 217521F6.txt RPM-GPG-KEY-EPEL  
cd /root  
wget "http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm"  
rpm -i epel-release-5-4.noarch.rpm
```

Installer le DRBD

cd /root

wget http://srvinstallation.tranquil.it/logiciels/drbd/drbd-utils-8.4.3-2.x86_64.rpm

rpm -i drbd-utils-8.4.3-2.x86_64.rpm

modprobe drbd

Configuration interfaces réseau drbd

Nous utiliserons eth1 sur chaque Xen pour la réplication DRBD.

Ces interfaces devront être seulement utilisées pour le trafic DRBD.

sur l'un des XenServer

xe pif-list device=eth1

Un uuid est retourné pour chaque eth1 des 2 Xenservers

uuid (RO) : 082915b1-af96-74cd-f955-2517d3f794b9

device (RO): eth1

currently-attached (RO): true

VLAN (RO): -1

network-uuid (RO): c3c23dfe-1a7a-d121-adc0-ca4046917607

uuid (RO) : 67077c00-1a56-799d-13d3-a067a6fb3cdb

device (RO): eth1

currently-attached (RO): true

VLAN (RO): -1

network-uuid (RO): c3c23dfe-1a7a-d121-adc0-ca4046917607

Il faut supprimer les définitions de ces PIF et le NETWORK

xe pif-unplug uuid=082915b1-af96-74cd-f955-2517d3f794b9

xe pif-unplug uuid=67077c00-1a56-799d-13d3-a067a6fb3cdb

xe pif-forget uuid=082915b1-af96-74cd-f955-2517d3f794b9

xe pif-forget uuid=67077c00-1a56-799d-13d3-a067a6fb3cdb

xe network-destroy uuid=c3c23dfe-1a7a-d121-adc0-ca4046917607

- A faire sur srvxen1

vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1

XENMANAGED=no

ONBOOT=yes

BOOTPROTO=static

IPADDR=10.0.0.1

NETMASK=255.255.255.0

MTU=9000

Activer l'interface réseau

ifup eth1

Relancer xapi pour vérifier le tout

/etc/init.d/xapi restart

- A faire sur srvxen2

```

vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
XENMANAGED=no
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.0.0.2
NETMASK=255.255.255.0
MTU=9000
Activer l'interface réseau
ifup eth1
Relancer xapi pour vérifier le tout
/etc/init.d/xapi restart

```

Configuration le DRBD

- A faire sur les 2 XenServer

```

vi /etc/drbd.d/drbd1.res
resource drbd1 {
    protocol B;

    on srvxen1.test.lan {
        device /dev/drbd1;
        disk /dev/xenserver/drbd1;
        address 10.0.0.1:7090;
        meta-disk internal;
    }
    on srvxen2.test.lan {
        device /dev/drbd1;
        disk /dev/xenserver/drbd1;
        address 10.0.0.2:7090;
        meta-disk internal;
    }
}
vi /etc/drbd.d/drbd2.res
resource drbd2 {
    protocol B;

    on srvxen1.test.lan {
        device /dev/drbd2;
        disk /dev/xenserver/drbd2;
        address 10.0.0.1:7091;
        meta-disk internal;
    }
    on srvxen2.test.lan {
        device /dev/drbd2;
        disk /dev/xenserver/drbd2;
        address 10.0.0.2:7091;
        meta-disk internal;
    }
}

```

```

}
vi global_common.conf
global {
    usage-count no;
}

common {
    disk{
        no-disk-barrier;
        no-disk-flushes;
    }

    syncer {
        rate 800M;
        al-extents 3833;
    }
    startup {
        degr-wfc-timeout 15;
    }
    net {
    }
}
}

```

Configurer IPTABLES

- A faire sur les 2 xenserver

vi /etc/sysconfig/iptables

Ajouter

#drbd

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 7090 -j ACCEPT

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 7091 -j ACCEPT

/etc/init.d/iptables restart

Monter le DRBD et lancer les synchros

!!! Attention, à ne faire qu'au premier lancement des DRBD !!!
sur srvxen1

drbdadm create-md drbd1

drbdadm up drbd1

!!! Attention, ne lancer la commande ci-dessous que sur srvxen1!!!

drbdadm -- --overwrite-data-of-peer primary drbd1

drbdadm create-md drbd2

drbdadm up drbd2

sur srvxen2

drbdadm create-md drbd2

drbdadm up drbd2

!!! Attention, ne lancer la commande ci-dessous que sur srvxen2!!!

drbdadm -- --overwrite-data-of-peer primary drbd2

drbdadm create-md drbd1

drbdadm up drbd1

La commande

drbd-overview

Connecter les volumes répliqués DRBD au pool de serveurs Xen

Récupérer les uuid des host

xe host-list

uuid (RO) : 63b226c8-fa0d-4435-b8e3-8b44024d2e90

name-label (RW): srvxen2.test.lan

name-description (RW): Default install of XenServer

uuid (RO) : 6bcfc4fd-f31d-4720-ba56-c0dfa661135b

name-label (RW): srvxen1.test.lan

name-description (RW): Default install of XenServer

Créer un Storage Repository sur le volume drbd1

- sur srvxen1

**xe sr-create content-type=user device-config:device=/dev/drbd1 host-uuid="uuid
srvxen1" name-label=drbd1 shared=false type=lvm**

Créer un Storage Repository sur le volume drbd2

- sur srvxen2

**xe sr-create content-type=user device-config:device=/dev/drbd2 host-uuid="uuid
srvxen2" name-label=drbd2 shared=false type=lvm**

Les Storage Repositories dans XenCenter

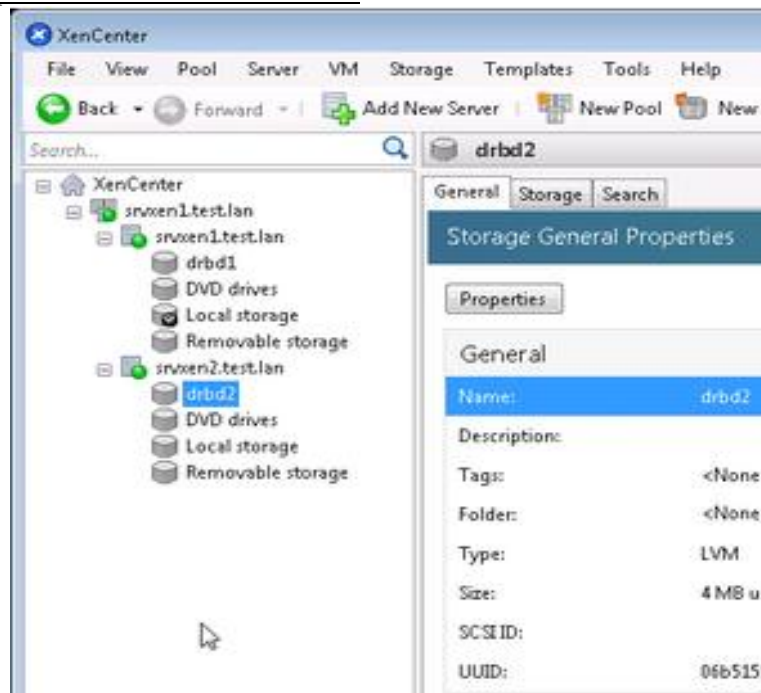


Illustration A.3 : volumes répliqués DRBD au pool de serveurs Xen

Comme on peut le voir sur cette illustration les deux volumes s'affichent dans XenCenter.

Migrer le master

passer srvxen2 en master (en ssh)

xe pool-emergency-transition-to-master

Après le passage de srvxen2 en master, on reconfigure le XenCenter pour qu'il se connecte à srvxen2.

Après le passage de srvxen2 en master, on reconfigure le XenCenter pour qu'il se connecte à srvxen2.

Transférer le DRBD sur srvxen2

- sur srvxen2

drbdadm primary drbd1

vgscan

vgchange -ay

La commande drbd-overview lancer sur srvxen2 montre que la ressource drbd1 est en primary et qu'il possède un "VG_XenStorage...".

[root@srvxen2 ~]# drbd-overview

1:drbd1/0 WFCConnection Primary/Unknown UpToDate/DUnknown B r----- lvm-pv: VG_XenStorage-9856988f-e574-551d-91df-e2d5cf2668f9 1853.37G 18.03G

2:drbd2/0 Connected Primary/Secondary UpToDate/UpToDate B r----- lvm-pv: VG_XenStorage-54f3ed4d-a211-bb51-13d1-03bba2127ed6 1853.37G 0.00G

Créer pbd sur srvxen2

- Récupérer l'uuid du SR drbd1

```
[root@srvxen2 ~]# xe sr-list name-label=drbd1
```

uuid (RO) : 9856988f-e574-551d-91df-e2d5cf2668f9

name-label (RW): drbd1

name-description (RW):

host (RO): <not in database>

type (RO): lvm

content-type (RO): user

- Récupérer l'uuid du XenServer srvxen2

```
[root@srvxen2 ~]# xe host-list name-label=srvxen2.test.lan
```

uuid (RO) : 5633c39a-bcea-47c5-8a55-49e4a40f013e

name-label (RW): srvxen2.test.lan

name-description (RW): Default install of XenServer

- créer le pbd sur srvxen2

```
[root@srvxen2 ~]# xe pbd-create sr-uuid=9856988f-e574-551d-91df-e2d5cf2668f9
```

```
host-uuid=5633c39a-bcea-47c5-8a55-49e4a40f013e device-config:device=/dev/drbd1
```

retourne un uuid exemple:

8e7d5107-10f6-fe2a-9f7a-47be24bee5ea

- plug du pbd avec l'uuid créer

```
xe pbd-plug uuid=8e7d5107-10f6-fe2a-9f7a-47be24bee5ea
```

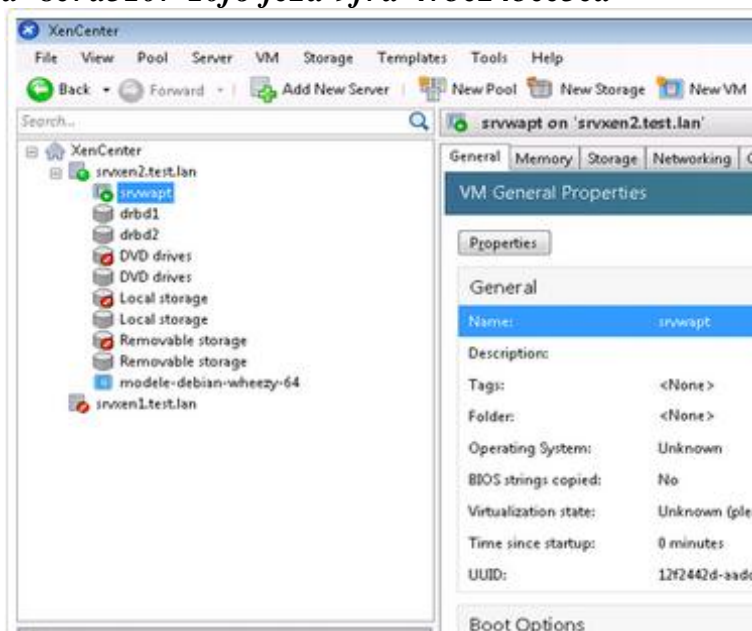


Illustration A.4: migration du master

Le Storage Repository est déplacé sur srvxen2 et la VM srvwapt est démarrée sur ce SR.