

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



*Mémoire de fin de cycle
En vue d'obtention du diplôme
de Master en Informatique*

Spécilité: Administraration et Sécurité des Réseaux Informatiques

Thème

Etude et Mise en oeuvre d'une solution Cloud Computing

Réalisé par:

Mlle KHENOUS Hakima

Mlle MEZIANE Kahina

Soutenu devant le jury composé de :

Président	M ^r ACHROUFENE Achour	U. A/Mira Béjaïa.
Examineur	M ^{me} GHIDOUCHE Kahina	U. A/Mira Béjaïa.
Examineur	M ^{me} KHALED Hayette	U. A/Mira Béjaïa.
Encadreur	M ^r SIDER Abderrahmane	U. A/Mira Béjaïa.

Université de Béjaia 2013

** * ** Remerciements * * ***

*M*erci à Dieu de nous avoir donné la force et le courage de tenir jusqu'à la fin de ce travail.

*E*n tout premier lieu, nous remercions grandement Monsieur "SIDER Abderrahmane" d'avoir accepté de nous encadrer pour pouvoir réaliser notre mémoire. Nous le remercions aussi pour la confiance qu'il nous a accordée et surtout pour ses encouragements qui nous ont accompagnés durant tout le parcours du travail. Nous n'oublierons pas ses recommandations qui nous ont beaucoup aidés pour finaliser ce mémoire.

*N*ous tenons également à remercier les membres de jury M^r ACHROUFENE Achour, M^{me} GHIDOUCHE Kahina et M^{me} KHALED Hayette qui nous font honneur en acceptant de juger notre travail.

*N*otre reconnaissance se porte également à nos familles qui ont su nous apporter leur soutiens durant toutes ces longues années d'étude.

*E*nfin, nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de notre projet.

*** *Dédicaces* ***

A mes très chers parents qui n'ont jamais cessé de me soutenir tout au long de mon parcours d'étude.

A la mémoire de mes chères grand parents.

A mes chers frères et soeur ainsi à mes belles soeurs qui m'ont encourgés durant la réalisation de ce projet .

A mon fillancé qui m'a vraiment soutenu.

A mes adorables petits neuveux hocine, mohamed, ryna, manal, imane, lilya .

A mes ancles, mes tantes, cousins et cousines.

A ma chère amie hakima avec laquelle j'ai eu le plaisir de travailler.

A mes amis (es): Sabrina, Sara, Rabia, Aida, Saida et hassiba.

A tous ceux dont les noms ny figurent pas pour une raison ou une autre trouve l'expression de ma profonde gratitude.

Kahina

*** *Dédicaces* ***

A mes très chers parents qui m'ont jamais cessé de me soutenir, et que j'aime énormément.

A mes grands parents.

A mon adorable soeur Souad et à mon beau frère.

A mes frères toufik et karim qui m'ont encourgés durant toute la réalisation de ce projet.

A mes oncles, tantes, cousins et cousines.

A ma chère amie du parcours Kahina.

A mes copines: Sabrina, Sarra, Rabia, Aida.

Et A tout(e)s mes ami(e)s.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

Hakima

Liste des abréviations

AMI	Amazon Machine Image
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ASP	Application Service Provider
AWS	Amazon Web Service
BSD	Berkeley Software Distribution license
DHCP	Dynamic Host Configuration Protocol
EBS	Elastic BlockStore
EC2	Elastic Compute Cloud
HPC	High Performance Computer
GNU GPL	GNU General Public License
HTTP	Hyper Text Transfer Protocol
IAAS	Infrastructure As AService
IDS	Inrusion Detection System
IM	Information Manager
IP	Internet Protocol
ISCSI	Internet Small Computer System Interface
IT	Information Technology
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LXC	Linux Containers

LVM	L ogical V olume M anager
MAC	M ulti A ccess C omputer
MySQL	M y S tructured Q uery L anguage
NAT	N etwork A ddress T ranslation
NAS	N etwork A ttached S torage
NASA	N ational A eronautics and S pace A dministration
NFS	N etwork F ile S ystem
PAAS	P late-forme A s A S ervice
PHP	P ersonal H ome P age
POSIX	P ortable O perating S ystem I nterface
QEMU	Q uick E MUlator
REST	R Epresentational S tate T ransfer
S3	S imple S torage S ervice
SAAS	S oftware A s A S ervice
SAN	S torage A rea N etwork
SLA	S ervice L evel A greement
SOAP	S imple O bject A ccess P rotocol
SPOF	S ingle P oints O f F ailure
SSH	S ecure S hell
TCP	T ransmission C ontrol P rotocol
TM	T ransfert M anager
UML	U ser M ode L inux
URL	U niform R esource L ocator
VLAN	V irtual L ocal A rea N etwork
VM	V irtual M achine
VMDK	V irtual M achine D isk
VMM	V irtual M achine M onitor
VNC	V irtual N etwork C omputing
VPN	V irtual P rivate N etwork
WAN	W ide A rea N etwork
WSDL	W eb S ervices D escription L anguage
XML	e X tenssible M arkup L anguage

TABLE DES MATIÈRES

Table des Matières	iii
Liste des figures	vii
Liste des tableaux	ix
Introduction Générale	1
1 Etat de l'art sur la virtualisation et de Cloud Computing	3
Introduction	3
1.1 Définition	3
1.2 La virtualisation	4
1.2.1 Définition	4
1.2.2 Les Avantages et les inconvénients de la virtualisation	4
1.2.3 Les technologies de virtualisation	5
1.2.4 Solutions de virtualisation	10
1.3 La relation entre le cloud computing et la virtualisation	11
1.4 Historique du Cloud Computing	11
1.5 Les différents services du Cloud Computing	12
1.5.1 Saas (Software as a Service)	12
1.5.2 Paas (Plateform as a Service)	14
1.5.3 Iaas (Infrastructure as a Service)	14
1.6 Types de Cloud	16
1.6.1 Le Cloud privé(interne)	16

1.6.2	Le Cloud public (externe)	17
1.6.3	Le Cloud hybride (interne et externe)	17
1.6.4	Le Cloud communautaire:	17
1.7	Avantages et Inconvénients du Cloud Computing	18
1.7.1	Avantages du Cloud Computing	18
1.7.2	Inconvénients du Cloud Computing	19
1.8	Les principaux acteurs du Cloud Computing	19
1.8.1	Le Cloud d’Amazon	19
1.8.2	Le Cloud de Google	20
1.8.3	Le Cloud de Microsoft	20
1.9	Le Choix entre le Cloud privé et le Cloud public	21
1.9.1	Inconvénients du cloud computing public	21
1.9.2	Le Cloud Computing privé	21
1.10	Problématique	22
2	Les solutions Cloud Computing open source	24
	Introduction	24
2.1	La solution Eucalyptus	24
2.1.1	présentation	24
2.1.2	Architecture Eucalyptus	25
2.1.3	Caractéristiques	28
2.1.4	Gestion des images	29
2.1.5	Réseau et adressage	30
2.1.6	Sécurité	32
2.1.7	Stockage	32
2.1.8	Les interfaces fournies	32
2.2	La solution OpenNebula	33
2.2.1	Présentation	33
2.2.2	Architecture OpenNebula	33
2.2.3	Gestion des images	35
2.2.4	Réseau et adressage	36
2.2.5	Sécurité	37
2.2.6	Stockage	37

2.2.7	Les interfaces fournies	38
2.3	La solution OpenStack	38
2.3.1	Présentation	38
2.3.2	Architecture OpenStack	39
2.3.3	La relation entre les services d'OpenStack	41
2.3.4	Gestion des images	42
2.3.5	Réseau et adressage	43
2.3.6	Sécurité	43
2.3.7	Stockage	44
3	La comparaison et le choix entre les solutions Cloud Computing	45
	Introduction	45
3.1	Tableau comparatif des solutions Cloud Computing (IaaS)	46
3.2	Performance et évolutivité	50
3.3	Recommandation matérielle	52
3.4	Avantages et inconvénients	52
3.4.1	Avantages	52
3.4.2	Inconvénients	52
4	Installation et administration de la solution OpenStack	54
	Introduction	54
4.1	Installation d'OpenStack	54
4.1.1	Outils à utiliser pour l'installation d'openstack	55
4.2	Installation des paquets de base et préparation du système	56
4.2.1	La mise à jour du système	56
4.2.2	Installation des bridges	57
4.2.3	Configuration réseau	57
4.2.4	Ajout du repository Ubuntu Cloud Archive	58
4.2.5	Installation et Configuration de MySQL	59
4.2.6	Installation et configuration du serveur NTP	59
4.2.7	Installation de RabbitMQ	60
4.3	Installation et configuration des composants OpenStack	60
4.3.1	Installation et configuration de Keystone	61

4.3.2	Installation et configuration de Glance	73
4.3.3	Installation et configuration de Nova	75
4.3.4	Installation du Dashboard	87
4.4	Conclusion	94
	Conclusion générale	95
	Bibliographie	97
	Problème Rencontrés	99
	ANNEXE	102

TABLE DES FIGURES

1.1	Architecture générale d'un système virtualisé	6
1.2	Architecture de la virtualisation complète	7
1.3	Architecture de la paravirtualisation	8
1.4	Architecture de la virtualisation à hyperviseur	9
1.5	Les services du Cloud Computing	16
1.6	Les différents types de cloud	18
2.1	Architecture Eucalyptus	28
2.2	Architecture modulaire d'Eucalyptus	29
2.3	Architecture OpenNebula	35
2.4	La relation entre les composants d'OpenStack.	42
4.1	Les propriétés du rôle admin	64
4.2	Les propriétés du rôle Member	64
4.3	Les propriétés du rôle KeystoneAdmin	65
4.4	Les propriétés du rôle KeystoneServiceAdmin	65
4.5	Les propriétés du tenant admin	66
4.6	Les propriétés du tenant service	66
4.7	Les propriétés de l'utilisateur admin	67
4.8	Les propriétés de l'utilisateur glance	68
4.9	Les propriétés de l'utilisateur nova	68
4.10	Les propriétés du service keystone	69
4.11	Les propriétés du service glance	69

4.12	Les propriétés du service nova compute	69
4.13	Les propriétés du service nova volume	70
4.14	Les propriétés du point d'accès du keystone	71
4.15	Les propriétés du point d'accès glance	71
4.16	Les propriétés du point d'accès du nova compute	72
4.17	Les propriétés du point d'accès du nova volume	73
4.18	L'identificateur de l'image après l'ajout à glance	75
4.19	Le bon fonctionnement de nova	79
4.20	Les images disque fournies par glance	79
4.21	La règle de parefeu pour l'accès SSH	80
4.22	La règle de parefeu pour le ping	81
4.23	La création de la clé SSH	81
4.24	Les modèles de VMs existant	82
4.25	Les propriétés de l'instance créée	83
4.26	Les instances créées.	83
4.27	Les propriétés du volume créé.	85
4.28	L'adresse IP publique allouée.	86
4.29	L'attribution manuelle d'une adresse publique.	87
4.30	L'interface d'authentification du dashboard.	88
4.31	L'interface de l'utilisateur admin	89
4.32	L'interface des utilisateurs avec le rôle member	89
4.33	Les services créés.	91
4.34	L'image fournie par le service glance	91
4.35	La création d'une instance	92
4.36	Les instances créées par l'administrateur	92
4.37	Création d'un utilisateur par l'administrateur	93
4.38	Apperçu du bouton help de l'interface dashboard	93

LISTE DES TABLEAUX

2.1	Caractéristiques des différents modes réseaux d'Eucalyptus.	31
3.1	Tableau comparatif des solutions Cloud Computing.	49
3.2	Les versions d'OpenStack.	51

Introduction Générale

Vu que la technologie de l'internet se développe de manière exponentielle depuis sa création, une nouvelle tendance dans le monde des IT (Technologies de l'information et de la communication) a fait son apparition pour accroître la productivité des entreprises et répondre à l'évolution des systèmes d'information en terme de ressources, d'espace et de coût. Il s'agit du Cloud Computing, qui donne la possibilité d'utiliser une puissance de calcul et une capacité de stockage offerte par des serveurs distants.

Cette Technologie, s'appuyant sur le WEB 2.0, donne la possibilité aux sociétés de réduire les coûts d'exploitation des logiciels et des ressources matériels par leur utilisation directement en ligne, elle permet aussi l'utilisation des services, disponibles en tout lieu, à la demande, et pour toute organisation, quelle que soit sa taille. C'est le déplacement progressif de l'informatique vers les réseaux et les centres de données.

Cette technologie est utilisée depuis longtemps sans se rendre compte. Il s'agit de l'utilisation de Facebook, Gmail, Hotmail et beaucoup d'autres services.

Le Cloud Computing s'appuie sur la virtualisation qui permet de mutualiser les ressources et d'utiliser les applications, les environnements de développement et de l'infrastructure matérielle en tant que services.

Dans le présent projet nous allons mettre en place une solution de Cloud Computing, il faudrait pour autant, bien analyser les solutions Open Source et gratuites mises sur le marché et enfin choisir la plus adéquate à nos besoins.

Ce mémoire est organisé en quatre (4) chapitres:

Le *chapitre 1* est consacré à définir le Cloud Computing, citer ses avantages et ses inconvénients, ses différents types, les services qu'il offre, ainsi certaines notions pour la bonne compréhension de ce concept, nous allons aussi étudier la virtualisation sur laquelle repose le Cloud Computing.

Le *chapitre 2* est voué à l'étude de quelques solutions de Cloud Computing Open Source mises sur le marché, l'étude de leur architectures et leur fonctionnements.

Le *chapitre 3* est consacré à l'établissement d'un comparatif entre les trois solutions étudiées précédemment et détailler la solution retenue pour bien comprendre son fonctionnement.

Dans le *chapitre 4* nous allons nous pencher sur l'installation de la solution de Cloud Computing privé choisie, nous allons décrire les différentes étapes à suivre pour la bonne installation et configuration de cette dernière.

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

CHAPITRE 1

Etat de l'art sur la virtualisation et de Cloud Computing

Introduction

L'informatique a toujours évolué, au gré des nouvelles technologies pour répondre à de nouvelles demandes. L'informatique est centralisée avec l'avènement des centres de données. Et surtout, elle se dématérialise et devient "l'informatique dans les nuages", ou Cloud Computing. La puissance informatique se virtualise et se consomme de l'endroit et au moment où on en a besoin et devient extensible tout ceci grâce à l'internet.

Dans ce chapitre nous allons établir un état de l'art sur le Cloud Computing et la virtualisation en expliquant ces deux concepts et la relation entre eux.

1.1 Définition

Le " Cloud " est une métaphore désignant un réseau de ressources informatiques accessibles publiquement ou en privé sur Internet ou Intranet. Le Cloud Computing est une technologie qui utilise l'Internet et des serveurs distants pour gérer les données et les applications. Il permet aux consommateurs et aux entreprises d'utiliser des applications sans installation et d'accéder à leur fichiers personnels depuis n'importe quel ordinateur. Il consiste à fournir des services sur des machines virtuelles allouées dans des plates-formes

de grandes tailles. Par conséquent, la notion de machine n'existe plus, l'entreprise stocke ses données sur Internet et utilise le Cloud pour travailler[1].

On utilise aussi l'expression " Pay-for-use " pour définir le Cloud computing, c'est-à-dire qu'il s'agit d'un service que l'on paye seulement quand on en a besoin[1].

Avant d'entamer la partie Cloud nous allons décrire la technologie de virtualisation sur la quelle repose fortement le nouveau concept "Cloud Computing".

1.2 La virtualisation

Dans ce qui suit nous allons définir la virtualisation et expliquer ses techniques ainsi ses solution.

1.2.1 Définition

La virtualisation est l'ensemble des techniques matérielles et/ou logicielles qui permettent de faire fonctionner sur une seule machine physique plusieurs systèmes d'exploitation et/ou applications cloisonnées et séparés les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes[2].

Le système d'exploitation virtualisé aussi appelé système invité est vu par l'hôte comme un simple programme, il n'a pas d'accès direct au matériel contrairement à l'hôte[2].

1.2.2 Les Avantages et les inconvénients de la virtualisation

1. Les Avantages de la virtualisation

- Utiliser un autre système d'exploitation sans redémarrer son ordinateur.
- Tester des logiciels dans des environnements isolés et sécurisés.
- Meilleure utilisation des ressources machines.
- Gain de place physique, économie d'énergies.
- Réduction du nombre de machines donc du coût du matériel et de sa maintenance.
- Optimisation des ressources d'un serveur (surtout au niveau du processeur).

- Possibilité d'installer plusieurs systèmes (Windows, Linux) sur une même machine[2].

2. Les inconvénients de la virtualisation

- Pertes plus importantes en cas de chute d'une machine physique (plusieurs services indisponibles).
- Vulnérabilité généralisée: si l'hyperviseur est exposé à une faille de sécurité, les machines virtuelles peuvent l'être également et ne sont plus protégées. Cela a été pris en considération par les développeurs de solutions de virtualisation en adoptant des techniques d'isolation qu'on va détailler dans la partie suivante[2].

Les intérêts de la virtualisation sont atteints grâce à plusieurs technologies ayant des buts différents. Il est important de définir ces technologies.

1.2.3 Les technologies de virtualisation

1. La virtualisation complète

La virtualisation complète (full virtualization), consiste à émuler l'intégralité d'une machine physique pour le système invité. Le système invité " croit " s'exécuter sur une véritable machine physique.

Le logiciel chargé d'émuler cette machine s'appelle une machine virtuelle (VM), son rôle est de transformer les instructions du système invité en instructions pour le système hôte. Comme le montre la figure suivante, la machine virtuelle est un programme comme un autre du point de vue du système hôte, au même titre qu'un navigateur Internet ou un traitement de texte, etc.

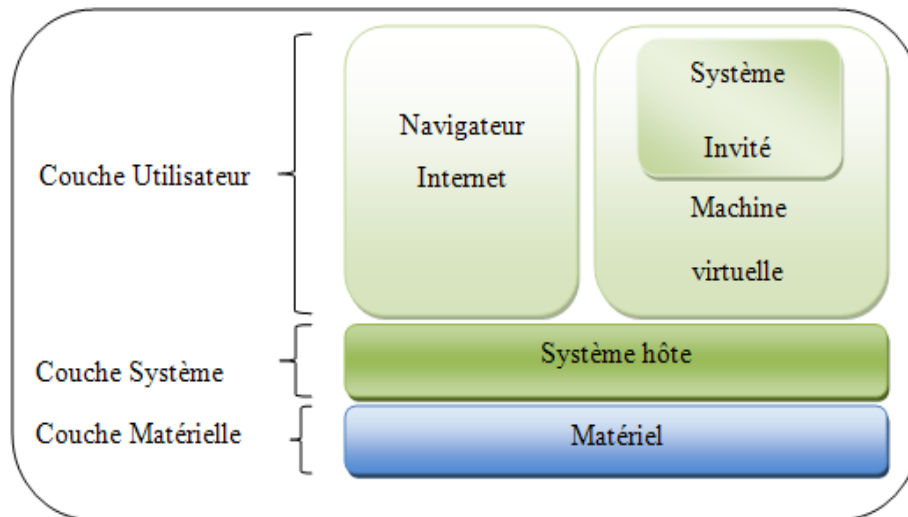


FIG. 1.1 – Architecture générale d'un système virtualisé

[2]

Les programmes utilisateurs n'ont pas d'accès direct au matériel, mais uniquement aux couches d'abstraction. La machine virtuelle émule donc de manière logique (c'est-à-dire avec du code) tout le matériel habituel de l'architecture de l'ordinateur cible.

La caractéristique principale de la virtualisation complète est que les systèmes invités n'ont pas à être modifiés pour être utilisés dans une machine virtuelle utilisant une technologie de virtualisation[2]. La Figure.1.2 montre l'architecture générale d'un système utilisant la technique de virtualisation complète.

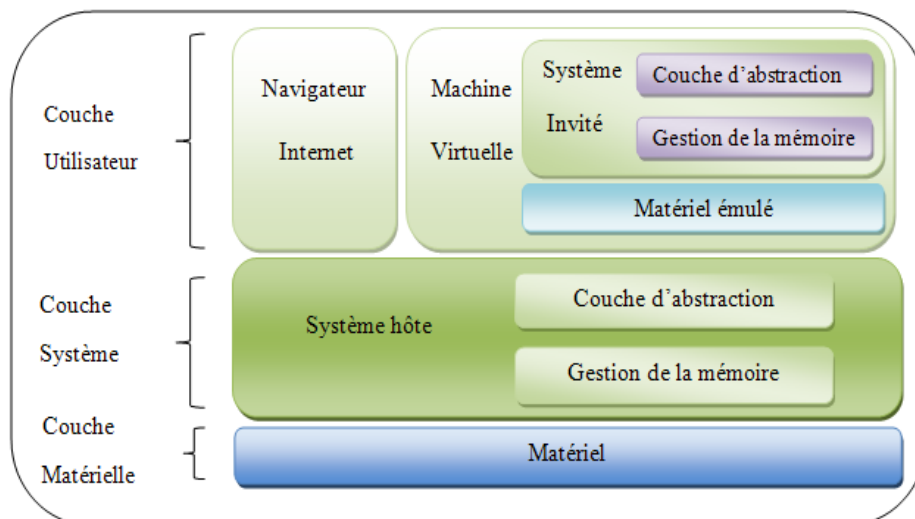


FIG. 1.2 – Architecture de la virtualisation complète

[2]

2. La paravirtualisation

La paravirtualisation 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. Dans une solution de paravirtualisation, le système invité est modifié pour être exécuté par la machine virtuelle[2].

En pratique, un système paravirtualisé possède quelques pilotes de périphériques et sous-systèmes modifiés, qui lui permettent de communiquer directement avec la machine virtuelle, sans avoir à passer par une couche d'abstraction pour parler au matériel virtuel.

Les pilotes paravirtualisés échangent directement des données avec la machine virtuelle sans avoir à passer par une émulation du comportement de matériel. Les parties du système invité généralement modifiées pour tirer profit de la paravirtualisation sont la gestion de la mémoire et la gestion des E/S. La Figure.1.3 présente l'architecture générale d'un système utilisant la paravirtualisation[2].

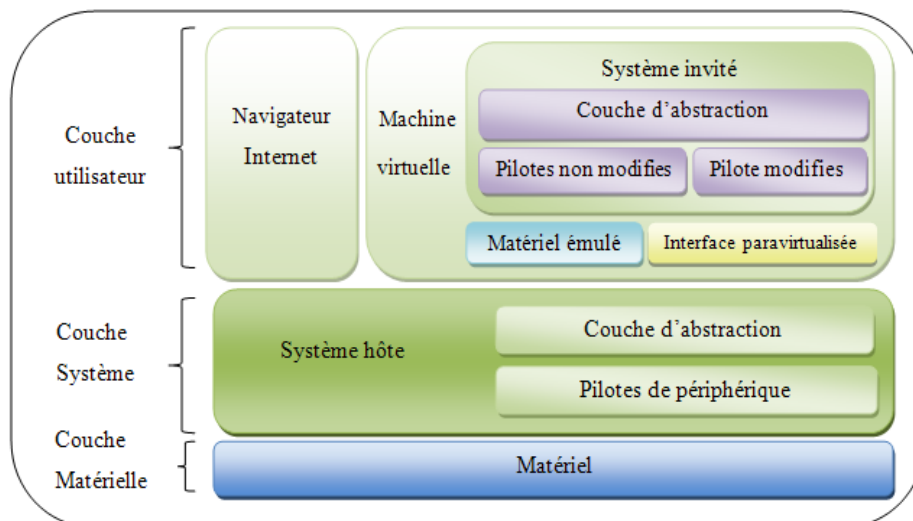


FIG. 1.3 – Architecture de la paravirtualisation

[2]

Tout comme la virtualisation complète, la paravirtualisation garde une séparation nette entre le système invité et le système hôte. De ce fait, seul le système hôte a un accès direct et exclusif au matériel. Le système invité doit donc toujours passer par la machine virtuelle pour accéder au matériel, qui passe à son tour par la couche d'abstraction. On peut donc améliorer davantage le processus en laissant au système invité un accès direct mais contrôlé au matériel. C'est le but des systèmes à hyperviseur[2].

3. Les systèmes à hyperviseur

L'utilisation d'un hyperviseur (hypervisor) est en quelque sorte l'évolution de la paravirtualisation, si l'on recherche encore une amélioration des performances. Dans les technologies précédentes, le système hôte était le seul à avoir un accès direct au matériel; avec un hyperviseur, le système hôte partage cet accès avec les systèmes invités.

L'hyperviseur prend le contrôle du matériel. Ensuite, il fait appel à un système d'exploitation complet, qui sera donc exécuté par-dessus de l'hyperviseur. Ainsi, le système d'exploitation doit passer par l'hyperviseur pour tout accès au matériel.

L'hyperviseur s'assure qu'un système invité n'accède qu'aux ressources autorisées, sans perturber le fonctionnement des autres systèmes. La Figure.1.4 présente l'architecture générale d'un système à hyperviseur.

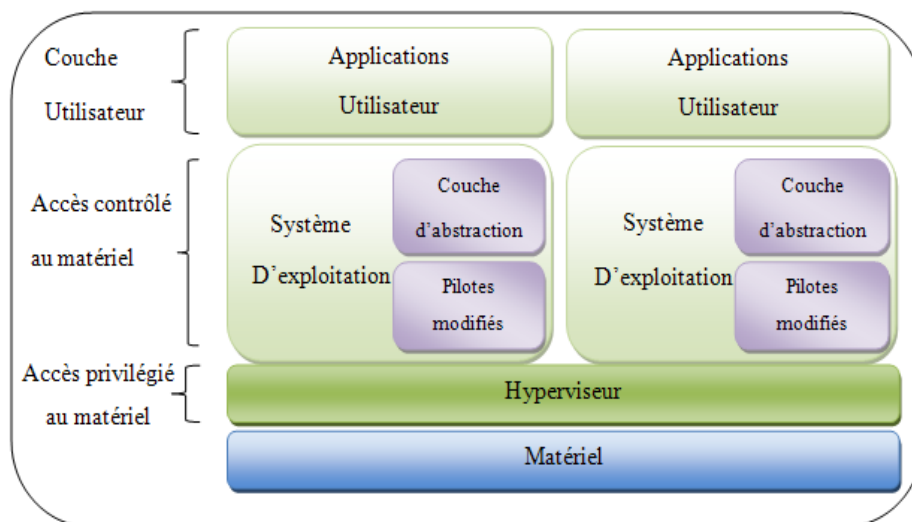


FIG. 1.4 – Architecture de la virtualisation à hyperviseur

[2]

A la différence des technologies exposées précédemment, il n'y a cette fois pas d'accès direct au matériel (rectangle bleu) pour le système d'exploitation, uniquement une couche d'abstraction minimale fournie par l'hyperviseur (rectangle vert), le système hôte et les systèmes invités cohabitent donc au même niveau de privilège, uniquement régulés par l'hyperviseur.

Si les deux technologies vues précédemment (virtualisation complète et paravirtualisation) utilisaient une machine virtuelle pour émuler le matériel, il n'en va pas de même avec un hyperviseur. Chaque système d'exploitation a un accès presque direct au matériel, par l'intermédiaire de l'hyperviseur. Il n'y a donc plus de couche d'abstraction logicielle, le matériel accessible est celui de la machine physique, avec toutes les fonctionnalités qu'il peut offrir[2].

4. Le cloisonnement

Le cloisonnement est une technologies visant à séparer les processus s'exécutant sur un même système d'exploitation. Elle vise à isoler chaque processus dans un

conteneur dont il est théoriquement impossible de sortir. Un processus isolé ne saura pas quels autres processus s'exécutent sur le même système, et n'aura qu'une vision limitée de son environnement. Le but principal de cette technologie est d'améliorer la sécurité du système d'exploitation et des applications.

Ainsi, nous ne pourrions pas avec le cloisonnement proposer un système d'exploitation différent pour un processus isolé. Par exemple, si le système hôte est Solaris (version d'Unix propriétaire développée par Sun), alors tous les processus s'exécutant à l'intérieur d'une " zone " auront accès à la même version de ce Solaris[2].

Parmi les solutions majeures et Open Source de la virtualisation, nous pouvons citer: QEMU, Xen, KVM et VMware etc.

1.2.4 Solutions de virtualisation

Dans ce qui suit nous allons présenter les solutions QEMU, KVM.

1. QEMU

QEMU (Quick EMUlator) est diffusé sous la licence GNU GPL. QEMU utilise la virtualisation complète. Il fonctionne sur les plates-formes les plus courantes (Microsoft Windows, GNU/Linux, Mac OS X).

Ce qui distingue QEMU des autres est sa grande simplicité d'usage. En effet, comme c'est un projet utilisant la virtualisation complète, il y a un simple programme à exécuter sur le système hôte pour obtenir une nouvelle machine virtuelle contenant un système invité. L'un des gros points forts de QEMU est la flexibilité (s'adapter aux besoins à la demande) qu'il offre au niveau des options pour la configuration de la machine virtuelle. Qemu est la technologie de virtualisation sur laquelle repose KVM, VirtualBox...[28].

2. KVM

KVM (Kernel-based Virtual Machine - machine virtuelle dans le noyau) est une solution de virtualisation Open Source sous licence GNU GPL. C'est un projet de virtualisation complète qui utilise les instructions de virtualisation.

KVM propose en effet la migration "à chaud" des systèmes invités entre les hôtes, sans interruption de services, la migration à chaud est transparente pour l'utilisateur final, la machine virtuelle reste sous tension, les connexion réseau restent actives et les applications utilisateurs continuent à fonctionner pendant que la machine virtuelle est relocalisée dans un nouvel hôte physique.

En plus de la migration à chaud, KVM supporte l'enregistrement de l'état actuel d'une machine virtuelle sur le disque (snapshot), afin de la dupliquer et de la restaurer à une date ultérieure[28].

1.3 La relation entre le cloud computing et la virtualisation

Les fortes avancées dans le domaine de la virtualisation ont rendu possible le Cloud Computing. Elle permet d'optimiser les ressources matérielles en les partageants entre plusieurs environnements (" time-sharing ").

Le Cloud Computing couplé aux technologies de virtualisation, permet la mise à disposition d'infrastructures et de plate-forme à la demande. Mais le Cloud Computing ne concerne pas seulement l'infrastructure (IaaS), il inclut aussi la plate-forme d'exécution (PaaS) et les applications (SaaS), le Cloud est à la fois transversal et vertical.

1.4 Historique du Cloud Computing

Il n'y a pas de date-clé à laquelle nous puissions dire que le Cloud Computing est né. Mais il est admis que le concept de Cloud Computing a été initié par Amazon en 2002. Ce dernier avait alors investi dans un parc informatique afin de palier aux surcharges des serveurs dédiés au commerce en ligne constatées durant les fêtes de fin d'année. A ce moment-là, Internet comptait moins de 600 millions d'utilisateurs mais les achats en ligne étaient en pleine augmentation. En dépit de cette augmentation, les ressources informatiques d'Amazon restaient peu utilisées une fois que les fêtes de fin d'année étaient passées. Ce dernier a alors eu l'idée de louer ses capacités informatiques le reste de l'année

à des clients pour qu'ils stockent les données et qu'ils utilisent les serveurs. Ces services étaient accessibles via Internet et avec une adaptation en temps réel de la capacité de traitement, le tout facturé à la consommation. Cependant, ce n'est qu'en 2006 qu'Amazon comprit qu'un nouveau mode de consommation de l'informatique et d'internet faisait son apparition[29].

Bien avant la naissance du terme de Cloud Computing, les informaticiens utilisaient déjà des services de Cloud comme le webmail2, le stockage de données en ligne (photos, vidéos, ...) ou encore le partage d'informations sur les réseaux sociaux[29].

Dans les années 1990, un autre concept avait déjà préparé le terrain au Cloud Computing. Il s'agit de l'ASP (Application Service Provider) qui permettait au client de louer l'accès à un logiciel installé sur les serveurs distants d'un prestataire, sans installer le logiciel sur ses propres machines. Le Cloud Computing ajoute à cette offre la notion d'élasticité avec la possibilité d'ajouter de nouveaux utilisateurs et de nouveaux services d'un simple clic de souris[29].

La virtualisation a été la première pierre vers l'ère du Cloud Computing, ce concept permet de formaliser une offre de services informatiques dématérialisés à la demande[29].

Le Cloud Computing est enfin apparu avec les différents progrès technologiques réalisés durant ces 50 dernières années, tant sur le plan matériel, logiciel et conceptuel, aux avancées des mécanismes de sécurité, à l'élaboration de réseaux complexes mais standardisés comme Internet, et à l'expérience dans l'édition et la gestion de logiciels, services, infrastructures et stockage de données[7].

1.5 Les différents services du Cloud Computing

Il existe trois types de cloud computing :

1.5.1 Saas (Software as a Service)

Le Saas est un concept consistant à proposer un abonnement à un logiciel plutôt que l'achat d'une licence. On oublie donc le modèle client-serveur et aucune application n'est

installée sur l'ordinateur, elles sont directement utilisables via un navigateur Web.

L'utilisation reste transparente pour les utilisateurs, qui ne se soucient ni de la plateforme, ni du matériel[6].

Le SaaS remplace l'ASP (Application Service Provider), aussi appelé fournisseur d'applications hébergées (FAH). Une ASP désigne une application fournie comme un service. Plutôt que d'installer le logiciel sur le poste client en ayant à assurer les phases d'installations et de maintenance sur chaque poste, ces applications sont hébergées et centralisées sur un serveur unique et accessible par les clients au travers de protocole standard. C'est par exemple le cas avec des applications Web accessibles par http, il n'y a alors plus de déploiement ou de maintenance à effectuer sur le poste utilisateur, celui-ci n'a alors besoin que d'un simple navigateur Internet. Le déploiement, la configuration, la maintenance, la sauvegarde, etc sont désormais de la responsabilité du fournisseur du service, le client est alors consommateur[7].

- **Avantage:**

- Plus d'installation.
- Plus de mise à jour (elles sont continues chez le fournisseur).
- Paiement à l'usage.

- **Inconvénient:**

- Limitation par définition au logiciel proposé.
- Pas de contrôle sur le stockage et la sécurisation des données associées au logiciel.

- **Fournisseurs:** tous les fournisseurs en ligne. Citons par exemple: googleApp et « Exchange Online »¹ de Microsoft.

- **Consommateurs:** la quasi-totalité des internautes.

Gmail et facebook sont des exemples de SaaS très connus, très utilisés et gratuits.

1. « Exchange Online »: permet de bénéficier d'une messagerie fournie en tant que service hébergé par Microsoft, sans devoir la gérer en interne.

1.5.2 Paas (Platform as a Service)

Il s'agit des plateformes du nuage, regroupant principalement les serveurs mutualisés et leurs systèmes d'exploitation. En plus de pouvoir délivrer des logiciels en mode SaaS, le PaaS dispose d'environnements spécialisés au développement comprenant les langages, les outils et les modules nécessaires. En bref, cette catégorie représente les services que les développeurs utilisent pour créer des applications.[6][7].

- **Avantage:**

- Ces environnements sont hébergés par un prestataire basé à l'extérieur de l'entreprise ce qui permet de ne disposer d'aucune infrastructure et de personnel de maintenance et donc de pouvoir se consacrer au développement.
- Pas de logiciel supplémentaire à acheter ou à installer.

- **Inconvénient:**

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

- **Fournisseurs:** Force.com, Google App Engine.

- **Consommateurs:** développeurs Web et développeurs d'applications.

1.5.3 IaaS (Infrastructure as a Service)

Il s'agit de la mise à disposition, à la demande, de ressources d'infrastructures dont la plus grande partie est localisée à distance dans des Datacenter ². L'IaaS permet l'accès aux serveurs et à leurs configurations pour les administrateurs de l'entreprise.

Le client a la possibilité de louer des clusters ³, de la mémoire ou du stockage de données. Le coût est directement lié au taux d'occupation. Une analogie peut être faite

2. Datacenter: centre de traitement de données.

3. Clusters: est une grappe de serveurs constituée de deux serveurs au minimum et partage une baie de disque commune, pour assurer une continuité de service et/ou répartir la charge de calcul et/ou la charge réseau.

avec le mode d'utilisation des industries des commodités (électricité, eau, gaz) ou des télécommunications[6][7].

- **Avantage:**

- Grande flexibilité, contrôle total des systèmes (administration à distance par SSH ⁴ou Remote Desktop ⁵).
- Tout est de la responsabilité du prestataire.
- Solution moins coûteuse (plus rentable de louer des serveurs virtuels)

- **Inconvénient:**

- Besoin d'administrateurs système comme pour les solutions de serveurs classiques sur site.
- La confidentialité des données.
- Plus d'accès aux données en cas de défaillance de la liaison.

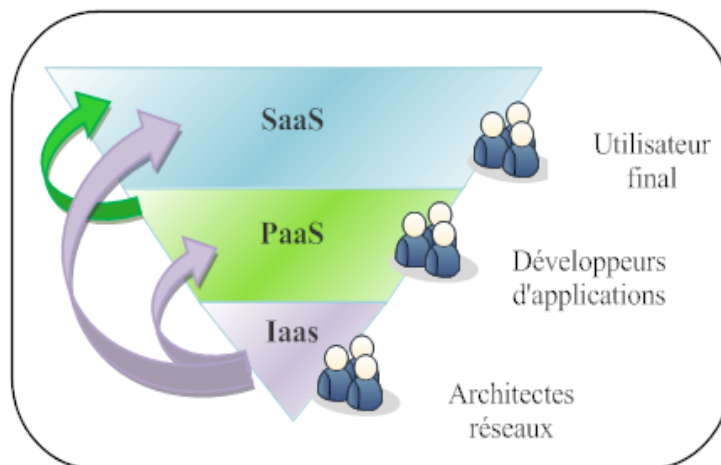
- **Fournisseurs:** l'un des exemples d'IaaS est Amazons Elastic Compute Cloud (EC2).

- **Consommateurs:** fournisseurs de développement et fournisseurs informatiques[7].

Citons quelques services IaaS ouverts au grand public qui permettent le stockage en ligne: Dropbox, googleDrive etc, Eucalyptus est aussi l'une des solutions IaaS mais qui est privée. La Figure.1.5 montre les différents services du Cloud Computing et ses utilisateurs.

4. SSH: est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées. Il devient donc impossible d'écouter ce que fait l'utilisateur.

5. Remote Desktop: permet aux utilisateurs de voir et de contrôler leurs PC connectés à distance, à travers l'Internet à partir de n'importe quel ordinateur.

FIG. 1.5 – *Les services du Cloud Computing*

[7]

Pour simplifier ces différentes définitions, on peut retenir qu'avec le SaaS on utilise une application, avec le PaaS on construit ses applications et finalement l'IaaS permet d'héberger le tout.

1.6 Types de Cloud

Selon le type d'hébergement de l'offre de Cloud Computing, nous trouvons quatre (04) catégories:

1.6.1 Le Cloud privé(interne)

Le Cloud privé est un réseau informatique propriétaire ou un centre de données qui fournit des services hébergés pour un nombre limité d'utilisateurs.

Il peut être géré par l'entreprise utilisatrice elle-même ou par un prestataire externe qui met à disposition de l'utilisateur un parc de machines s'adaptant à la demande de l'utilisateur (Cloud privé virtuel). A noter qu'une même infrastructure peut accueillir plusieurs clouds privés virtuels appartenant à différents utilisateurs, chacun pouvant accéder à son cloud privé via son propre réseau[29].

1.6.2 Le Cloud public (externe)

Dans le Cloud public les utilisateurs ont accès à des services Cloud via l'Internet sans savoir précisément où sont hébergées leur données ni où sont exécutés leur traitements. Les ressources informatiques et bases de données de l'utilisateur peuvent être hébergées dans n'importe quel datacenter du prestataire et peuvent passer d'un datacenter à l'autre pour optimiser les capacités du prestataire.

Un Cloud public est un service IaaS, PaaS ou SaaS proposé et hébergé par un tiers[29]. Dans l'heure actuelle il existe un bon nombre de clouds publics: Amazon Elastic Compute Cloud (EC2), Sun Cloud, Google App Engine.

1.6.3 Le Cloud hybride (interne et externe)

Le Cloud hybride associe à la fois les infrastructures des Cloud privés et publics. Une partie des données ou des infrastructures est gérée en interne par l'entreprise, dans ses locaux ou chez un prestataire et communique avec des ressources Cloud.

Il permet de différencier le lieu de traitement des données selon qu'elles soient stratégiques ou pas: les données sensibles pourront alors être traitées dans les murs de l'entreprise alors que les autres le seront par un Cloud public. Le Cloud public peut également être une solution pour lisser un pic d'activité lorsque les capacités de l'entreprise sont dépassées[29].

1.6.4 Le Cloud communautaire:

Le Cloud communautaire permet à plusieurs entreprises ou organisations de partager des ressources en mode Cloud, qui sont alors exclusivement dédiées à ces organisations. Le Cloud communautaire peut être géré par les organisations membres ou par un prestataire externe.

Il peut également permettre à plusieurs utilisateurs de se constituer un Cloud ayant les caractéristiques d'un Cloud privé en termes de sécurité et de ressources dédiées, à moindre coût et avec un gage d'indépendance vis-à-vis d'un prestataire de Cloud public[29]. La Figure.1.6 représente les les différents type du Cloud.

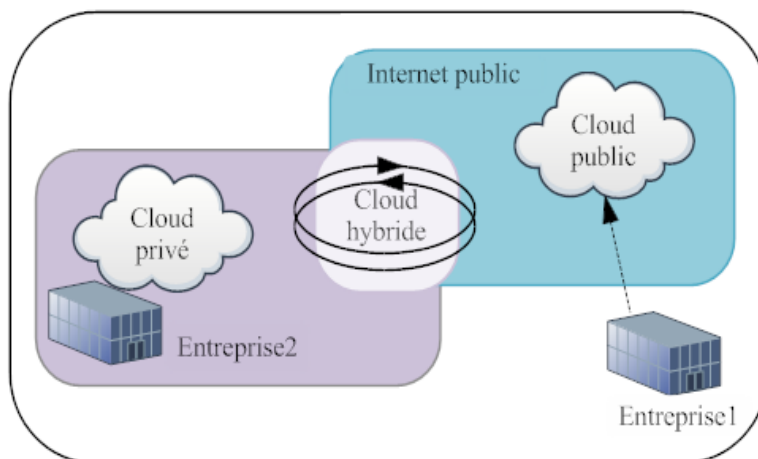


FIG. 1.6 – Les différents types de cloud

[5]

1.7 Avantages et Inconvénients du Cloud Computing

1.7.1 Avantages du Cloud Computing

- **Une solution flexible:** les systèmes Cloud sont évolutifs, les services et l'utilisation peuvent ainsi être augmentés ou diminués à la demande, en fonction de l'activité et du besoin des utilisateurs.
- **Souplesse d'évolution:** le client dispose souvent en temps réel des évolutions de la plateforme de Cloud Computing, étant donné qu'il n'y a pas de logiciels à installer et que l'accès se fait avec un simple navigateur web.
- **Simplicité:** l'entreprise cliente n'a plus besoin de développements coûteux et déplace la responsabilité du fonctionnement et de maintenance sur le fournisseur.
- **Liberté de changer:** le Cloud étant généralement facturé à la demande ou par abonnement mensuel, il est très facile pour une entreprise d'arrêter le service si elle n'en a plus besoin ou si elle souhaite aller chez un concurrent.
- **Mobilité:** à partir du moment où les employés disposent d'un accès Internet, ils peuvent travailler de n'importe quel endroit, puisque toutes les données sont stockées dans le Cloud, elles restent toujours accessibles quoi qu'il arrive aux ordinateurs portables et les utilisateurs n'ont plus besoin de les porter où ils vont.
- **Coût:** le Cloud permet des coûts d'entrée faibles à ses services, sa force réside dans

la possibilité de proposer le même service à un grand nombre d'utilisateurs. Ainsi dans la conversion du CAPEX ⁶ en OPEX ⁷[4][5].

1.7.2 Inconvénients du Cloud Computing

- **Confidentialité et sécurité des données:** les données sont hébergées en dehors de l'entreprise. Cela peut donc poser un risque potentiel pour l'entreprise de voir ses données mal utilisées ou volées. Il s'agit donc de s'assurer que le fournisseur dispose d'une sécurité suffisante et qu'il propose une politique de confidentialité concernant les données de l'utilisateur par les SLA ⁸.
- **Dépendance à Internet:** en absence de connexion, on n'a plus accès aux services.
- **Mauvaise utilisation des ressources:** plutôt que d'utiliser la puissance répartie de millions d'ordinateur, on centralise les traitements sur quelques serveurs et on sature la bande passante d'Internet[4][5].

1.8 Les principaux acteurs du Cloud Computing

Dans cette partie nous allons parcourir les principaux fournisseurs de services dans le Cloud et observer les différentes offres proposées.

1.8.1 Le Cloud d'Amazon

Amazon, met à disposition un Cloud public au travers d'Amazon Web Services (AWS) depuis 2006.

Amazon propose une infrastructure avec " Amazon EC2 " qui présente un environnement informatique virtuel, permettant d'utiliser des interfaces de service Web pour lancer des instances avec une variété de systèmes d'exploitation[7].

6. CAPEX: c'est le fait d'acheter des serveurs et des équipements de stockage pour constituer son propre centre de données.

7. OPEX: c'est payer à l'utilisation un ensemble de services " Cloud " d'hébergement et de stockage.

8. SLA (Service Level Agreement): c'est un document ou contrat qui définit la qualité de service requise entre un prestataire et un client.

Amazon met à disposition des machines virtuelles nommées les " Amazon Machine Images " (AMI), qui sont déjà préconfigurées et prêtes à l'emploi avec différents systèmes d'exploitations disponibles dont Windows Server, Ubuntu, Debian... chaque AMI dispose d'une variété de logiciels pour la gestion de bases de données (MySQL, Oracle, Apache, etc.), le développement d'applications, etc[8].

Amazon propose aussi le service Amazon Simple Storage (S3), qui est un service de stockage en ligne, accessible au travers d'API standard mettant à disposition un stockage de fichiers illimités, il est utilisé conjointement avec d'autres fonctionnalités des services Cloud d'Amazon[7].

1.8.2 Le Cloud de Google

Google propose une plateforme nommée Google App Engine qui permet de développer des applications au moyen de Java, JPython⁹ et autre. Google permet aussi un service SaaS avec Google Apps, telles que Google Docs qui est une suite bureautique Web permettant le stockage et l'édition de documents, de feuilles de calcul, de schémas ou de présentations. Gmail est l'un des " web apps " gratuit de Google[7][8].

1.8.3 Le Cloud de Microsoft

La plateforme Windows Azure qui permet de migrer une application existante ou d'en développer dans différents langages de programmation dont Rubis, Python, Java ou PHP à l'aide d'outils de programmation classiques tel que Visual Studio ou Eclipse. La plateforme met également à disposition le service SQL Azure qui un service de base de données basé sur Microsoft SQL Server[8].

Les principales solutions SaaS de Microsoft sont Office 365 qui propose des applications web compatibles avec les formats de la suite Microsoft Office classique (Word, Excel, PowerPoint, Publisher...). Plusieurs personnes peuvent donc éditer un document en même temps, il est possible d'avoir ensuite un aperçu des personnes ayant consulté et édité un document[8].

9. JPython: c'est un Langage de script, libre, orienté objet au quel est ajouté un framwork java

1.9 Le Choix entre le Cloud privé et le Cloud public

Pour les solutions de type IaaS ou PaaS, il est important de choisir entre l'utilisation des services qu'offre une solution de Cloud privé ou public. Dans ce qui suit nous allons définir les inconvénients du Cloud public qui renforce le choix de la mise en place d'une solution du Cloud privé.

1.9.1 Inconvénients du cloud computing public

- **Cadre légal:** il n'y a aucun accès physique aux données transférées dans le Cloud public. l'utilisateur n'a aucune connaissance de l'endroit où sont stockées ses données car ces dernières ne sont pas forcément présentes sur le territoire national de l'utilisateur, ceci peut poser un problème de mauvaise utilisation de ses données. C'est actuellement le problème majeur du Cloud Computing public
- **Débit internet:** l'utilisation du Cloud public nécessite le transfert de données d'une manière intensive, pour cela il est obligatoire d'avoir une connexion très performante, ce qui est difficile pour des entreprises situées dans des endroits peu desservis.
- **La pérennité du service:** toutes les entreprises utilisant le Cloud public sont dépendantes de leur hébergeur. L'arrêt de son activité, par exemple, pourrait être très problématique. Et un changement d'hébergeur prend du temps, et peut nécessiter un recodage des applications[31].

1.9.2 Le Cloud Computing privé

Si le Cloud Computing public propose des ressources informatiques hébergées (serveurs, stockage, puissance de calcul, applications...) distantes et mutualisées, les offres de Cloud Computing privées se distinguent par leur aspect dédié:

- L'entreprise est propriétaire de son infrastructure et permet de répondre à un besoin personnalisé de ressources informatiques à la demande par le biais d'un réseau interne.
- **Réactivité:** le clonage, les migrations à chaud, ou encore le déploiement de VMs sont des opérations très rapides à réaliser.

- **Sécurité:** le Cloud privé offre un niveau de sécurité maximal vu que les données sont stockées en interne. L'isolation est garantie et des normes de sécurité sont définies spécifiquement pour l'entreprise[31].

Tout comme le Cloud Computing public, le Cloud privé repose sur quatre briques technologiques complémentaires et indissociables. Il s'agit de la virtualisation, des serveurs, du stockage et du réseau. Un ensemble d'éléments auxquels peut s'ajouter une cinquième, à savoir une solution d'administration et d'allocation dynamique des ressources virtuelles sous forme de services[31].

Nous allons développer notre raisonnement sur une solution de Cloud privé en étudiant quelques solutions les plus connues.

1.10 Problématique

A l'heure actuelle la majorité des utilisateurs travaillent sur leur ordinateurs personnels et stockent leur données sur ces derniers, ils sont donc obligés de travailler sur des machines performantes et puissantes (en terme d'espace disque, RAM...), en cas de panne leur données sont perdues. Ainsi quand un utilisateur développe une application par exemple il sera obligé d'installer les outils nécessaires (langage de programmation...) sur chaque machine où il travaille et d'effectuer à chaque fois les mises à jours requises pour le bon fonctionnement de ses applications.

Le Cloud Computing privé permettra aux utilisateurs ayant accès, la gestion flexible du provisionnement des noeuds de calcul, du provisionnement du stockage et du réseau pour répondre à leurs besoins spécifiques. Au lieu d'effectuer ces actions de façon centralisée comme cela se passe avec la virtualisation, les utilisateurs pourront ajouter des ressources à la demande à leur infrastructure, selon des règles définies par l'administrateur de l'infrastructure ce qui décentralise le provisionnement.

l'utilisateur peut facilement instancier une machine virtuelle, il suffira donc de sélectionner dans le catalogue l'image du système d'exploitation désiré(par exemple windows) et d'instancier cette infrastructure "à la demande". En quelques minutes, celle-ci sera disponible

et prête à l'emploi. C'est ici toute la puissance du Cloud Computing: une informatique à la demande.

Il serait alors très utile et très pratique de mettre en place une solution de Cloud Computing pour répondre à ces besoins.

La solution de Cloud Computing permet:

- D'unifier les moyens d'accès à des applications et à des environnements de travail.
- L'utilisateur peut créer, lancer et arrêter des instances de serveurs (ou VMs) en fonction de ses besoins (CPU, RAM...). Une fois s'est fait, il peut utiliser librement les ressources allouées.
- Stocker des données dans la limite de ses possibilités
- Personne n'a le droit de modifier ou supprimer ou lire les données d'un autre utilisateur.
- L'utilisateur peut travailler sur sa propre machine virtuelle de n'importe quel endroit.
- L'accès au service se fait par un navigateur web.

En bref l'objectif principal de notre projet étant de trouver une solution de Cloud et la mettre en oeuvre, celle ci va permettre à l'utilisateur de faire appel à une machine virtuelle ou plus, selon des règles dictées par l'administrateur, et cela après son authentification au près du système, cette VM est choisie ou créée selon son système d'exploitation ainsi son espace disque et autre. Elle ne sera utilisée que par son utilisateur dédié, qui va gérer son environnement de travail lui même.

Conclusion

Cette partie nous a permis d'approfondir nos connaissances par rapport à la virtualisation, ses techniques et ses solutions existantes. Ainsi se familiariser avec le concept Cloud Computing et de bien comprendre son objectif, ses types, ses services et les offres proposées par les différents acteurs. Dans le chapitre suivant nous allons proposer quelques solutions de Cloud Computing et les étudier.

CHAPITRE 2

Les solutions Cloud Computing open source

Introduction

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

Dans ce chapitre on va se baser sur les solutions Open source et gratuites, nous allons étudier leur fonctionnement, leur architecture ainsi que les mécanismes de sécurité qu'elles offrent.

On devra donc choisir la solution adéquate à nos besoins, pour cela une étude approfondie de chaque solution permettra de choisir correctement.

2.1 La solution Eucalyptus

2.1.1 présentation

Son nom fait référence à l'acronyme " Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems " qui peut se traduire en " Utilitaire d'Architecture informatique élastique pour relier vos programmes à des systèmes fonctionnels " .

Eucalyptus est un ensemble d'outils disponible sous Licence BSD ¹, issue d'un projet de recherche de l'université de Californie à partir de 2007, cette plate-forme Cloud open source est intégrée(en 2009) dans Ubuntu Server et Debian. Ecrite en C, Java et Python ², elle permet de créer des Clouds Iaas (Infrastructure as a service) de type privé, public ou hybride, supporte des machines virtuelles Linux ainsi que les hyperviseurs Xen, KVM et VMware. Elle est compatible avec EC2 d'Amazon[9][10].

Eucalyptus se décline en deux (02) versions, une version Open Source gratuite et une version Entreprise payante. En ce qui concerne l'installation de cette plateforme, différentes solutions liées aux systèmes d'exploitation sont proposées tel que les distributions Windows sont capables de supporter Eucalyptus mais seulement avec la version Enterprise, payante[11].

Eucalyptus a été développé pour soutenir le calcul haute performance (HPC). Il peut être déployé sur toutes les principales distributions Linux, y compris Ubuntu, RHEL / CentOS, OpenSUSE, et Debian[11].

2.1.2 Architecture Eucalyptus

L'architecture d'Eucalyptus se base sur cinq (05) composants principaux :

1. Le Cloud Controller (CLC)

C'est un programme Java qui sert de point d'entrée (Front-End) à l'infrastructure. Il offre d'un côté une interface de gestion et de contrôle conforme aux services EC2/S3, et de l'autre côté permet d'interagir avec les autres composants de l'infrastructure. Il est également possible d'avoir une interface web qui permet aux utilisateurs de gérer certains aspects de l'infrastructure. Il assure trois (03) rôles principaux:

- Surveiller la disponibilité des ressources sur les différentes composantes de l'infrastructure du Cloud.

1. BSD (Berkeley Software Distribution license): est une licence libre utilisée pour la distribution de logiciels, les versions modifiées et redistribuées peuvent ne pas garder une licence libre(libre ou propriétaire).

2. Python: langage de script, libre, orienté objet.

- L'arbitrage des ressources, c'est à dire décider quel cluster sera utilisé pour la virtualisation des instances.
- Monitorer les instances en cours d'exécution.

En résumé, le CLC a une connaissance approfondie de la disponibilité et l'utilisation des ressources dans le nuage et permet de surveiller son état[10][11][35].

2. Le Cluster Controller (CC)

Ce contrôleur sert à déployer et gérer les différents contrôleurs de noeuds. Il sert également à gérer la mise en place du réseau entre les instances des différents noeuds. C'est lui qui communique l'ensemble des informations au contrôleur du cloud (CLC). Il assure quatre (04) fonctions principales:

- Il reçoit les requêtes de déploiement des instances du contrôleur de cloud.
- Il décide sur quel contrôleur de noeuds les instances seront déployées.
- Il contrôle le réseau virtuel entre les instances.
- Il collecte les informations des contrôleurs de noeuds enregistrés et les rapporte au contrôleur de cloud[10][11][35].

3. Node Controller (NC)

Le rôle du noeud est d'héberger et déployer les instances fonctionnantes sur l'hyperviseur. Le contrôleur de noeud fonctionne sur chaque noeud et est chargé de vérifier le cycle de vie des instances en cours d'exécution sur le noeud. Il interagit avec le système d'exploitation et l'hyperviseur en cours d'exécution sur le noeud d'un côté et le contrôleur de cluster (CC) de l'autre côté.

Le contrôleur interroge le système d'exploitation s'exécutant sur le noeud afin de découvrir les ressources physiques du noeud (le nombre de coeurs, la taille de la mémoire, l'espace disque disponible) et aussi de s'informer sur l'état des instances en cours d'exécution sur le noeud et propage ces données au contrôleur de cluster[10][11][35].

4. Walrus Storage Controller (WS3)

Aussi appelé (contrôleur de stockage walrus). Il fournit un service de stockage persistant simple à l'aide des API REST ³ et SOAP ⁴. Il est aussi compatible avec les API S3. ⁵ Il assure trois (03) fonctions principales:

- Le stockage des images de machines virtuelles.
- Le stockage des images prises en fonctionnement à un instant précis (Snapshots).
- Le stockage des fichiers et les services en utilisant l'API S3.

WS3 peut être considéré comme un simple système de stockage de fichiers[10][11][35].

5. storage Controller (SC)

Ce contrôleur fournit un service de stockage persistant pour les instances (les images de machines virtuelles), les données utilisateurs. Ces données peuvent être publiques ou privées, dans ce cas elles seront encryptées. C'est similaire au service Elastic Block Storage (EBS) ⁶[10][35]. Il assure donc trois (03) fonctions essentielles:

- La création de dispositifs EBS persistants.
- Fournir le système de stockage de blocs aux instances.
- Autoriser la création d'images des volumes pour permettre leurs sauvegardes[11].

Le schéma suivant montre le fonctionnement des principaux composants d'Eucalyptus:

3. API REST(Representational State Transfer): est une manière de construire une application pour les systèmes distribués comme le World Wide Web, c'est le style architectural original du Web.

4. SOAP (Simple Object Access Protocol): est un protocole qui permet la transmission de messages entre objets distants.

5. API S3: un service de stockage en ligne d'Amazon via une interface de services web

6. EBS (Elastic Block Store): fournit des volumes de stockage en mode bloc à utiliser avec les instances Amazon EC2.

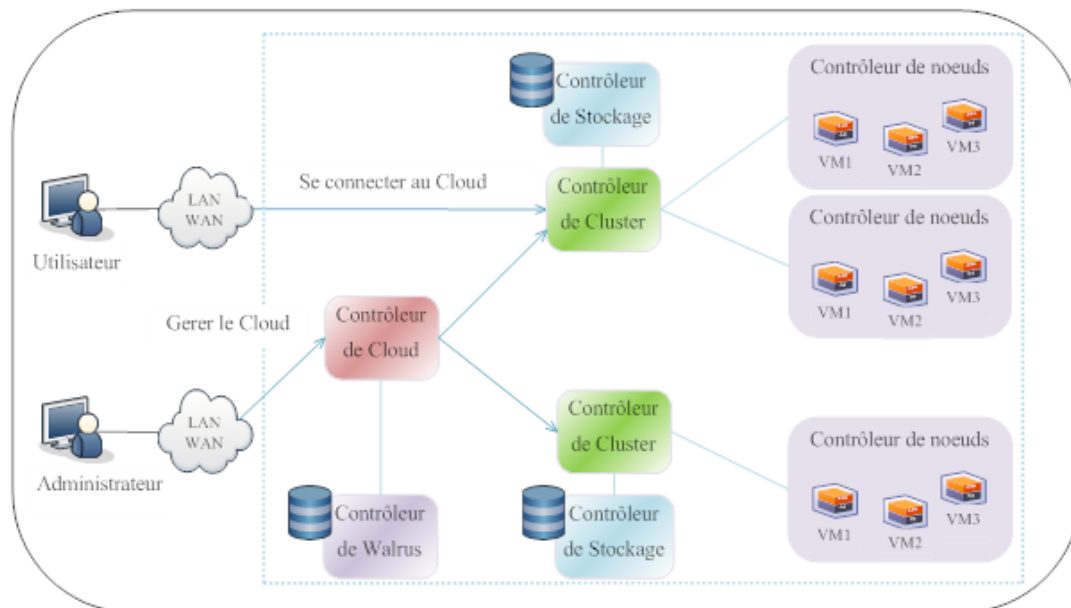


FIG. 2.1 – Architecture Eucalyptus

[35]

2.1.3 Caractéristiques

Ce qui fait la force d'une architecture fondée sur Eucalyptus est sa grande modularité. En effet, les composants d'Eucalyptus ont des interfaces bien définies (via WSDL ⁷).

Le fonctionnement en sous-ensemble qui forme chacun un cluster distinct, permet de faciliter son déploiement sur une infrastructure multi-sites. L'ensemble étant ensuite géré au niveau du contrôleur du cloud, le Walrus assure la communication avec les contrôleurs de stockages[10]. La Figure.2.2 décrit l'architecture modulaire d'Eucalyptus.

7. WSDL: décrit une Interface publique d'accès à un Service Web. C'est une description fondée sur le XML qui indique comment communiquer pour utiliser le service.

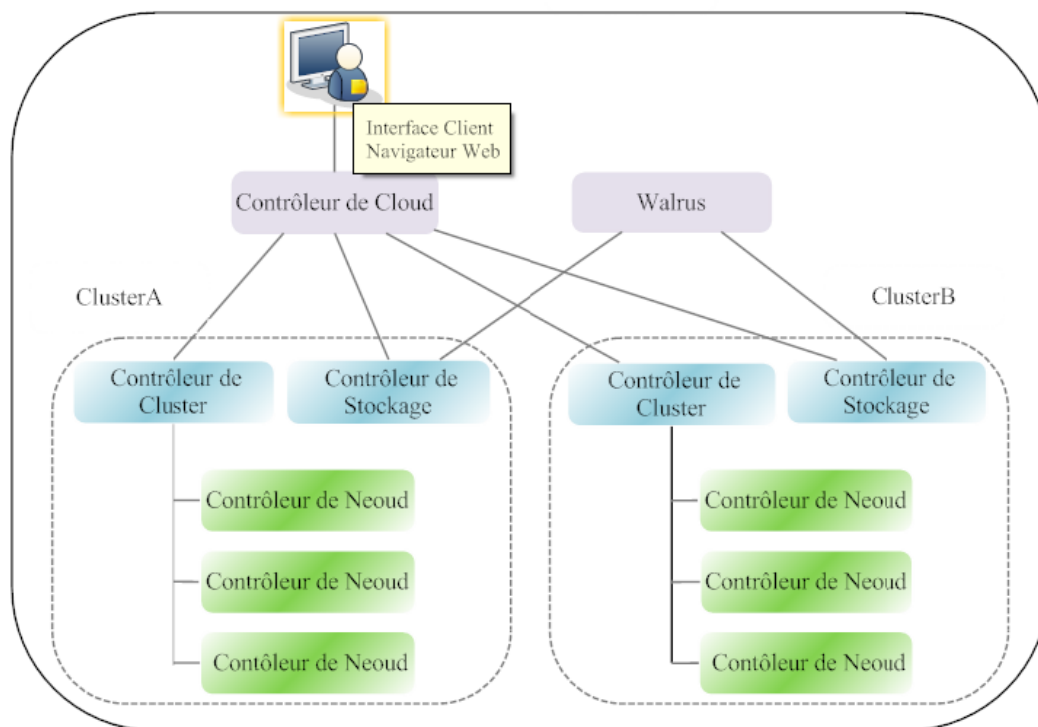


FIG. 2.2 – Architecture modulaire d'Eucalyptus

[10].

Sur le schéma ci-dessus, on constate que le contrôleur de cloud (CLC) et le "Walrus" sont des composants de haut niveau. Ensuite les autres composants sont répartis sur les différents clusters que l'on veut utiliser.

Eucalyptus est flexible et peut être installé sur une configuration très minime, comme il peut être installé sur des milliers de coeurs et des téraoctets de stockage. Et il peut le faire comme une superposition à une infrastructure existante[10].

2.1.4 Gestion des images

La gestion des images des VMs est vitale pour une plateforme de Cloud Computing. En effet, à chaque instantiation de VM, l'image disque "template"⁸ correspondante devra être transférée sur la machine hôte afin de pouvoir démarrer le système désiré. Le stockage et le transfert de ces images est géré de différentes façons[12].

8. Template: il représente un modèle de machine virtuelle à l'état initial avant le démarrage de la VM.

Eucalyptus propose une implémentation de l'API REST Amazon S3 nommée Walrus. Il permet non seulement le stockage de ces images mais propose également un panneau de fonctionnalités permettant leur gestion efficace, via les standards EC2 d'Amazon[12].

Ces images peuvent être encryptées par authentification utilisateur tel que, quand une machine hôte requiert une image pour l'instanciation d'une VM, Walrus est chargé de la vérifier, de la décrypter puis enfin de la transférer. Notons également que Walrus dispose d'une cache d'images décryptées afin d'accélérer le processus lors de l'instanciation d'un grand nombre de machines virtuelles[12].

2.1.5 Réseau et adressage

Afin de mettre en place une infrastructure Cloud, l'administrateur du Cloud devra paramétrer la plateforme choisie afin de proposer un adressage MAC et IP aux machines virtuelles. Cela doit se faire en tenant compte de l'adressage physique des machines hôtes, et de la topologie du réseau les accueillant (les machines hôtes peuvent par exemple être réparties sur plusieurs sites et donc dans des sous-réseaux physiques différents).

Eucalyptus propose quatre (04) modes réseaux différents: MANAGED, MANAGED-NOVLAN, STATIC et SYSTEM:

- **Le mode MANAGED:** permet de profiter de l'ensemble des fonctionnalités réseau proposées par Eucalyptus, notamment l'isolation de groupes de machines virtuelles au sein de différents VLANs. Néanmoins, cela nécessite que le réseau physique des machines hôtes ne comporte pas de VLANs afin d'éviter des conflits entre les VLANs physiques et les VLANs virtuels créés par Eucalyptus. S'il n'y a pas de VLAN présent, l'isolement peut être réalisé en utilisant des groupes de sécurité dans les différents sous-réseaux.
- **Le mode MANAGED-NOVLAN:** offre l'ensemble des fonctionnalités du mode MANAGED mis à part les VLANs. Il permet ainsi de créer des règles de sécurité (groupes de sécurité) et d'utiliser un système d'attribution dynamique des IPs.
- **Le mode STATIC:** permet à l'administrateur du Cloud de réaliser une correspondance statique entre les adresses MAC et les adresses IP, son inconvénient est qu'il ne permet pas la création de groupes de sécurité et des VLAN.

- **Le mode SYSTEM:** permet d'utiliser un serveur DHCP ⁹ externe à Eucalyptus, contrairement aux modes précédents où l'adressage est réalisé de manière interne. Le serveur DHCP externe est chargé de donner une adresse IP à chaque nouvelle VM, selon des règles édictées par l'administrateur réseau (qui peut être différent de l'administrateur cloud). Les VMs apparaissent alors sur le réseau comme des machines physiques. L'inconvénient du mode SYSTEM est qu'il ne permet pas les fonctionnalités d'IPs élastiques, ni de groupes de sécurité, ni de VLAN, il offre donc le plus petit nombre de fonctionnalités de réseautage[12].

Le tableau suivant donne une comparaison entre les modes réseaux d'Eucalyptus

Mode réseau	L'utilisation de serveur DHCP en externe	L'utilisation d'un serveur DHCP en interne	Isolation des instances	Filtrage d'accès
Managed	Non	Oui	Oui	Oui
Managed-NOVLAN	Non	Oui	Non	Oui
Static	Non	Oui	Non	Non
System	Recommandé	Non	Non	Non

TAB. 2.1 – *Caractéristiques des différents modes réseaux d'Eucalyptus.*

9. DHCP (Dynamic Host Configuration Protocol): est un protocole de configuration dynamique d'hôte qui permet d'allouer à la demande des adresses IPs aux machines se connectant au réseau.

2.1.6 Sécurité

Eucalyptus propose dans le mode réseau MANAGED des VLANS permettant d'isoler le trafic réseau des VMs selon des groupes sécurisés d'utilisateurs. Ainsi, un utilisateur d'un groupe A ne peut accéder au trafic présent sur le VLAN d'un groupe B. De plus, l'utilisation des interfaces web services d'Eucalyptus est sécurisée à l'aide de WS-Security¹⁰, qui permet d'encrypter et de signer les requêtes SOAP. Ainsi les données stockées dans le contrôleur de stockage (SC) sont cryptés[12].

Il ya une très forte séparation de l'espace utilisateur et l'espace administrateur. Un Accès root est nécessaire pour que tout soit fait par l'administrateur sur les machines physiques. Les utilisateurs sont autorisés à accéder au système par l'intermédiaire d'une interface Web ou un certain type d'outils frontaux (Euca2ools)[10].

Chaque contrôleur est authentifié par clé SSH et des fichiers de permission pour authentifier toutes les transactions[10].

2.1.7 Stockage

Consiste en un contrôleur principal Walrus qui fournit un service de stockage persistant compatible avec les API S3. Ainsi des contrôleurs de stockage sur chaque contrôleur de noeud pour les instances et les données utilisateurs, C'est similaire au service Elastic Block Storage (EBS) d'Amazon.

2.1.8 Les interfaces fournies

L'interface actuelle d'Eucalyptus qui est "Euca2ools" est compatible avec les API Amazon EC2, S3 et EBS gérant toutes les facettes de la virtualisation (VM, stockage, ...), pour la gestion des utilisateurs du Cloud et des machines virtuelles, rendant plus facile la maintenance et l'utilisation du logiciel[17].

10. WS-Security(Web Services Security): est un protocole de communications qui permet d'appliquer de la sécurité aux services web.

2.2 La solution OpenNebula

2.2.1 Présentation

OpenNebula est un projet Open Source de Cloud Computing de type IaaS pour construire des nuages privés, public ou hybride. OpenNebula est soutenu par le projet européen " Reservoir ", qui propose une architecture complète pour la gestion de data-centers et la création de services Cloud[10].

Le projet a été lancé en 2005. Ecrit en C++, Ruby ¹¹et Shell, la première version stable est sortie en 2008, sous licence Apache2.0 ¹². Il a pour but la gestion de machines virtuelles à grande échelle sur des infrastructures distribuées ou de cluster, et supporte plusieurs technologies d'hyperviseur : Xen, KVM et VMware[10].

OpenNebula permet aussi de combiner les infrastructures locales et publiques (elle permet de s'interfacer avec le Cloud d'Amazon EC2)[16].

OpenNebula gère le stockage, le réseau et les technologies de virtualisation afin de permettre la mise en place dynamique de services multiniveaux (des groupes de machines virtuelles interconnectées) sur les infrastructures distribuées[13].

OpenNebula est intégré dans Debian Sid, Ubuntu Natty et OpenSuse. La dernière version étant OpenNebula 3.8[16].

2.2.2 Architecture OpenNebula

OpenNebula est un gestionnaire qui permet de centraliser l'infrastructure virtuelle. OpenNebula adopte une architecture classique dite de " cluster " avec un frontend et un ou plusieurs noeuds qui exécutent et hébergent les machines virtuelles, avec un réseau physique reliant le frontend aux noeuds.

11. Ruby: est un langage de programmation libre. Il est interprété, orienté objet.

12. Apache2.0: la licence Apache est une licence de logiciel libre et open source.

1. Front-End:

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

- **Démon OpenNebula (ONED):** il permet de gérer tous les services du cloud et d'orchestrer les opérations de tous les modules:
 - Il gère le cycle de vie des VMs ainsi que le fonctionnement de l'hyperviseur.
 - Gère les réseaux virtuels.
 - Déploie des machines virtuelles, selon la politique d'attribution, il décide l'emplacement du lancement des machines virtuelles.
 - Gère les images des machines virtuelles et leur stockage.
 - Supervise l'état des machines virtuelles lancées et leur consommation.
- **Drivers (pilotes):** programmes utilisés par le processeur pour l'interfaçage avec un sous-système spécifique, par exemple un système de stockage de fichiers. Il existe trois (03) types de drivers:
 - **Pilotes de virtualisation(VMM):** interfaçage avec les hyperviseurs (KVM, VMware ou Xen).
 - **Pilotes de transfert(TM):** interfaçage avec le système de stockage des images.
 - **Pilotes d'informations(IM):** utilisés pour surveiller les noeuds du cluster.

2. Nodes (noeuds):

Les noeuds fournissent les ressources nécessaires pour les machines virtuelles, ce sont les machines physiques qui hébergent (exécutent) les machines virtuelles, c'est à dire des serveurs hôtes de virtualisation. Chaque noeud dispose des trois éléments suivants:

- **un hyperviseur:** il permet la virtualisation de plusieurs machines virtuelles sur une seule machine physique.

- **un bridge**: il permet de relier les interfaces réseaux virtuelles des VMs à l'interface réseaux physique de la machine hôte.
- **un serveur ssh**: OpenNebula utilise ssh pour copier les images des machines virtuelles.

3. Image repository (Référentiel d'images)

Tout support de stockage qui contient les images de base des machines virtuelles. En général, il s'agit d'un répertoire de fichiers NFS ¹³ à qui l'administrateur et l'utilisateur peuvent accéder[10][14].

La Figure.2.3 représente l'architecture générale d'OpenNebula.

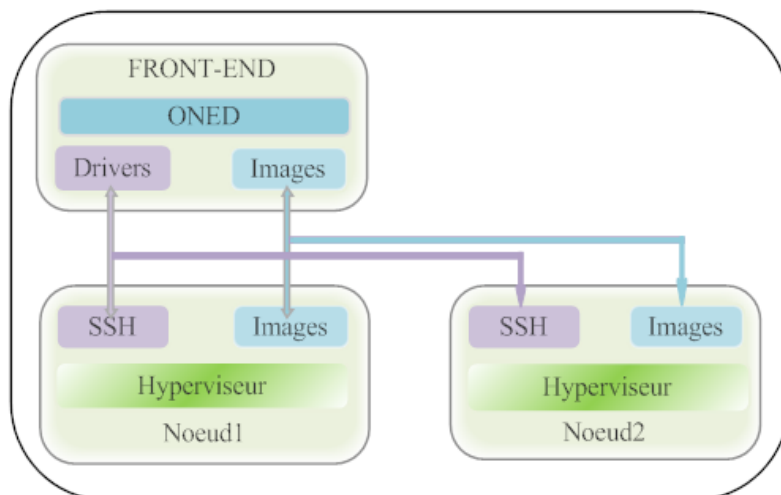


FIG. 2.3 – Architecture OpenNebula

[14]

2.2.3 Gestion des images

OpenNebula ne propose pas de solution prête à l'emploi, mais permet à l'administrateur d'utiliser le système de fichier de son choix (par exemple NFS) sur le système de stockage de son choix (par exemple un NAS ¹⁴). Pour effectuer les transferts entre le serveur de fichiers et les machines hôtes, l'administrateur aura ensuite la liberté soit

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

14. NAS(Network Attached Storage): est un serveur de fichier autonome, relié à un réseau dont la principale fonction est le stockage de données en un volume centralisé.

d'installer sur ces dernières le pilote correspondant au système de fichier, soit d'utiliser un transfert SSH[12].

OpenNebula distingue plusieurs instances de stockage en leur attribuant différents rôles:

- **Les Filesystem Datastore:** ont la responsabilité du stockage des images "template". Il peut exister plusieurs Filesystem Datastore, par exemple en vue d'une duplication dans chaque cluster afin d'éviter un engorgement, ou bien pour grouper des images d'un même type et appliquer une politique de qualité de service différenciée (par exemple un serveur plus puissant pour les images Xen...).
- **Le System Datastore:** il est chargé de maintenir une copie des images des VMs en cours d'exécution. Ce dernier permet la migration live d'une VM de son hôte initial vers un nouvel hôte, c'est à dire la relocalisation d'une machine virtuelle d'une machine physique sur une autre sans interruption pour l'utilisateur[12].

2.2.4 Réseau et adressage

Contrairement à Eucalyptus, OpenNebula ne gère pas de serveur DHCP en interne. Il propose alors deux (02) modes:

- Un mode où il ne gère que les adresses MAC, et il faut alors dans ce cas faire intervenir un serveur DHCP externe permettant l'attribution d'adresses IP aux VMs.
- Un mode où il propose une règle simple d'attribution de l'IP à partir de l'adresse MAC, et cette attribution se fait entre l'hyperviseur et la VM.

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

Plusieurs drivers sont proposés au sein d'OpenNebula :

- **fw:** permet l'utilisation de règles simple de pare-feu .
- **802.1Q, VMware, ovswitch, ebttables:** technologies diverses permettant la gestion des ponts entre la connexion physique de la machine hôte et les NICs ¹⁵ virtuels

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

des VMs. Ces drivers participent entre autre à la mise en place de réseaux virtuels.

Notons que tous les drivers ne sont pas compatibles avec tous les hyperviseurs (le driver VMware étant par exemple spécifique à l'hyperviseur du même nom)[12].

2.2.5 Sécurité

Les mécanismes d'authentification et d'autorisation avec OpenNebula sont basés sur les mots de passe, paires de clés ssh rsa, certificats X509¹⁶, LDAP¹⁷ ou Active Directory¹⁸[16].

OpenNebula propose ainsi une fonctionnalité de gestion de différents VLANs. Cependant, OpenNebula est souvent décrié par rapport à ses concurrents comme étant peu sécurisé et facilement hackable. Cela est probablement dû à la haute flexibilité de la plateforme: elle permet en effet de mettre en place de nombreux procédés visant à sécuriser son utilisation (tunnels de transferts, authentifications encryptées, séparation des VLANs...) mais cela n'est pas configuré ainsi par défaut. Il faut ainsi une certaine expertise pour pouvoir se protéger d'éventuelles attaques. Notons que le transfert des images en clair entre le serveur de fichiers NFS et la machine hôte est l'une des faiblesses les plus mentionnées[12].

2.2.6 Stockage

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

- **Banques de données du système de fichiers:** pour stocker des images disque dans des systèmes de fichiers partagés (NFS, GlusterFS¹⁹, ...).

réseau informatique.

16. Certificats X.509: c'est un fichier texte, transmis par l'autorité de certification sur l'ordinateur de client. Il contient la clé publique de l'utilisateur et la signature de la CA ainsi que différents champs normalisés.

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

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

19. GlusterFS: un système de fichiers libre distribué en parallèle, qui permet de stocker jusqu'à plusieurs pétaoctets (10^{15} octets).

- **ISCSI** ²⁰ /**LVM**: pour stocker des images disque sous une forme périphérique bloc.
- **VMware**: banque de données spécialisée pour l'hyperviseur VMware qui gère le format VMDK ²¹[15].

2.2.7 Les interfaces fournies

OpenNebula fournit de nombreuses interfaces différentes qui peuvent être utilisées pour interagir avec les fonctionnalités offertes pour gérer les ressources physiques et virtuelles. Il existe différentes perspectives permettant d'interagir avec OpenNebula:

- Interfaces Cloud Computing pour les consommateurs Cloud, comme le CICO et EC2 Query et EBS interfaces.
- Les interfaces d'administration pour les utilisateurs avancés et opérateurs de Cloud, comme une interface de ligne de commande et l'interface graphique Sunstone[16].

2.3 La solution OpenStack

2.3.1 Présentation

OpenStack est un projet de Cloud Computing privé, public et hybride sous licence Apache2.0, écrit en Python. la NASA ²² a développé spécialement la partie calcul d'OpenStack et Rackspace Cloud ²³ a contribué à la partie stockage. à partir de juillet 2010, ces deux sociétés ont ensuite été rejointes par Cloud.com, Canonical, Cisco et autre pour le développement d'OpenStack[10].

OpenStack est architecturé pour fournir la flexibilité pour concevoir un nuage, sans matériel propriétaire ou logiciel requis et la capacité d'intégrer avec les systèmes

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

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

22. NASA: Organisme du gouvernement qui prend en charge la majeure partie du programme spatial civil des Etats-Unis.

23. Rackspace Cloud: un fournisseur de plate-forme de cloud computing.

existants et des technologies tierces. Il est conçu pour gérer et automatiser des pools de ressources de calcul et peut travailler avec des technologies de virtualisation largement disponibles, ainsi le calcul haute performance (HPC)[18].

OpenStack est packagé dans Ubuntu Server et dans Debian. D'autres distributions telles que RedHat Enterprise Linux et CentOS sont maintenant supportées.

OpenStack supporte la plupart des solutions de virtualisation du marché: ESX, Hyper-V, KVM, LXC, QEMU, UML, Xen et XenServer[23].

2.3.2 Architecture OpenStack

Il existe actuellement sept (07) composants essentiels d'OpenStack: Nova, Swift, Glance, Keystone, Quantum, Cinder et Dashboard.

1. OpenStack Compute(Nova)

Compute sert à la gestion de larges réseaux de machines virtuelles. Il fournit une interface d'administration et l'API nécessaire à l'orchestration du Cloud qui permet:

- La gestion des instances serveurs.
- La gestion du réseau.
- Le contrôle d'accès.

Compute ne nécessite pas de prérequis matériel et est totalement indépendant de l'hyperviseur. C'est l'équivalent d'Amazon Elastic Compute Cloud (EC2)[18].

2. OpenStack Object Storage (Swift)

Object Storage sert à la création d'espace de stockage redondant pour le stockage de plusieurs pétaoctet de données. Il ne s'agit pas réellement d'un système de fichier traditionnel, mais plutôt un système de stockage distribué conçu pour le stockage à long terme de gros volumes de données telles que les images de machines virtuelles, stockage photo, stockage e-mail, etc. Son

architecture distribuée offre plusieurs points d'accès pour éviter les SPOF²⁴. C'est l'équivalent d'Amazon Simple Storage Service (S3)[24].

3. OpenStack Imaging Service (Glance)

Fournit les services de stockage, de découverte, d'enregistrement et de distribution pour les images disques de machines virtuelles. Il fournit également une API compatible REST permettant d'effectuer des requêtes pour obtenir des informations sur les images hébergées par les différents magasins de stockages. Glance offre les mêmes caractéristiques que le catalogue d'Amazon AMI[25].

4. OpenStack Identity Service (Keystone)

Assure l'authentification et l'autorisation pour tous les services OpenStack. Il fournit également un catalogue de services au sein du nuage. Il supporte de multiples formes d'authentification, y compris le standard de nom et mot de passe, les systèmes à base de jetons[32][33].

5. OpenStack Networking (Quantum)

Dispose de "la connectivité réseau en tant que service" entre les dispositifs d'interface gérés par les services d'OpenStack. Le service fonctionne en permettant aux utilisateurs de créer leurs propres réseaux. Ainsi le réseau OpenStack dispose d'un cadre permettant l'extension des services réseau supplémentaires, tels que les systèmes de détection d'intrusion (IDS), équilibrage de charge, pare-feu et des réseaux privés virtuels (VPN) pour permettre une meilleure sécurité[34].

6. Block Storage (Cinder)

Fournit un stockage persistant par bloc de machines virtuelles. Il est approprié pour des scénarios de rendement sensibles telles que le stockage de base de données et des instances en cours d'exécution (snapshots) dans Nova. Cinder fournit des services similaires à blocs EBS d'Amazon[32].

24. SPOF (Single Point Of Failure): est un point d'un système informatique dont le reste du système est dépendant et dont une panne entraîne l'arrêt complet du système.

7. Horizon (Dashboard)

Fournit une interface utilisateur modulaire basé sur le Web pour tous les services OpenStack. Cette interface graphique Web, permet aux administrateurs de nuages et aux utilisateurs de contrôler leur infrastructure (instances, stockage et les ressources réseau)[30].

En tant qu'administrateur de nuage, le Dashboard fournit une vue d'ensemble sur la taille et l'état du nuage. Il permet de créer des utilisateurs et des projets, affecter des utilisateurs à des projets et fixer des limites sur les ressources pour ces projets (quotas), en revanche il fournit aux utilisateurs un portail self-service et met a leur disposition leur propres ressources dans les limites fixées par les administrateurs[30].

Le projet OpenStack est conçu pour fournir un nuage hautement évolutif. Pour ce faire, chacun des services constitutifs sont conçus pour fonctionner ensemble pour fournir une infrastructure complète en tant que service (IaaS). Cette intégration est facilitée par des interfaces de programmation d'applications publiques (API) qui permettent à chacun de ces services d'utiliser un autre service, ce sont les mêmes API qui sont disponibles pour les utilisateurs finaux du nuage[19].

2.3.3 La relation entre les services d'OpenStack

La figure ci-dessous nous montre les différents services ainsi que la relation entre eux:

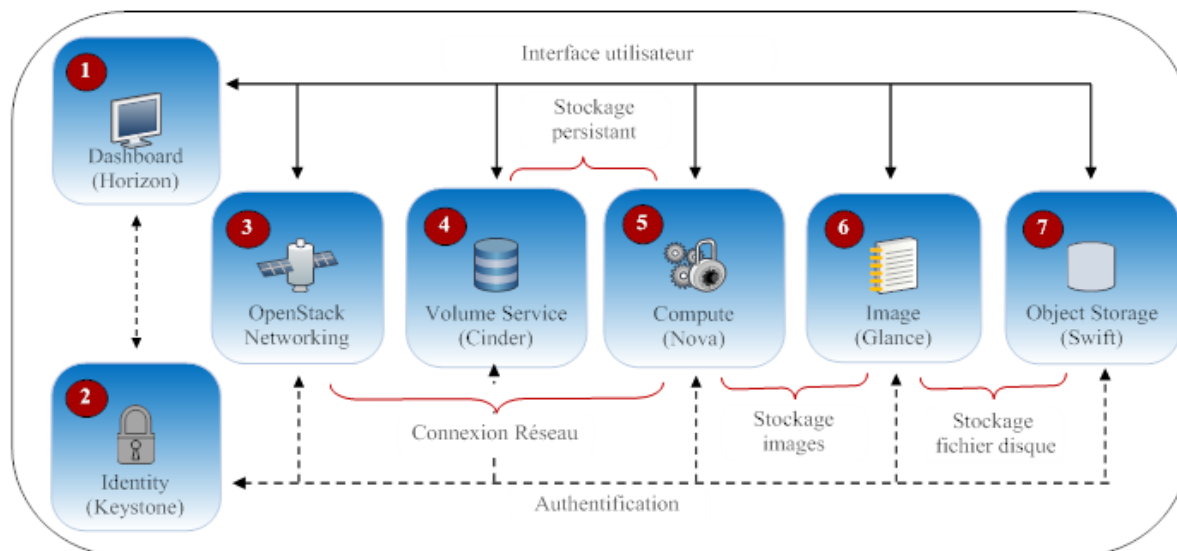


FIG. 2.4 – La relation entre les composants d'OpenStack.

[32]

- Dashboard fournit une interface web pour les services OpenStack.
- Nova stocke et récupère des disques virtuels (images) et les métadonnées associées à l'image Glance.
- Quantum fournit un réseau virtuel de Calcul.
- Cinder fournit des volumes de stockage pour Compute.
- Glance permet de stocker les fichiers des disques virtuels dans le magasin d'objets Swift.
- Tous les services doivent s'authentifier auprès de Keystone.

2.3.4 Gestion des images

OpenStack Compute abstrait la gestion des images en deux couches différentes:

- **Une Image Store:** mettant à disposition une API pour les recherches et appels d'image, leur transfert et leur stockage.
- **Un Object Store:** contenant les fichiers des images proprement dits, mais dont le rôle est générique et pouvant donc stocker d'autres données (des fichiers d'archives par exemple).

Ce découplage permet à l'Image Store, qui est interne à la plateforme OpenStack Compute, d'utiliser différents Object Store par défaut OpenStack Swift, mais on

peut également stocker les fichiers images sur Amazon S3 en utilisant un serveur HTTP ou encore d'autres serveurs de fichiers plus traditionnels.

Lorsqu'on utilise Swift, on bénéficie de son architecture distribuée qui remplit deux (02) objectifs:

- Prévenir une panne du système de stockage en cas de défaillance d'un serveur, la distribution permet d'avoir des serveurs de secours.
- Augmenter aisément l'échelle du parc de machines en ajoutant de nouveaux points de stockages permettant d'alléger la charge des points existants et de garantir le fonctionnement de la plateforme[12].

2.3.5 Réseau et adressage

Le fonctionnement d'OpenStack Compute se rapproche de celui d'Eucalyptus: la configuration réseau est séparée en trois (03) modes différents :

- **Le mode VLAN Network:** entièrement géré par OpenStack Compute, permet l'attribution d'IPs virtuelles privées aux nouvelles VMs. Dans ce mode, un serveur DHCP est démarré pour chaque VLAN et est chargé d'attribuer les adresses IP virtuelles.
- **Le mode flat DHCP:** se rapproche du mode SYSTEM d'Eucalyptus, un serveur DHCP externe intervient afin d'attribuer dynamiquement des IPs aux machines virtuelles, la notion de VLANs n'existe pas.
- **Le mode flat:** se rapproche du mode STATIC d'Eucalyptus. Dans ce mode, la correspondance IP/MAC est statique. La seule différence se situe dans le fait que cette correspondance intervient hors de la plateforme OpenStack, c'est l'administrateur réseau qui doit configurer manuellement les ponts des machines hôtes avec le réseau physique, la notion de groupes de sécurité n'existe pas[13].

2.3.6 Sécurité

OpenStack propose une authentification utilisateur par mot de passe, l'accès à tous les services OpenStack se fait par l'authentification et l'autorisation assuré par

le composant " Keystone "[23].

OpenStack propose aussi des VLANs permettant d'isoler le trafic réseau des VMs selon des groupes sécurisés d'utilisateurs[12].

Ainsi la création des groupes de sécurité qui fournit un moyen d'isoler le trafic entrant et sortant de machines virtuelles. Chaque groupe de sécurité peut avoir plusieurs règles qui lui sont associés, en spécifiant l'adresse IP source, type de protocole, les ports de destination, etc. Tous les paquets correspondant à ces paramètres spécifiés dans une règle sera autorisé à entrer, le reste des paquets seront bloqués[37].

2.3.7 Stockage

OpenStack à un support pour le stockage d'objets (swif) à la fois et de stockage au niveau bloc (cinder)[23].

OpenStack Object Storage n'est pas utilisé comme un disque dur traditionnel. Les objets et les fichiers sont écrits sur plusieurs disques durs répartis dans les serveurs du centre de données, Le stockage est évolutif horizontalement en ajoutant simplement de nouveaux noeuds. Ainsi les dispositifs de stockage persistent jusqu'à leurs suppression[20][24].

Conclusion

Dans cette partie nous avons étudié trois (03) solutions Open Source: Eucalyptus, OpenNebula et OpenStack, qui permettent la mise en place d'une solution de Cloud Computing de type Iaas. Ces solutions répondent à des besoins différents donc il est bon de savoir ce que l'on veut faire avant d'en choisir une pour ne pas perdre trop en performance.

CHAPITRE 3

La comparaison et le choix entre les solutions Cloud Computing

Introduction

Dans ce chapitre nous allons établir un comparatif entre les trois solutions étudiées précédemment et détailler la solution retenue pour pouvoir la mettre en place.

3.1 Tableau comparatif des solutions Cloud Computing (IaaS)

	Eucalyptus	OpenNebula	OpenStack
Type de Cloud	<ul style="list-style-type: none"> • Privé, public et hybride. 	<ul style="list-style-type: none"> • Privé, public et hybride. 	<ul style="list-style-type: none"> • Privé, public et hybride.
Hypirviseurs supportés	<ul style="list-style-type: none"> • Xen, KVM, VMware (pour les solutions non-open source). 	<ul style="list-style-type: none"> • Xen, KVM et VMware. 	<ul style="list-style-type: none"> • ESX, Hyper-V, KVM, LXC, QEMU, UML, Xen et XenServer.
Stockage	<p>Très bien.</p> <ul style="list-style-type: none"> • Contrôleur principal Walrus compatible avec les API S3 d'Amazon • Des contrôleurs de stockage sur chaque noeud. 	<ul style="list-style-type: none"> • un système de fichier partagé(par défaut NFS) mais qui est d'utilisation très centralisée,ce qui crée rapidement un embouteillage pour l'accès aux données. 	<p>Bonne solution de stockage.</p> <ul style="list-style-type: none"> • Comporte un Objet de stockage (Swift) qui sert à la création d'espace de stockage redondant. • Espace de stockage partagé et évolutif.
Migration à chaud	Non	Oui	Oui

	Eucalyptus	OpenNebula	OpenStack
Lancement d'une machine virtuelle	<ul style="list-style-type: none"> • Demande beaucoup de commandes pour le lancement. 	<ul style="list-style-type: none"> • Seulement en deux commandes. 	<ul style="list-style-type: none"> • Seulement en deux commandes.
Installation	<ul style="list-style-type: none"> • Dépend de l'environnement réseau, difficulté en environnement hétérogène. 	<ul style="list-style-type: none"> • Installation facile, simple et très rapide. 	<ul style="list-style-type: none"> • Installation automatisée et documentée.
Configuration	<ul style="list-style-type: none"> • La solution est facile à mettre en oeuvre sur une architecture classique en revanche, sur une architecture complexe, le déploiement devient difficile. 	<p>Moyenne</p> <ul style="list-style-type: none"> • Les fonctions de base sont simples à utiliser, le passage à un environnement complexe l'est moins • Nécessité d'une grande maîtrise de la part de l'administrateur. 	<p>Moyenne</p> <ul style="list-style-type: none"> • Possibilité de configuration par des scripts d'installation.

	Eucalyptus	OpenNebula	OpenStack
Documentation	<ul style="list-style-type: none"> • Complète mais pas toujours à jour. 	<p>Complète</p> <ul style="list-style-type: none"> • Documentations officielle à la portée. 	<p>Bonne</p> <ul style="list-style-type: none"> • Documentation officielle disponible.
Sécurité	<p>Bien.</p> <ul style="list-style-type: none"> • Chaque contrôleur est authentifié par clé SSH et des fichiers de permission pour authentifier toutes les transactions. • Isolation du trafic réseau avec le mode réseau MANAGED. • Communication sécurisé par l'encryption des requêtes SOAP via WS- Security. 	<p>Peu sécurisé.</p> <ul style="list-style-type: none"> • malgré l'utilisation des certificats X509, authentifications, VLANs...Ces éléments ne sont pas configurés par défaut. • Le transfert des images est en clair entre le serveur de fichiers NFS et la machine hôte. 	<p>Très bien</p> <ul style="list-style-type: none"> • Accès par authentification assuré par le composant " Keystone ". • Isolation du trafic réseau (VLANs, groupes de sécurités).

	Eucalyptus	OpenNebula	OpenStack
Réseau	<ul style="list-style-type: none"> • Eucalyptus offre quatre (04) modes réseaux avec des différences du niveau de sécurité et du mode d'emploi. • Limité pour un déploiement sur un réseau complexe ou d'une grande taille. 	<ul style="list-style-type: none"> • OpenNebula ne gère pas de serveur DHCP en interne. • Configuration manuelle difficile. 	<ul style="list-style-type: none"> • OpenStack peut s'installer sur des réseaux complexe.

TAB. 3.1 – Tableau comparatif des solutions Cloud Computing.

Concernant les différents services fournies, les trois solutions Open Source étudiées: Eucalyptus, OpenNebula et OpenStack proposent presque les mêmes fonctionnalités concernant la gestion des VMs, la gestion des utilisateurs etc, avec des différences au niveau sécurité, stockage, mode de réseautage et autre. Avec chacune ses points forts et faibles.

OpenStack est la solution la plus utilisée par rapport à Eucalyptus et OpenNebula qui sont délaissées actuellement, elle est en forte évolution et entourée d'une forte communauté[25].

OpenStack possède une architecture modulaire, car elle est composée de services complémentaires pour assurer le bon fonctionnement d'une infrastructure. Elle offre

des fonctionnalités avancées au niveau sécurité et réseau par rapport à Eucalyptus et OpenNebula.

OpenStack s'avère bonne en terme de simplicité et permet une bonne supervision qui est un éléments crucial pour le controle de l'état de notre infrastructure et du fonctionnement des VMs et ceci par le Dashboard qui est riche en fonctionnalités.

Après l'étude qui a été faite sur les trois solutions et le tableau coparatif établi, nous avons opté pour l'installation d'OpenStack. Après ce choix nous allons aborder quelques concepts importants sur cette solution.

3.2 Performance et évolutivité

OpenStack possède une très engagée communauté de plus de 6700 personnes et plus de 190 entreprises, y compris Rackspace, Dell, HP, IBM et RedHat.

OpenStack dans sa sixième version " Folsom " est le résultat d'un développement continu, contient des améliorations continue grâce à l'enrichissement permanent de la communauté OpenStack qui devient plus grande à chaque nouvelle version.

L'analyse fournie par le site officiel d'OpenStack montre que plusieurs centaines de personnes issues de 49 sociétés différentes ont contribué au code au cours des sept projets principaux de Folsom. La version Folsom du projet OpenStack inclut 185 nouvelles fonctions[25][26][27].

Chaque version d'OpenStack à un nom, dans l'ordre alphabétique croissant (par exemple, Grizzly suit Folsom), les noms de versions et leurs dates de sortie sont indiqués dans le tableau ci-dessous:

Nom de version	Date de sortie	Composants
Havana	17 October 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Grizzly	4 April 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Folsom	27 September 2012	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Essex	5 April 2012	Nova, Glance, Swift, Horizon, Keystone
Diablo	22 September 2011	Nova, Glance, Swift
Cactus	15 April 2011	Nova, Glance, Swift
Bexar	3 February 2011	Nova, Glance, Swift
Austin	21 October 2010	Nova, Swift

TAB. 3.2 – Les versions d’OpenStack.

En commençant par la libération Cactus, OpenStack a adopté un calendrier de sortie de six mois. La libération Havane est prévue pour Octobre 2013.

Les nouveautés de la version OpenStack Folsom réside dans les deux nouveaux projets notables: Quantum et Cinder.

- Quantum est le projet réseau qui était connu sous le nom nova-network, offre aux utilisateurs d’OpenStack la capacité de contrôler le réseau de leurs nuage; Quantum apporte un vrai multi-tenant, sans aucune restriction de VLAN.
- Cinder est le projet de stockage OpenStack bloc qui était autrefois connu sous le nom nova-volume. Cette reconnaissance dans son propre projet permet à ce composant essentiel des fonctionnalités de base pour se développer indépendamment de Nova (Compute). Ce qui inclu de nouvelles fonctionna-

litées et des corrections de bugs afin de répondre aux besoins d'hébergement d'aujourd'hui.

Mettre en place tous les éléments de Folsom ensemble permet d'avoir une suite logicielle Cloud qui permet d'évoluer facilement, sur un environnement de calcul et de stockage objet hautement disponible au sein d'un centre de données, ce qui permet aussi de redimensionner les machines virtuelles ou réallouer les ressources en adaptant rapidement l'infrastructure aux besoins des utilisateurs[25][26][27].

3.3 Recomendation matérielle

La mise en place de la solution OpenStack n'est possible que sur des machines avec un processeur doté des jeux d'instructions de virtualisation matérielle AMD-V ou Intel-VT[22].

3.4 Avantages et inconvénients

3.4.1 Avantages

- OpenStack est un projet Open Source et gratuit: le code source du projet est libre, la modification et la distribution du code est autorisé sous toute forme (libre ou propriétaire, gratuit ou commercial).
- Utilisée par la plupart des développeurs dans le monde, et l'un des plus points forts du projet est sa communauté grandissante.
- Elle consiste en une architecture modulaire, architecture enfichable et en constante évolution, elle permet un ensemble d'outils qui permet une intégration dans les infrastructures d'autres fournisseurs de services.

3.4.2 Inconvénients

- Des fonctions intéressantes du projet OpenStack sont peu documentées, ce qui fait de cette dernière une solution puissante mais parfois délicate à appréhender et requiert une certaine expertise.

Conclusion

Dans ce chapitre nous avons établi un tableau comparatif entre trois (03) solutions de Cloud Computing Open Source. Après cette comparaison nous avons choisi la solution openStack, la suite de ce chapitre consistait à détailler la solution retenue, définir ses avantages et ses limites ainsi d'autres points importants pour la bonne compréhension de cette dernière.

Dans ce qui suit, nous allons aborder les étapes d'installation d'OpenStack et la configuration de ses différents composants.

CHAPITRE 4

Installation et administration de la solution OpenStack

Introduction

Nous nous intéressons dans ce chapitre, aux principales étapes d'installation d'OpenStack Folsom.

Nous allons commencer par la préparation du système, sur lequel sera implémentée cette solution, et ceci par la configuration réseau, ainsi l'installation des différents paquets requis pour le bon fonctionnement des différents services et composants.

Ensuite on passe à l'installation et la configuration de Keystone, Glance, Nova etc. Nous finalisons par la mise en oeuvre du Dashboard qui est l'interface graphique et l'outil de gestion d'OpenStack.

4.1 Installation d'OpenStack

L'installation d'OpenStack peut se faire de deux (2) manières:

- **Single Node ("All in One") Déploiements:** dans cette configuration, tous les services sont installés et exécutés sur un système unique.

Ce type de déploiement est adapté à des fins d'évaluation. Un tel déploiement n'est cependant pas adapté pour une utilisation dans un environnement de production.

- **Multi Node:** les composants d'OpenStack seront installés sur des noeuds différents.

Dans notre cas nous allons installer OpenStack sur une seule machine.

OpenStack propose aussi plusieurs types d'installation:

- installation via des scripts.
- installation via les packages.

L'installations via des scripts présente l'avantage de pouvoir installer les composants rapidement mais il n'y a aucune explication sur ce que les scripts installent et sur la configuration [10].

Pour mieux comprendre l'installation des différents composants nous avons donc choisi une installation manuelle via les packages.

OpenStack a la capacité de s'installer sur une multitude de machines physiques, de l'ordinateur portable au serveurs d'entreprise, tout est possible[38]. Dans notre projet l'installation se fera sur un ordinateur portable ayant la configuration suivante:

- **Processeur:** Intel(R) Core(TM) i5 CPU M540 @ 2.53GHz 64-bit.
- **Mémoire:** 4Go de RAM.
- **Espace disque:** 400Go.

4.1.1 Outils à utiliser pour l'installation d'openstack

Dans cette partie nous allons citer les différents outils logiciels utilisés pour l'installation d'OpenStack.

1. VirtualBox

Au lieu d'installer OpenStack directement sur la machine hôte, le logiciel Virtual-Box sera utilisé pour créer une machine virtuelle, sur laquelle seront installés les composants d'OpenStack, et cela pour créer le nombre d'interfaces désirées, l'installation d'OpenStack sur une machine virtuelle a aussi pour objectif d'isoler la machine

hôte et la machine virtuelle pour ne pas affecter le reste de notre environnement de travail.

Cette machine virtuelle sera configurée avec 2 Go de RAM et 20 Go de disque dur, sur la quelle sera installée la distribution Ubuntu 12.04 LTS et possèdera trois interfaces réseau :

- La première sera une interface NAT qui permet à notre machine virtuelle de se connecter au réseau en dehors de VirtualBox pour télécharger les paquets.
- La deuxième interface qui sera l'interface publique de notre machine sur la quelle seront installés les composants d'OpenStack .
- La troisième interface sera utilisée pour notre réseau privé, utilisé par OpenStack Compute pour la communication interne entre les différentes machines virtuelles .

2. Ubuntu 12.04 LTS Nous avons choisi d'installer la version 12.04 LTS d'ubuntu car cette dernière est la plus simple à utiliser avec OpenStack et ne souffre pas de problèmes de compatibilité par rapport aux autres versions.

3. version d'OpenStack Nous avons choisis de configurer la version Folsom d'OpenStack décrite précédemment dans le troisième chapitre. Notre choix s'est porté sur cette version car c'est la dernière version jusqu'à la réalisation de ce projet, qui est améliorée de fonctionnalités par rapport aux versions qui la précède.

4.2 Installation des paquets de base et préparation du système

Après avoir terminé l'installation de la distribution Ubuntu 12.04 LTS, nous passons en tant qu'utilisateur root jusqu'à la fin de l'installation:

```
openstack-VirtualBox: sudo su
```

4.2.1 La mise à jour du système

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

```
root@openstack-VirtualBox:/home/openstack# apt-get update
root@openstack-VirtualBox:/home/openstack# apt-get upgrade
root@openstack-VirtualBox:/home/openstack# apt-get dist-upgrade
```

4.2.2 Installation des bridges

Le bridge est nécessaire pour configurer les interfaces Ethernet en mode pont et permet le partage de connexion entre ces interfaces, il permet de relier une interface réseau physique à une interface réseau virtuelle.[10].

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

```
root@openstack-VirtualBox:/home/openstack# apt-get install bridge-utils
```

4.2.3 Configuration réseau

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

```
#L'interface réseau loopback
auto lo
iface lo inet loopback
#L'interface réseau primaire
auto eth0
iface eth0 inet dhcp
#L'interface public
auto eth1
iface eth1 inet static
address 172.16.0.1
netmask 255.255.0.0
network 172.16.0.0
broadcast 172.16.255.255
#L'interface privée
auto br100
iface br100 inet manual
bridge_ports eth2
bridge_stp off
bridge_maxwait 0
bridge_fd 0
up ifconfig eth2 up
```

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

```
root@openstack-VirtualBox:/home/openstack# /etc/init.d/networking restart
```

4.2.4 Ajout du repository Ubuntu Cloud Archive

La version 12.04 d'Ubuntu ne contient pas les paquets OpenStack Folsom par défaut. Canonical fournit un repository où se trouvent ces paquets.

Les commande ci-dessous permettent l'ajout du repository Ubuntu Cloud Archive:


```
root@openstack-VirtualBox:/home/openstack# echo deb http://ubuntu-
cloud.archive.canonical.com/ubuntu precise-updates/folsom main; /
etc/apt/source.list.d/folsom.list
root@openstack-VirtualBox:/home/openstack# apt-get install ubuntu-cloud-keyring
```

4.2.5 Installation et Configuration de MySQL

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

```
root@openstack-VirtualBox:/home/openstack# apt-get install mysql-server python-
mysqldb
```

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

```
root@openstack-VirtualBox:/home/openstack# sed -i 's/127.0.0.1/0.0.0.0/g'
/etc/mysql/my.cnf
```

Ensuite, nous redémarrons le serveur MySQL:

```
root@openstack-VirtualBox:/home/openstack# service mysql restart
```

4.2.6 Installation et configuration du serveur NTP

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

```
root@openstack-VirtualBox:/home/openstack# apt-get install ntp
```

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

```
root@openstack-VirtualBox:/home/openstack# sed -i 's/server ntp.ubuntu.com/server
ntp.ubuntu.com \nserver 127.127.1.0 \nfudge 127.127.1.0 stratum 10/g' /etc/ntp.conf
```

Une fois la configuration terminée nous redémarrons le service:

```
root@openstack-VirtualBox:/home/openstack# service ntp restart
```

4.2.7 Installation de RabbitMQ

RabbitMQ est un courtier de messages se basant sur le standard AMQP afin d'échanger les informations avec différents clients. En bref c'est le service qui permet la communication entre les composants d'OpenStack[33].

```
root@openstack-VirtualBox:/home/openstack# apt-get install rabbitmq-server
```

4.3 Installation et configuration des composants OpenStack

Pour la mise en place d'OpenStack, nous allons opter à l'installation des composants: Keystone, Glance, Nova et Dashboard, tel que nous n'allons pas procéder à l'installation de Swift, Quantum et Cinder car la configuration de ces derniers complexifie beaucoup le temps de la mise en place, et nécessite du matériels performants.

De plus, les fonctionnalités avancées de Quantum de Swift et de Cinder ne sont pas vraiment utiles lors de l'installation sur un seul serveur et pour une installation de test. Pour cela nous avons opté à:

- Installer et configurer nova-network qui prend en charge le comportement de Quantum mais avec moins de fonctionnalités.
- La configuration de nova-volume au lieu du composant Cinder.
- Enfin, l'utilisation de nova-objectstore qui à un rôle similaire au composant Swift.

4.3.1 Installation et configuration de Keystone

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

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

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

2. **Gestion des services et points d'accès:**

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

Nous commençons par l'installation du paquet keystone:

```
root@openstack-VirtualBox:/home/openstack# apt-get install keystone
```

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

Création de la base de donnée MySQL

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

Lancer la console MySQL par la commande:

```
root@openstack-VirtualBox:/home/openstack# mysql -u root -p
```

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

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

```
mysql > CREATE DATABASE keystone;
```

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

```
mysql > GRANT ALL ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'openstack';
mysql > FLUSH PRIVILEGES;
mysql > Quit;
```

Et nous modifions le fichier `/etc/keystone/keystone.conf` comme indiqué ci-dessous: Nous allons décommenter les huit première ligne et nous allons rajouter la ligne "connection = mysql://keystone:openstack@172.16.0.1:3306/keystone" pour créer une connexion à la base de données "keystone".

```
[DEFAULT]
bind_host = 0.0.0.0
public_port = 5000
admin_port = 35357
# Mot de passe d'administration
admin_token = ADMIN
compute_port = 8774
verbose = True
debug = True
log_config = /etc/keystone/logging.conf
[sql]
connection = mysql://keystone:openstack@172.16.0.1:3306/keystone
idle_timeout = 200
```

Ensuite, nous redémarrons keystone:

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

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

```
root@openstack-VirtualBox:/home/openstack# keystone-manage db_sync
```

La création des rôles

Les rôles pouvant être attribués à un utilisateur sont:

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

Commençons par l'installation du paquet suivant:

```
root@openstack-VirtualBox:/home/openstack# apt-get install python-keystoneclient
```

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

```
root@openstack-VirtualBox:/home/openstack# keystone --username admin --password ADMIN --tenant_name admin --auth_url http://localhost:5000/v2.0
```

Il est possible d'utiliser des variables d'environnement pour éviter de ressaisir tous les arguments à chaque commande.

Pour éviter de refaire un export des variables à chaque ouverture de terminal, nous pouvons les exporter automatiquement, il suffit de créer un fichier **.openrc** dans notre dossier personnel:

```
root@openstack-VirtualBox:/home/openstack# gedit .openrc
```

Et d'ajouter les lignes suivantes:

```
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=ADMIN
export OS_AUTH_URL="http://172.16.0.1:5000/v2.0/"
```

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

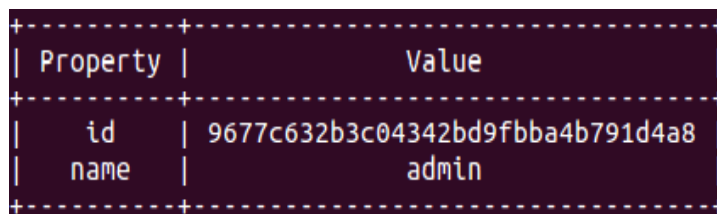
```
root@openstack-VirtualBox:/home/openstack# echo "source ~/ .openrc" .bashrc
```

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

1. La création du rôle admin

```
root@openstack-VirtualBox:/home/openstack# keystone role-create --name admin
```

L'affichage suivant été en résultat:

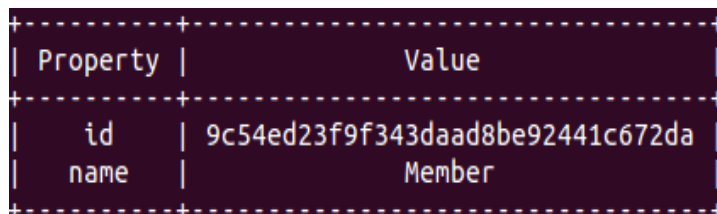


```
+-----+
| Property |          Value          |
+-----+
|   id    | 9677c632b3c04342bd9fbb4b791d4a8 |
|   name  |          admin          |
+-----+
```

FIG. 4.1 – Les propriétés du rôle admin

2. La creation du role member

```
root@openstack-VirtualBox:/home/openstack# keystone role-create --name Member
```



```
+-----+
| Property |          Value          |
+-----+
|   id    | 9c54ed23f9f343daad8be92441c672da |
|   name  |          Member          |
+-----+
```

FIG. 4.2 – Les propriétés du rôle Member

Les rôles `KeystoneAdmin` et `KeystoneServiceAdmin` sont des rôles internes nécessaires.

3. La creation du rôle KeystoneAdmin

```
root@openstack-VirtualBox:/home/openstack# keystone role-create --name KeystoneAdmin
```

Property	Value
id	f348a9687fb74258a2c7a8a7319bca8f
name	KeystoneAdmin

FIG. 4.3 – Les propriétés du rôle *KeystoneAdmin*

4. La creation du rôle **KeystoneServiceAdmin**

```
root@openstack-VirtualBox:/home/openstack# keystone role-create --name KeystoneServiceAdmin
```

Property	Value
id	84321cdbcaa44078945592127d693bb5
name	KeystoneServiceAdmin

FIG. 4.4 – Les propriétés du rôle *KeystoneServiceAdmin*

La creation des tenants

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

```
root@openstack-VirtualBox:/home/openstack# keystone tenant-create --name=admin --description "" --enabled true
```

Property	Value
description	
enabled	True
id	6efce00f6f144fc4be8e1389d3351f6b
name	admin

FIG. 4.5 – Les propriétés du tenant admin

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

```
root@openstack-VirtualBox:/home/openstack# keystone tenant-create --name=service -
-description "" --enabled true
```

Property	Value
description	
enabled	True
id	0c31b9eddf9e4a52bb59644a2d38160f
name	service

FIG. 4.6 – Les propriétés du tenant service

Ajout des utilisateur dans les tenants

Récupération de l'identificateur du tenant admin:

```
root@openstack-VirtualBox:/home/openstack# TENANT_ID=$(keystone tenant-list |
awk '/admin/{print $2}')
```

Creation de l'utilisateur admin:

```
root@openstack-VirtualBox:/home/openstack# keystone user-create --name admin --
tenant_id $TENANT_ID --pass ADMIN --email root@localhost --enabled true
```


Property	Value
email	admin@example.co
enabled	True
id	148c76736620458b81b5b24
name	admin
password	\$6\$rounds=40000\$SYsnkAR0bNGE1bjJ\$y3c7Y1Wg3YT37BybPBagzBZ1qanKjfb9USmV6Ka/JcG0kKrQipuVdvfQ6TfiJvoQupFh20Sym7QW10X0m4NLY.
tenantId	

FIG. 4.7 – Les propriétés de l'utilisateur admin

Récupération de l'identificateur du Rôle admin:

```
root@openstack-VirtualBox:/home/openstack# ROLE_ID=$(keystone role-list | awk '/admin/{print $2}')
```

Récupération de l'identificateur de l'utilisateur admin:

```
root@openstack-VirtualBox:/home/openstack# USER_ID=$(keystone user-list | awk '/admin/{print $2}')
```

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

```
root@openstack-VirtualBox:/home/openstack# keystone user-role-add --user $USER_ID --role $ROLE_ID --tenant_id $TENANT_ID
```

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

L'utilisateur glance après la création:

```

+-----+
| Property | Value
+-----+
| email    | glance@example.c
om
| enabled  | True
| id       | 3b1939e642d54835888f680
21edc5b5c
| name     | glance
| password | $6$rounds=40000$7XVpWQEA3qRPWp4z$0FYkQjppwi1ccxG.BYQk6xEG4CjBkQMuQB
M/pTw//0UHRJ3r1bQZPdRmpgrPgMaDYIYR7jW1XQbXDkJzH/JQT0
| tenantId |
+-----+

```

FIG. 4.8 – Les propriétés de l'utilisateur glance

L'utilisateur nova après la création:

```

+-----+
| Property | Value
+-----+
| email    | nova@example.com
| enabled  | True
| id       | b08ed78d0c804dc6b485c1d2d8befa66
| name     | nova
| password | $6$rounds=40000$djs72Quil7h8NOK0$/FHvXDhlnRMowt9vsYZ89XUSCHRRmUB9ngBVs0rVyOzK
PSYLiA2Z8GaYS3/rMc0W0QVAC0
| tenantId |
+-----+

```

FIG. 4.9 – Les propriétés de l'utilisateur nova

Création des services

1. Le service keystone

```

root@openstack-VirtualBox:/home/openstack# keystone service-create --name keystone
--type identity --description 'OpenStack Identity Service'

```

Property	Value
description	Keystone Identity Service
id	af5df559f4304132863726c7926aa2c3
name	keystone
type	identity

FIG. 4.10 – Les propriétés du service keystone

2. Le service Glance

```
root@openstack-VirtualBox:/home/openstack# keystone service-create --name glance --
type image --description 'OpenStack Image Service'
```

Property	Value
description	Glance Image Service
id	356efef75ec14567a3a3e7010932622c
name	glance
type	image

FIG. 4.11 – Les propriétés du service glance

3. Le service nova compute

```
root@openstack-VirtualBox:/home/openstack# keystone service-create --name nova --
type compute --description 'OpenStack Compute Service'
```

Property	Value
description	OpenStack Compute Service
id	b3dc1a2dc2b744db972181e381f0c8de
name	nova
type	compute

FIG. 4.12 – Les propriétés du service nova compute

4. Le service nova volume

```
root@openstack-VirtualBox:/home/openstack# keystone service-create --name volume -
-type volume --description 'Volume Service'
```

Property	Value
description	OpenStack Volume Service
id	8e5070c21e24451baec850cb9724b074
name	volume
type	volume

FIG. 4.13 – Les propriétés du service nova volume

Définition des points d'accès pour les services

1. Le point d'accès du service Keystone

Récupération de l'identificateur du service de keystone

```
root@openstack-VirtualBox:/home/openstack# ID=$(keystone service-list | awk '/ keystone / {print $2}')
```

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

```
root@openstack-VirtualBox:/home/openstack# PUBLIC="http://172.16.0.1:5000/v2.0"
root@openstack-VirtualBox:/home/openstack# ADMIN="http://172.16.0.1:35357/v2.0"
root@openstack-VirtualBox:/home/openstack# INTERNAL=$PUBLIC
```

```
root@openstack-VirtualBox:/home/openstack# keystone endpoint-create --region RegionOne --service_id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL
```

Property	Value
adminurl	http://172.16.0.1:35357/v2.0
id	c512b36186bd4045ad39e88c7940fa83
internalurl	http://172.16.0.1:5000/v2.0
publicurl	http://172.16.0.1:5000/v2.0
region	RegionOne
service_id	af5df559f4304132863726c7926aa2c3

FIG. 4.14 – Les propriétés du point d'accès du keystone

2. Le point d'accès du service Glance

Récupération de l'identificateur du service glance:

```
root@openstack-VirtualBox:/home/openstack# ID=$(keystone service-list | awk '/ glance / {print $2}')
```

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

```
root@openstack-VirtualBox:/home/openstack# PUBLIC="http://172.16.0.1:9292/v1"
root@openstack-VirtualBox:/home/openstack# ADMIN=$PUBLIC
root@openstack-VirtualBox:/home/openstack# INTERNAL=$PUBLIC
```

Création du point d'accès:

```
root@openstack-VirtualBox:/home/openstack# keystone endpoint-create --region RegionOne --service_id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL
```

Property	Value
adminurl	http://172.16.0.1:9292/v1
id	2216b351fc2140f3b87012d3cafa7ae6
internalurl	http://172.16.0.1:9292/v1
publicurl	http://172.16.0.1:9292/v1
region	RegionOne
service_id	356efef75ec14567a3a3e7010932622c

FIG. 4.15 – Les propriétés du point d'accès glance

3. Le point d'accès du service nova compute

Récupération de l'identificateur du service nova:

```
root@openstack-VirtualBox:/home/openstack# ID=$(keystone service-list | awk '/ nova / {print $2}')
```

```
root@openstack-VirtualBox:/home/openstack# PUBLIC="http://172.16.0.1:8774/v2/$(tenant_id)s"
```

```
root@openstack-VirtualBox:/home/openstack# ADMIN=$PUBLIC
```

```
root@openstack-VirtualBox:/home/openstack# INTERNAL=$PUBLIC
```

```
root@openstack-VirtualBox:/home/openstack# keystone endpoint-create --region RegionOne --service_id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL
```

Property	Value
adminurl	http://172.16.0.1:8774/v2/\$(tenant_id)s
id	236d22d2a2f745859bdf4e1311dbe8a0
internalurl	http://172.16.0.1:8774/v2/\$(tenant_id)s
publicurl	http://172.16.0.1:8774/v2/\$(tenant_id)s
region	RegionOne
service_id	b3dc1a2dc2b744db972181e381f0c8de

FIG. 4.16 – Les propriétés du point d'accès du nova compute

4. Le point d'accès du service nova volume

Récupération de l'identificateur du service volume:

```
root@openstack-VirtualBox:/home/openstack# ID=$(keystone service-list | awk '/ volume / {print $2}')
```

```
root@openstack-VirtualBox:/home/openstack# PUBLIC="http://172.16.0.1:8776/v1/$(tenant_id)s"
```

```
root@openstack-VirtualBox:/home/openstack# ADMIN=$PUBLIC
```

```
root@openstack-VirtualBox:/home/openstack# INTERNAL=$PUBLIC
```

```
root@openstack-VirtualBox:/home/openstack# keystone endpoint-create --region RegionOne --service_id $ID --publicurl $PUBLIC --adminurl $ADMIN --internalurl $INTERNAL
```

Property	Value
adminurl	http://172.16.0.1:8776/v1/(tenant_id)s
id	8ff370013a3e48588469afdb99133d70
internalurl	http://172.16.0.1:8776/v1/(tenant_id)s
publicurl	http://172.16.0.1:8776/v1/(tenant_id)s
region	RegionOne
service_id	a19723a892bc48abb5bcae3b3e2e4a33

FIG. 4.17 – Les propriétés du point d'accès du nova volume

4.3.2 Installation et configuration de Glance

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

Préparation de la base de données Mysql

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

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

Installation

Nous allons installer les paquets suivants:

```
root@openstack-VirtualBox:/home/openstack# apt-get install glance glance-api glance-client glance-common glance-registry python-glance
```

Configuration de glance pour l'utilisation de keystone

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

```
admin_tenant_name = service
admin_user = glance
admin_password = ADMIN
```

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

```
[paste_deploy]
flavor = keystone
```

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

```
sql_connection = mysql://glance:openstack@172.16.0.1:3306/glance
```

Et nous ajoutons à la fin du fichier:

```
[paste_deploy]
flavor = keystone
```

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

```
sql_connection = mysql://glance:openstack@172.16.0.1:3306/glance
sql_idle_timeout = 3600
```

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

```
admin_tenant_name = service
admin_user = glance
admin_password = ADMIN
```


Nous allons synchroniser la base de données MySQL:

```
root@openstack-VirtualBox:/home/openstack# glance-manage version_control 0
root@openstack-VirtualBox:/home/openstack# glance-manage db_sync
```

Et enfin, nous redémarrons les services pour la prise en compte des modifications:

```
root@openstack-VirtualBox:/home/openstack# service glance-api restart
root@openstack-VirtualBox:/home/openstack# service glance-registry restart
```

Utilisation de glance

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

```
root@openstack-VirtualBox:/home/openstack# wget http://uec-
images.ubuntu.com/releases/precise/release/ ubuntu-12.04-server-cloudimg-amd64-
disk1.img
```

Nous ajoutons maintenant l'image téléchargée aux images Glance:

```
root@openstack-VirtualBox:/home/openstack# glance add name="Ubuntu 12.04 clou-
ding amd64" is_public=true container_format=ovf disk_format=qcow2 < ubuntu-12.04-
server-cloudimg-amd64-disk1.img
```

Nous devons avoir un message qui indique que l'image a été bien ajoutée, avec l'affichage d'un identificateur pour cette image:

```
root@openstack-VirtualBox:/home/openstack# glance add name="Ubuntu 12.04 cloudimg amd64" is_public=true
ntainer_format=ovf disk_format=qcow2 < ubuntu-12.04-server-cloudimg-amd64-disk1.img
Added new image with ID: 0a58e4e0-2197-49d7-bd44-1dfc8bbff474
```

FIG. 4.18 – L'identificateur de l'image après l'ajout à glance

4.3.3 Installation et configuration de Nova

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

Préparation de la base de données Mysql

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

```
mysql > mysql -u root -p
mysql > CREATE DATABASE nova;
mysql > GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY
'openstack';
mysql > FLUSH PRIVILEGES;
mysql > Quit;
```

Installation

Nous allons installer les paquets suivants:

```
root@openstack-VirtualBox:/home/openstack# apt-get install nova-api nova-cert nova-
common nova-compute nova-compute-kvm nova-doc nova-network nova-objectstore
nova-scheduler novnc nova-consoleauth nova-volume nova-console python-nova python-
novaclient.
```

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

```
admin_tenant_name = service
admin_user = nova
admin_password = ADMIN
```

Enfin, nous allons éditer le fichier `/etc/nova/nova.conf` et le compléter par les lignes ci-dessous:

```
cc_host=172.16.0.1
nova_url=http://172.16.0.1:8774/v1.1/
routing_source_ip=172.16.0.1
s3_host=172.16.0.1 ec2_host=172.16.0.1
ec2_url=http://172.16.0.1:8773/services/Cloud
ec2_private_dns_show_ip
ec2_dmz_host=172.16.0.1
keystone_ec2_url=http://172.16.0.1:5000/v2.0/ec2tokens
scheduler_driver=nova.scheduler.simple.SimpleScheduler
scheduler_default_filters=AllHostsFilter
# AUTHENTICATION
auth_strategy=keystone
# Base de données
sql_connection=mysql://nova:openstack@172.16.0.1/nova
libvirt_type=qemu
allow_resize_to_same_host=True
start_guests_on_host_boot=true
resume_guests_state_on_host_boot=true
rabbit_host=172.16.0.1
GLANCE image_service=nova.image.glance.GlanceImageService
glance_api_servers=172.16.0.1:9292
# NETWORK
network_manager=nova.network.manager.FlatDHCPManager
fixed_range=10.0.1.0/24
flat_network_dhcp_start=10.0.1.2
flat_network_bridge=br100
flat_interface=eth1
flat_injected=False
my_ip=172.16.0.1
floating_range=172.16.1.0/24
public_interface=eth1
```

```
auto_assign_floating_ip=true
# NOVNC CONSOLE
vnc_enabled=true
novncproxy_base_url=http://172.16.0.1:6080/vnc_auto.html
vncserver_proxycient_address=172.16.0.1
vncserver_listen=172.16.0.1
```

Puis, nous redémarrons tous les services nova:

```
root@openstack-VirtualBox:/home/openstack# for a in libvirt-bin nova-network nova-
compute nova-api nova-objectstore nova-scheduler nova-volume nova-cert nova-
consoleauth novnc; do sudo service "$a" stop; done
root@openstack-VirtualBox:/home/openstack# for a in libvirt-bin nova-network nova-
compute nova-api nova-objectstore nova-scheduler nova-volume nova-cert nova-
consoleauth novnc; do sudo service "$a" start; done
```

Nous synchronisons la base de données:

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

Et nous redémarrons de nouveau tous les services:

```
root@openstack-VirtualBox:/home/openstack# for a in libvirt-bin nova-network nova-
compute nova-api nova-objectstore nova-scheduler nova-volume nova-cert nova-
consoleauth novnc; do sudo service "$a" stop; done
root@openstack-VirtualBox:/home/openstack# for a in libvirt-bin nova-network nova-
compute nova-api nova-objectstore nova-scheduler nova-volume nova-cert nova-
consoleauth novnc; do sudo service "$a" start; done
```

Utilisation de nova

Nous pouvons maintenant vérifier que tous les services fonctionnent, le résultat dans la colonne STATE doit être ":-)", si tout a été bien configuré. Un "XXX" sera affiché si quelque chose n'as pas fonctionné.

```
root@openstack-VirtualBox:/home/openstack# nova-manage service list
```

Binary	Host	Zone	Status	State	Updated_At
nova-consoleauth	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:20
nova-cert	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:20
nova-volume	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:18
nova-compute	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:16
nova-scheduler	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:18
nova-network	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:21
nova-console	cloud-VirtualBox	nova	enabled	:-)	2013-06-18 15:41:19

FIG. 4.19 – *Le bon fonctionnement de nova*

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

```
root@openstack-VirtualBox:/home/openstack# nova image-list
```

ID	Name	Status	Server
9ff2b557-f11b-49b7-9ca8-18d32a222648	Ubuntu 12.04 cloudimg amd64	ACTIVE	

FIG. 4.20 – *Les images disque fournies par glance*

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

- **Le réseau public:**

```
root@openstack-VirtualBox:/home/openstack# nova-manage floating create --
ip_range=172.0.1.0/24
```

- **Le réseau privé:**

```
root@openstack-VirtualBox:/home/openstack# nova-manage network create private --
fixed_range_v4=10.0.1.0/24 --num_networks=1 --bridge=br100 --bridge.interface=eth2
```

3. **Parefeu (groupe de sécurité):** Par défaut, les règles de parefeu bloquent les paquets entrants sur l'interface publique à destination des VMs. Il est possible de créer un ensemble de règles qui devra être spécifié au lancement de chaque instance.

Nous allons spécifier quelques règles sur l'ensemble "default" qui est créé automatiquement lors de l'installation de nova.

Pour pouvoir communiquer avec nos machines virtuelles depuis n'importe quelle machine, on doit autoriser certains protocoles sur nos machines virtuelles.

Dans notre cas, nous voulons autoriser l'accès SSH et le ping, pour toutes les VMs sur l'interface publique.

SSH est basé sur le protocole TCP connecté sur le port 22 et le ping est une requête icmp.

```
root@openstack-VirtualBox:/home/openstack# nova secgroup-add-rule default icmp -1
-1 0.0.0.0/0
```

Le icmp -1 correspond à tout les ports.

IP Protocol	From Port	To Port	IP Range	Source Group
icmp	-1	-1	0.0.0.0/0	

FIG. 4.21 – La règle de parefeu pour l'accès SSH

```
root@openstack-VirtualBox:/home/openstack# nova secgroup-add-rule default tcp 22 22
0.0.0.0/0
```

IP Protocol	From Port	To Port	IP Range	Source Group
tcp	22	22	0.0.0.0/0	

FIG. 4.22 – La règle de parefeu pour le ping

Première machine virtuelle

Maintenant que tout fonctionne, nous allons pouvoir créer notre première VM. Nous allons créer une clé SSH:

```
root@openstack-VirtualBox:/home/openstack# ssh-keygen -t rsa
```

Nous ajoutons cette clé au serveur (à la base de données nova):

```
root@openstack-VirtualBox:/home/openstack# nova keypair-add --pub_key
~/.ssh/id_rsa.pub key1
```

```
root@openstack-VirtualBox:/home/openstack# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
08:e2:7b:94:d5:3a:27:5b:f9:1b:8b:da:87:ad:e4:5e root@openstack-VirtualBox
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
| .
| . . . .
| . . + O .
| . o = S
| o * .
| . . . .oE
| . +.oo+
| .o*oo
|
```

FIG. 4.23 – La création de la clé SSH

Il faut définir les spécifications de la future VM, pour voir les possibilités, nous utilisons la commande ci-dessous:

```
root@openstack-VirtualBox:/home/openstack# nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor	Is_Public	extra_specs
1	m1.tiny	512	0	0		1	1.0	True	{}
2	m1.small	2048	10	20		1	1.0	True	{}
3	m1.medium	4096	10	40		2	1.0	True	{}
4	m1.large	8192	10	80		4	1.0	True	{}
5	m1.xlarge	16384	10	160		8	1.0	True	{}

FIG. 4.24 – Les modèles de VMs existant

Une même image virtuelle pourra être utilisée de plusieurs façon différentes selon la contextualisation qu'on lui appliquera.

Par la suite, nous allons utiliser l'ID "1" qui correspond à une machine disposant de 512 Mb de RAM, 1 CPU virtuel et aucun disque supplémentaire.

Nous lançons notre première VM avec la commande **nova boot**:

```
root@openstack-VirtualBox:/home/openstack# nova boot --flavor 1 --image d1b7defa-0c35-4e8c-aef5-0d58c8d80a52 machine1 --key_name key1 &
```

- `--flavor` indique quel modèle de machine choisi.
- `--image` correspond à l'ID de l'image fournie par glance.
- `machine1` est le nom de la machine créée.
- `--key_name` est la clé ssh utilisée.

Si aucun ensemble de règles de parefeu n'est indiqué, c'est l'ensemble "default" qui sera appliqué.

Un récapitulatif des propriétés de la machine créée s'affiche:

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	instance-00000001
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
accessIPv4	
accessIPv6	
adminPass	viRcaWdkwNq4
config_drive	
created	2013-06-04T16:01:38Z
flavor	m1.tiny
hostId	
id	f40f69e1-ddfb-418b-8f79-95f8502d5fd2
image	Ubuntu 12.04 cloudimg amd64
key_name	key
metadata	{}
name	mv
progress	0
security_groups	[[{u'name': u'default'}]]
status	BUILD
tenant_id	f05fdf3d217c41728f5edb3d0bd4f410
updated	2013-06-04T16:01:38Z
user_id	5b0d89265fff4a3e974198a4d8ce9c5d

FIG. 4.25 – Les propriétés de l'instance créée

Nous pouvons vérifier que l'instance créée est en état actif et qu'elle possède une adresse privée:

```
root@openstack-VirtualBox:/home/openstack# nova list
```

ID	Name	Status	Networks
749539df-a04c-4df6-b038-a744db998be7	machine1	ACTIVE	private=10.0.1.2
c9edb7fa-8906-4a9c-b174-71c47e8c0462	machine2	ACTIVE	private=10.0.1.3

FIG. 4.26 – Les instances créées.

Configuration du service nova-volume

Les volumes LVM serviront de disques durs supplémentaires pour les serveurs virtuels.

pour pouvoir utiliser nova-volume nous devons effectuer quelques changements à

notre machine virtuelle, en ajoutant à cette dernière un nouveau disque. L'ajout d'un nouveau disque à notre machine virtuelle dans VirtualBox se fera par les étapes suivantes:

- Eteindre la machine virtuelle.
- Ajouter un espace de stockage dans la section **Setting**.
- Puis dans la section **Storage tree** créer le nouveau disque avec le nom "nova-volume".

Nous allons passer à la configuration de notre élément de stockage avec nova-volume en utilisant **fdisk**:

```
root@openstack-VirtualBox:/home/openstack# sudo fdisk /dev/sdb
```

Dans le menu fdisk nous allons effectuer les opérations suivantes:

- Taper n, et cliquer sur la touche Entrer du clavier.
- Taper p, et cliquer sur la touche Entrer du clavier.
- Taper 1, puis Entrer.
- Taper Enter pour l'étape (default 2048).
- Taper Enter pour l'étape (default).
- Taper t, puis taper Enter.
- Taper 8e, puis Entrer pour définir le type de la partition.
- Pour confirmer la création de la partition LVM dans le nouveau volume, taper p puis Enter.
- Taper w pour enregistrer les modifications effectuées.

Nous allons créer le volume LVM dans la partition 1 de /dev/sdb:

```
root@openstack-VirtualBox:/home/openstack# sudo pvcreate /dev/sdb1
root@openstack-VirtualBox:/home/openstack# sudo vgcreate nova-volumes /dev/sdb1
```

Après cette étape nous allons installer les paquets suivants:

```
root@openstack-VirtualBox:/home/openstack# apt-get install tgt open-iscsi
```

Et nous démarrons les services tgt et open-iscsi:

```
root@openstack-VirtualBox:/home/openstack# service tgt start
root@openstack-VirtualBox:/home/openstack# service open-iscsi start
```

Ensuite, nous redémarrons le service nova-volume pour la prise en charge des modifications:

```
root@openstack-VirtualBox:/home/openstack# service nova-volume restart
```

Puis, nous rajoutons au fichier `/etc/nova/nova.conf` les lignes suivantes:

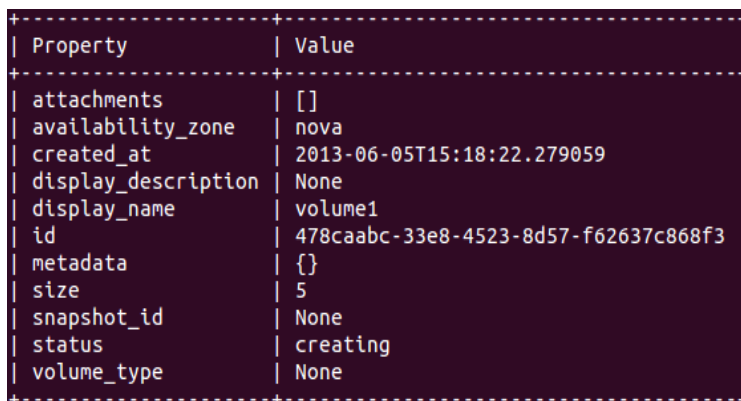
```
iscsi_ip_address=172.16.0.1
iscsi_helper=tgtadm
```

Et enfin, nous redémarrons le service nova-compute:

```
root@openstack-VirtualBox:/home/openstack# service nova-compute restart
```

Nous pouvons maintenant créer un disque dur supplémentaire:

```
root@openstack-VirtualBox:/home/openstack# nova volume-create --display_name "volume1" 5
```



Property	Value
attachments	[]
availability_zone	nova
created_at	2013-06-05T15:18:22.279059
display_description	None
display_name	volume1
id	478caabc-33e8-4523-8d57-f62637c868f3
metadata	{}
size	5
snapshot_id	None
status	creating
volume_type	None

FIG. 4.27 – Les propriétés du volume créé.

Attachons-le à la VM créée:

```
root@openstack-VirtualBox:/home/openstack# nova volume-attach machine1 /dev/vdb
```

Nous pouvons nous reconnecter à la VM pour partitionner ce disque et l'utiliser.

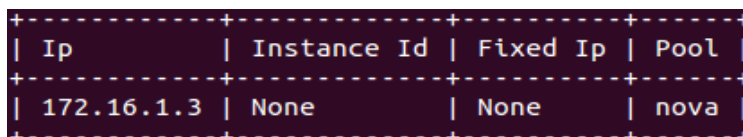
Attribution des adresses publiques

Maintenant nous allons connecter notre instance virtuelle à notre LAN par l'attribution d'adresses publique aux instances. Notons que l'attribution d'adresse IP publiques (floating) peut se faire de deux façons:

- Attribution dynamique en ajoutant dans le fichier nova.conf la ligne:
auto_assign_floating_ip=true
- L'attribution manuelle se fait en commentant cette ligne dans le même fichier et en procédant comme suite:

Il faut tout d'abord allouer une adresse IP:

```
root@openstack-VirtualBox:/home/openstack# nova floating-ip-create
```



Ip	Instance Id	Fixed Ip	Pool
172.16.1.3	None	None	nova

FIG. 4.28 – L'adresse IP publique allouée.

Puis nous allons attacher cette adresse à la VM:

```
root@openstack-VirtualBox:/home/openstack# nova add-floating-ip machine1 172.16.1.3
```

La commande nova list nous confirme l'attribution:

```
root@openstack-VirtualBox:/home/openstack# nova list
```

ID	Name	Status	Networks
749539df-a04c-4df6-b038-a744db998be7	machine1	ACTIVE	private=10.0.1.2, 172.16.1.3
c9edb7fa-8906-4a9c-b174-71c47e8c0462	machine2	ACTIVE	private=10.0.1.3, 172.16.1.4

FIG. 4.29 – L'attribution manuelle d'une adresse publique.

4.3.4 Installation du Dashboard

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

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

Nous allons installer les paquets suivants:

```
root@openstack-VirtualBox:/home/openstack# apt-get install openstack-dashboard
apache2 libapache2-mod-wsgi
```

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

```
OPENSTACK_HOST = "172.16.0.1"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v2.0" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "Member"
```

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

```
root@openstack-VirtualBox:/home/openstack# sudo service apache2 restart
```

Utilisation

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

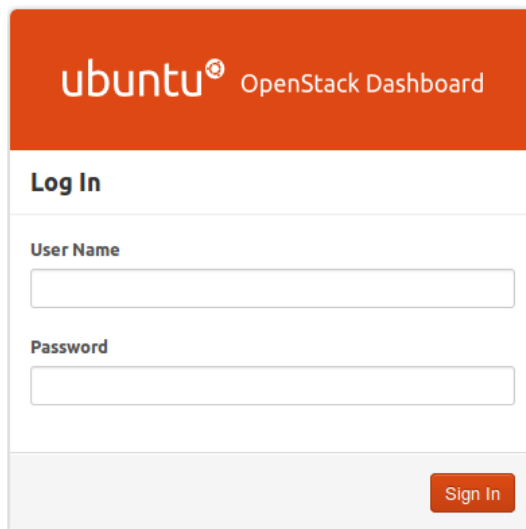
The image shows a web browser window displaying the Ubuntu OpenStack Dashboard login page. At the top, there is an orange header bar with the text "ubuntu® OpenStack Dashboard". Below the header, the page is titled "Log In". There are two input fields: "User Name" and "Password". At the bottom right of the form, there is a red "Sign In" button.

FIG. 4.30 – L'interface d'authentification du dashboard.

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

Les fonctions du Dashboard

Les comptes qui ont pour rôle "admin" ont accès à l'interface d'administration sur le Dashboard, ainsi qu'à leur(s) projet(s).

ubuntu[®] OpenStack Dashboard Logged in as: admin Settings Help Sign Out

Project **Admin** **Overview**

System Panel

- Overview
- Instances
- Volumes
- Services
- Flavors
- Images
- Projects
- Users
- Quotas

Select a month to query its usage:

June 2013

Active Instances: 4 Active RAM: 2GB This Month's VCPU-Hours: 138.51 This Month's GB-Hours: 0.00

Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours
admin	2	0	1GB	101.24	0.00
development	2	0	1GB	37.27	0.00

Displaying 2 items

FIG. 4.31 – L'interface de l'utilisateur admin

Les rôles "Member" n'ont accès qu'à leur(s) projet(s).

Project **Overview**

CURRENT PROJECT **development**

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

Quota Summary

Used 2 of 10 Available Instances

Used 2 of 20 Available vCPUs

Used 1,024 MB of 51,200 MB Available RAM

Select a month to query its usage:

June 2013

Active Instances: 2 Active RAM: 1GB This Month's VCPU-Hours: 0.17 This Month's GB-Hours: 0.00

Instance Name	VCPUs	Disk	RAM	Uptime
machine1	1	0	512MB	6 minutes
machine2	1	0	512MB	3 minutes

FIG. 4.32 – L'interface des utilisateurs avec le rôle member

1. Accès Admin

Dans l'ordre les différents menus :

- **Overview:** affiche un récapitulatif de l'usage des serveurs par projet, utilisation actuelle en nombre de CPU virtuels, RAM et Disques puis

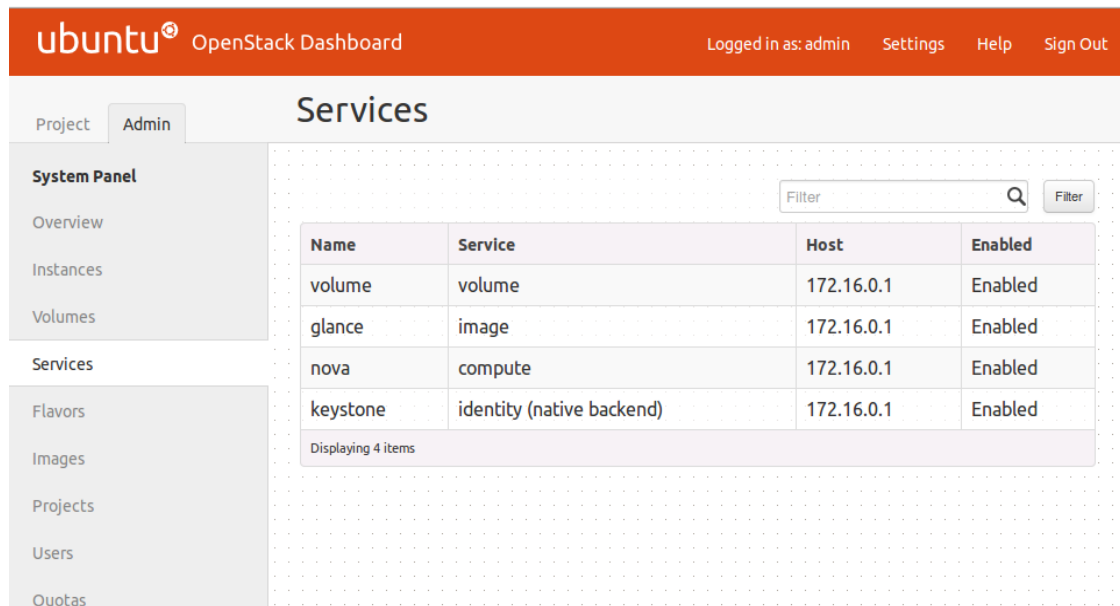
compteur en CPU et espace disque(GB) par heures.

- **Instances:** liste des instances de machines virtuelles actuelles plus quelques informations globales comme le projet auquel elles appartiennent, le serveur hôte, l'adresse IP (publique et privé), la taille, le statut et les actions en cours.
- **Services:** liste des services activés et le serveur hôte.
- **Flavors:** la liste des types d'instances disponibles, leurs spécifications en nombre de CPUs, mémoire, espace disque. La création de nouvelles définitions d'instance est possible.
- **Images:** les images disques stockées par le service Glance.
- **Projects:** les projets existants et leur statut. Il est possible de créer de nouveaux projets.
- **Users:** la liste des utilisateurs enregistrés, avec la possibilité d'ajouter ou d'éditer les détails mais pas d'ajouter l'utilisateur à plusieurs projets.
- **Quotas:** les quotas définis par l'administrateur sur les ressources des serveurs.

2. Accès projets

- **Overview:** récapitulatif, comme dans la partie Admin, mais ne concernant que le projet sélectionné.
- **Instances:** affiche la liste des instances existantes et la possibilité de les éditer.
- **Volumes:** permet la création ou modification des volumes disques virtuels.
- **Images & Snapshots:** affiche la liste des images autorisées pour le projet, sert à lancer de nouvelles instances. Regroupe aussi les instantanés disponibles, instances et volumes disques.
- **Acces & Security:** gère les adresses IP disponibles pour connecter les instances au réseau public avec possibilité d'ajouter ou de supprimer de nouvelles adresses, affiche les groupes de règles de parefeu et leur interface d'édition et enfin la liste des clés SSH .

La figure suivante donne un aperçu sur les différents services internes créés par l'administrateur.



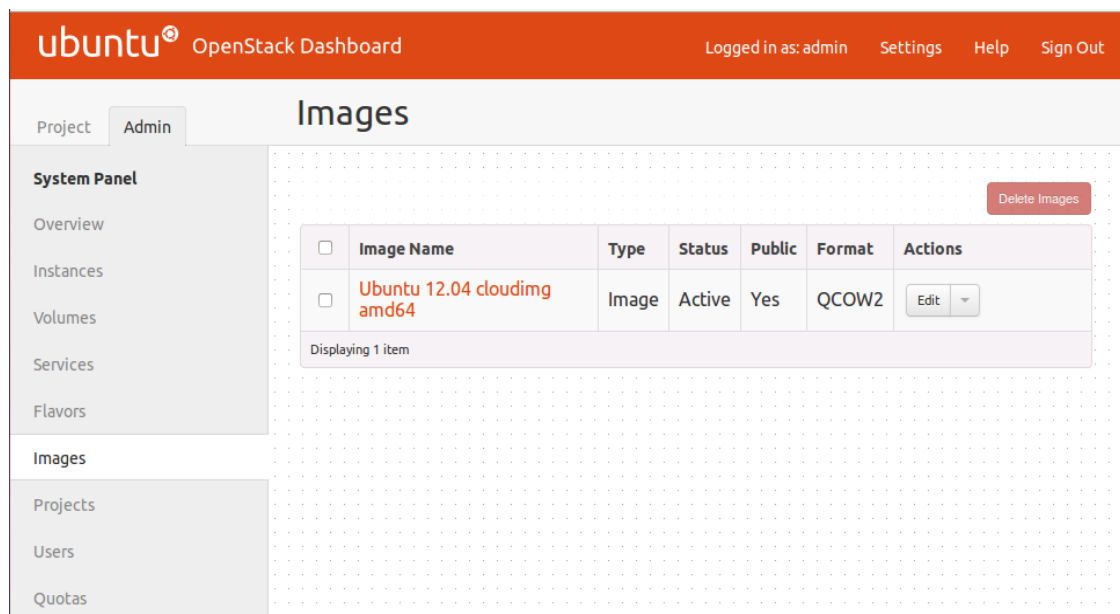
The screenshot shows the OpenStack Dashboard interface. The top navigation bar includes the Ubuntu logo, 'OpenStack Dashboard', and user information: 'Logged in as: admin', 'Settings', 'Help', and 'Sign Out'. The main content area is titled 'Services' and features a sidebar on the left with a 'System Panel' containing links for Overview, Instances, Volumes, Services, Flavors, Images, Projects, Users, and Quotas. The 'Services' section is active. A table displays the following data:

Name	Service	Host	Enabled
volume	volume	172.16.0.1	Enabled
glance	image	172.16.0.1	Enabled
nova	compute	172.16.0.1	Enabled
keystone	identity (native backend)	172.16.0.1	Enabled

Below the table, it indicates 'Displaying 4 items'.

FIG. 4.33 – Les services créés.

la figure ci-dessous décrit l'image disque utilisée pour instancier des machines virtuelles, ainsi ses propriétés.



The screenshot shows the OpenStack Dashboard interface for the 'Images' section. The top navigation bar is identical to the previous figure. The main content area is titled 'Images' and features a sidebar on the left with a 'System Panel' containing links for Overview, Instances, Volumes, Services, Flavors, Images, Projects, Users, and Quotas. The 'Images' section is active. A table displays the following data:

<input type="checkbox"/>	Image Name	Type	Status	Public	Format	Actions
<input type="checkbox"/>	Ubuntu 12.04 cloudimg amd64	Image	Active	Yes	QCOW2	Edit

Below the table, it indicates 'Displaying 1 item'. A 'Delete Images' button is visible in the top right corner of the table area.

FIG. 4.34 – L'image fournie par le service glance

La création d'une instance par l'administrateur ou par l'utilisateur en utilisant l'interface Dashboard se fait de la manière suivante:

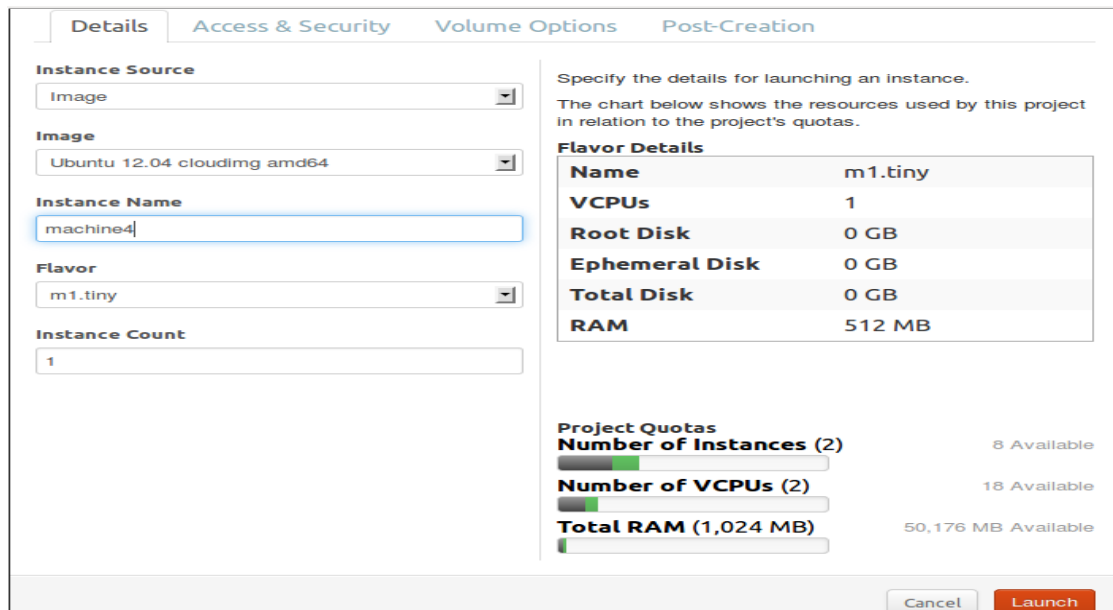


FIG. 4.35 – La création d'une instance

Après la création des instances en ligne de commande, ces dernières apparaissent sur la fenêtre **Instance**, ainsi leurs adresses Ip spécifique et leurs état.

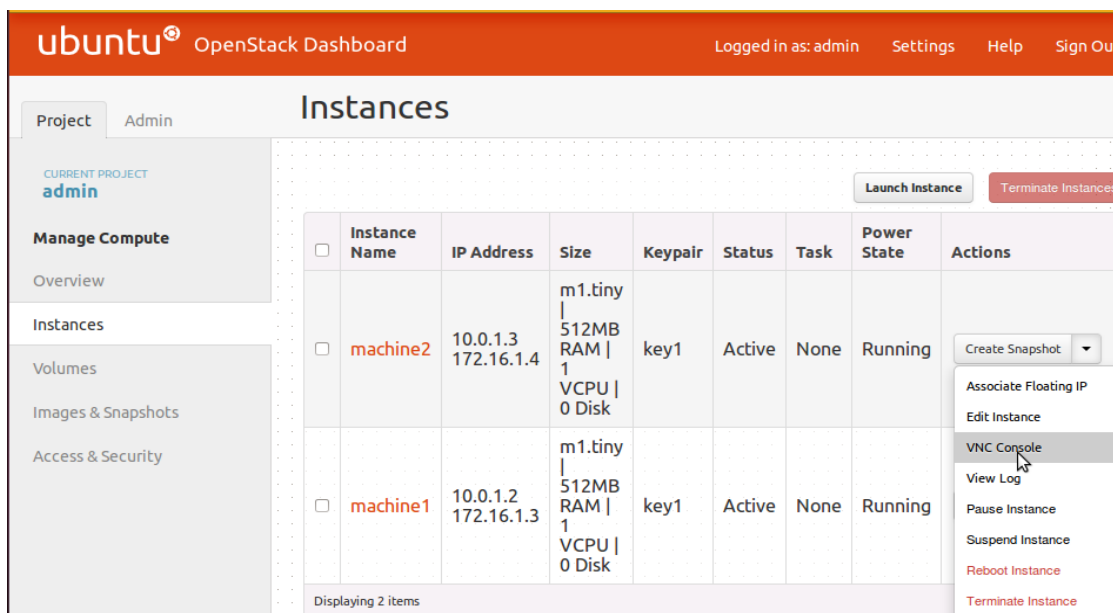


FIG. 4.36 – Les instances créées par l'administrateur

La création d'un utilisateur par un administrateur est montrée dans la fenêtre sui-

vante:

FIG. 4.37 – Création d'un utilisateur par l'administrateur

L'interface du Dashboard donne accès au site officiel d'OpenStack par le bouton **help**.

FIG. 4.38 – Aperçu du bouton help de l'interface dashboard

4.4 Conclusion

A travers ce chapitre nous avons décrit les étapes suivies pour la mise en place de la solution de Cloud Computing OpenStack, par l'installation et la configuration des composants: Keystone, Glance, Nova et dashboard.

Conclusion Générale et Perspectives

L'objectif de notre projet était d'étudier quelques solutions libres de Cloud Computing pour pouvoir mettre en place l'une d'entre elles. En effet, nous avons pu analyser trois solutions de cloud privé: Eucalyptus, OpenNebula et OpenStack et de tester cette dernière.

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

La mise en place d'une telle solution n'était pas facile à cause du manque de la documentation et de la communauté Francophone autour de cette solution.

Ce projet étant très compliqué, nous avons rencontré de nombreux problèmes lors de la configuration d'OpenStack, notamment en ce qui concerne le réseau. Tous ces problèmes nous ont montré la complexité de configurer une solution de Cloud Computing, et leur résolution nous a souvent retardé et pénalisé.

La gestion de l'environnement OpenStack par le biais d'une interface graphique Dashboard, permet de compléter le niveau de contrôle offert par la ligne de commande et rend le processus de gestion de Cloud plus facile pour les utilisateurs et les administrateurs.

En effet la solution OpenStack permet à un utilisateur de se connecter via un navigateur Web sur un Dashboard, en fonction de son profil, il a le droit ou non de créer des machines virtuelles, de créer des images, des snapshots, de gérer son réseau, etc.

En perspectives, nous proposons la mise en place de la solution de Cloud Computing OpenStack sur un environnement plus large à savoir sur un serveur pour l'université de Bejaia, qui permettra aux étudiants de profiter des avantages d'un Cloud et de créer leur propres machines virtuelles selon des règles dictées par l'administrateur.

Il sera donc important qu'un étudiant continue ses modifications sur un TP même en dehors des murs de l'université, il ne se soucie pas de la perte de ses données et rend ses ressources disponibles de par tout. Cela donnera lieu à l'installation et configuration des composants: Quantum, Swift et Cinder qui est nécessaire dans ce cas pour assurer le bon fonctionnement d'un tel environnement et prochainement l'installation sur un environnement multi nuds qui va permettre de voir de prêt comment remédier à une surcharge de demandes (migration à chaud), et de tester ses fonctionnalités qui sont nécessaires sur un environnement pareil.

BIBLIOGRAPHIE

- [1] S. Ayadi et all. *Cloud Computing Connectivity (Windows Azure, DRMAA) with ProActive Resource Manager*. Rapport, Université nice SOPHIA ANTIPOLIS, 2009-2010.
- [2] L. Bonnet Bearstech. *Etat de l'art des solutions libres de virtualisation pour une petite entreprise*.
- [3] M. Audin. *Etat de l'art du Cloud Computing et adaptation au Logiciel Libre Bearstech*. 2009.
- [4] R. Hennion et all. *Cloud Computing*. Groupe Eyrolles, ISBN: 978-2-212-13404-9, 2012.
- [5] Introduction to cloud computing. <http://www.dialogic.com>.
- [6] M. Lucas Nussbaum et all. *Administration de systèmes, réseaux et applications à base de logiciels libres*. IUT Nancy Charlemagne, 2009/2010.
- [7] Wygwam Bureau d'expertise technologique de Aurélien Prunier. *Le Cloud Computing: Réelle révolution ou simple évolution?* 2012.
- [8] Dr. S. Hüsemann. *Bachelor en Gestion d'entreprise; Les enjeux du Cloud Computing en entreprise*. Université de Fribourg, Suisse Département d'informatique, 2012.
- [9] T. Wu et all. A survey of open-source cloud infrastructure using futuregrid testbed, Publié le 14 mai 2012.
- [10] Y. Parent et all. *Cloud Computing*. Rapport, Université Nancy 2 IUT Nancy-Charlemagne, Année universitaire 2010-2011.

-
- [11] N. Broisin. *Plateforme de stockage dans le Cloud EUCALYPTUS*. 09/01/2012 - 24/02/2012.
- [12] Y. Nouri P. Neveux. Opennebula: Etude comparative, fiche technique synthétique. website, consulté le 11/05/2012. [http://\[INF346\]OpenNebula.htm](http://[INF346]OpenNebula.htm).
- [13] ERenaux A.Deotto et all. *Le Cloud Computing: Comment installer son Cloud privé*. Université Nancy 2 IUT Nancy-Charlemagne, Publié le 14 mai 2012.
- [14] Site officiel d'opennebula. website, consulté en mars 2013. <http://opennebula.org/documentation:archives:rel3.0:plan>.
- [15] Site officiel d'OpenNebula. Opennebula caractéristiques et fonctionnalités. website, consulté en mars 2013. <http://opennebula.org/documentation:features>.
- [16] Site officiel d'OpenNebula. Un aperçu de opennebula. website, consulté en mars 2013. <http://opennebula.org/documentation:rel3.8:intro>.
- [17] <http://perso.telecom-paristech.fr/~nouri/OpenNebula2.html>.
- [18] site officiel d'OpenStack. <http://www.openstack.org/software/openstack-compute/>.
- [19] <http://www.inzecloud.fr/construire-un-cloud-compatible-vmware-avec-une-solution-libre-openstack/>.
- [20] site officiel d'OpenStack. <http://docs.openstack.org/trunk/openstack-compute/admin/content/concept%ual-architecture.html>.
- [21] D. Thain P. Sempolinski. *A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus*. University of Notre Dame.
- [22] <https://help.ubuntu.com/community/UEC/PackageInstall>.
- [23] http://www.rackspace.com/cloud/private/openstack_software/.
- [24] site officiel d'OpenStack. <http://www.openstack.org/software/openstack-storage/>.
- [25] <http://fr.redhat.com/about/news/archive/2012/11/red-hat-openstack-technic%al-preview-now-with-folsom>.
- [26] site officiel d'OpenStack. <http://docs.openstack.org/trunk/openstack-compute/install/yum/content/v%ersion.html>.
- [27] <http://uksysadmin.wordpress.com/2012/09/23/what-to-expect-from-the-openstack-folsom-release/>.

- [28] B. Donnette et all. *La virtualisation*. Edité par le Groupe LINAGORA paris, Décembre 2007.
- [29] *le cloud computing une nouvelle filière fortement structurante*. Direction régionale des entreprises, de la concurrence, de la consommation du travail et de l'emploi (DIRECCTE) ILE-DE-FRANCE, septembre 2012.
- [30] site officiel d'OpenStack. <http://www.openstack.org/software/openstack-dashboard/>.
- [31] E. Renaux. *Le cloud computing: Comment installer son cloud privé*, Publié le 14 mai 2012.
- [32] *Red Hat OpenStack Red Hat OpenStack 2.1 (Folsom) Getting Started Guide*. 2012, 2013.
- [33] K. Jackson. *OpenStack Cloud Computing Cookbook*. ISBN 978-1-84951-732-4, September 2012.
- [34] site officiel d'OpenStack. <http://www.openstack.org/software/openstack-networking/>.
- [35] D. Johnson, 25 mai 2010. <http://ebookbrowse.com/search/eucalyptus-beginners-guide-uec-edition1>.
- [36] <http://doc.ubuntu-fr.org/virtualbox>.
- [37] Site officiel d'OpenStack. <http://docs.openstack.org/essex/openstack-compute/starter/content/Security\%20Overview-d1e2505.html>.
- [38] <https://github.com/obuisson/Guide-Installation-OpenStack-Folsom-Noeud-Unique/blob/master/source/index.rst>.

Problèmes Rencontrés

- Lors de la création des bases de données concernant les différents services (Keystone, Glance), nous avons utilisé la commande: **sudo keystone-manage db_sync** pour keystone et **sudo glance-manage db_sync** pour glance pour créer les tables nécessaires au fonctionnement de ces services, pour synchroniser la base de donnée nova nous avons fait: **sudo nova-manage db_sync** mais les tables ne sont pas créées. Il fallait utiliser la commande suivante: **nova-manage db sync**.
- Lors du démarrage de l'instance elle s'affiche en statut: **ERROR**
Nous avons vérifié dans le fichier `/etc/nova/nova-compute.conf`: nous avons trouvé que le mode de virtualisation utilisé est kvm avec la ligne: `libvirt_type=kvm`. Nous avons modifié la ligne précédente par: `libvirt_type=qemu`, car VirtualBox fonctionne avec le mode de virtualisation qemu.
- Nous avons aussi eu le problème de non attribution de l'adresse IP privée à ces instances, avec le mode flat Dhcp, malgré que le fichier de la configuration réseau `Network/interfaces` et le fichier `nova.conf` sont bien configurés, nous avons essayé de changer le mode réseau en Flat et même en mode Vlan Network mais c'était toujours le même problème, nous avons opté à la suppression des paquets: `Network-Manager`, et `resolvconf` car ces derniers créent un conflit avec `nova-network` et le problème s'est réglé.
- Après l'installation du dashboard et de apache2 nous avons redémarré le service apache2: `sudo /etc/init.d/apache2 restart`, mais l'erreur suivante s'affiche:

Could not reliably determine the servers fully qualified domain name, using 127.0.1.1 for ServerName” Error on Ubuntu

Pour résoudre ce problème, nous avons modifiés le fichier httpd.conf comme suite:

```
gedit / etc/apache2/httpd.conf.
```

Par défaut le fichier est vide, nous avons ajouté la ligne: ServerName cloud computing.

A. Ubuntu 12.04 LTS

A.1. Présentation

Ubuntu 12.04 LTS est la sixième version d'Ubuntu et la quatrième version LTS (Long Term Support). Ubuntu 12.04 LTS est disponible en version finale depuis le 26 avril 2012), cette version se définit comme étant la dernière version stable d'Ubuntu.

B. Notion importantes

B.1. repository Ubuntu Cloud Archive

C'est le dépôt de paquets logiciels offert aux individus et aux équipes de développeurs.

B.2. Les différents composants internes de nova et glance

B.2.1. nova-api

Accepte et répond aux appels d'API des utilisateurs. Il prend en charge l'API OpenStack, API d'Amazon EC2, elle initie également la plupart des activités d'orchestration (comme l'exécution d'une instance).

B.2.2. nova-compute

Est un processus qui crée et met fin aux instances de machines virtuelles par les APIs des hyperviseur (XenAPI pour XenServer / XCP, libvirt pour KVM ou QEMU, VMwareAPI pour VMware, etc.)

B.2.3. Nova-volume

Gère la création, l'attachement et le détachement des volumes persistants pour les noeuds de calcul (fonctionnalité similaire à Elastic Block Storage Amazon). Ils sont utilisés pour la gestion des volumes LVM.

B.2.4. nova-network

IL est très similaire à nova-Compute et nova-volume. Il accepte les tâches de mise en réseau à partir de la file d'attente, exécute ensuite les commandes du système pour la manipulation du réseau (comme la mise en place d'interfaces de transition, modification des règles iptables, l'allocation des adresses IP, la configuration des VLAN pour les projets, la mise en oeuvre des groupes de sécurité et la configuration de réseaux pour les nuds de calcul).

B.2.5. nova-scheduler

Il est conceptuellement le plus simple processus dans Nova, son role est de recevoir une demande de machine virtuelle par exemple à partir de la file d'attente et déterminer l'endroit où elle devrait fonctionner (sur quel hote devrait elle fonctionner). Un ordonnanceur peut fonder ses décisions sur divers facteurs tels que la charge, la mémoire, la distance physique de la zone de disponibilité, de l'architecture du processeur, etc

B.2.6. glance-api

Accepte les appels de découverte , la recherche et le stockage des images de machines virtuelles.

B.2.7. glance-registry

Enregistre les processus et récupère les métadonnées sur les images (taille, type, etc.).

B.3. Interface réseau

Une interface réseau est un nom associé à une carte réseau physique (carte wifi, carte ethernet, modem...) ou virtuelle (boucle locale, VPN...). Les interfaces réseaux suivent généralement les conventions de nommage suivantes :

lo : boucle locale (interface de loopback) : elle sert à manipuler la machine comme une machine.

eth0, eth1, ... : interface ethernet ou les cartes wifi intel.

wlan0, ... : cartes wifi.

vboxnet0, ... : interface réseau pour communiquer avec une machine virtuelle.

B.4. Signification des lignes dans le fichier nova.conf

sql_connection=mysql://nova@172.16.0.1/nova: c'est là où se situe la base de données nova.

s3_host=172.16.0.1: indique où se situe le service qui contient les images des machines virtuelles.

rabbit_host=172.16.0.1: là où se situe le serveur de messagerie rabbit AMQP.

ec2_url: là où se situe l'interface nova-api.

ec2_host=172.16.0.1: désigne l'adresse IP externe du service nova-api.

ec2_dmz_host=172.16.0.1: désigne l'adresse IP interne du service nova-api.

public_interface=eth1: c'est l'interface de l'hôte exécutant nova et que les clients utiliseront pour accéder aux instances.

image_service=nova.image.glance.GlanceImageService: précise que pour cette installation, glance sera utilisé pour la gestion de nos images.

glance_api_servers=172.16.0.1:9292: précise le serveur qui exécute le service d'imagerie glance.

auto_assign_floating_ip=true: indique que lorsqu'une instance est créée, elle reçoit automatiquement une adresse IP, affectée à partir d'un intervalle défini.

scheduler_default_filters=AllHostsFilter: précise que le scheduler peut envoyer des requêtes à tous les hôtes de calcul.

libvirt_type=qemu: définit le mode de virtualisation utilisé.

C. Références thématiques sous Ubuntu

Cette partie offre descriptif rapide des principales commandes sous Linux utilisées pour réaliser notre travail. Pour une référence complète des commandes en question, on peut utiliser `man` commande ou `info` commande (ou `help` commande s'il s'agit d'une commande intégrées à l'interpréteur).

C.1.Fonctions réseau

Ifconfig	configure les interfaces réseau et affiche leur état
ifdown	désactive une interface réseau
ifup	active une interface réseau
scp	transfère des fichiers d'une manière chiffrée via SSH
wget	télécharge des fichiers ou des répertoires via FTP/http
Ping	teste la connexion réseau vers un autre ordinateur
ssh	permet de se connecter à d'autres ordinateurs du réseau

D. Références thématiques sous la plateforme OpenStack

Commande	Description
keystone role-list	Affiche la liste des rôle
keystone tenant-list	Affiche la liste des tenants
keystone user-list	Affiche la liste des utilisateurs
keystone service-list	Affiche la liste des services
keystone endpoint-list	Affiche la liste des points d'accès
keystone user-delete ID_user	Supprime un utilisateur
keystone service-delete ID_service	Supprime un service
glance details	Affiche des informations détaillées sur toutes les images
glance image-list ou bien glance index	donnent une liste des images
nova image-list	Liste les images disque fournies par le service glance
nova flavor-list	Affiche la liste des types d'instances disponibles
nova-list	Affiche la liste des instances
nova show nom_de_l'instance	Affiche les propriétés d'une instance spécifiée
nova volume-list	Affiche la liste des volumes
nova secgroup-list-rules default	Liste les règles de ports autorisés sur le groupe de règles "default"
nova keypair-list	Affiche la list des clés créées
nova-manage network list	Affiche le réseau privé crée
nova-manage floating list	Affiche le réseau publique crée
nova-manage fixed list	Affiche la liste des adresses privés créées

E. Sources d'information pour l'installation

Ubuntu :

Site web d'Ubuntu <http://www.ubuntu.com>

Communauté francophone <http://www.ubuntu-fr.org/>

Manuel d'installation [http:// \[Tutoriel\] Installer Ubuntu 12.04 LTS, The Precise Pangolin Benedictux.mht](http:// [Tutoriel] Installer Ubuntu 12.04 LTS, The Precise Pangolin Benedictux.mht)

Mysql:

<http:// install\mysql - Documentation Ubuntu Francophone.mht>

Virtualbox:

Site web : <http://www.virtualbox.org>

OpenStack

Kevin Jackson, OpenStack Cloud Computing Cookbook, ISBN 978-1-84951-732-4, September 2012

<http:// www.packtpub.com>

http://docs.openstack.org/folsom/basic-install/content/basic-install_controller.html

<http:// http://docs.obn.me/Guide-Installation-OpenStack-Folsom-Noeud-Unique/>

<http:// Installation d'OpenStack - Infrastructure - Espace de travail Another Service>

https://github.com/mseknibilel/OpenStack-Folsom-Install-guide/blob/master/OpenStack_Folsom_Install_Guide_WebVersion.rst

<https://github.com/obuisson/Guide-Installation-OpenStack-Folsom-Noeud-Unique/blob/master/source/index.rst>

Résumé

Trouver une solution de Cloud Computing appropriée à un environnement spécifique et la mettre en oeuvre est une étape importante dans le processus de Cloud. Quelque soit la solution choisie, celle-ci a ses avantages et ses inconvénients. L'important est de choisir une solution dite bonne et qui convient le plus à l'environnement sur lequel nous voulons la mettre en place. Grâce à l'étude bibliographique, certains points essentiels à la compréhension du concept Cloud Computing ont été abordés, en particulier les solutions Open Source les plus convoitées et notamment la solution OpenStack qui a été retenue pour être mise en place. Les démarches d'installation de la solution Open Source OpenStack ainsi la configuration de ses composants et la mise en oeuvre de Dashboard comme outil de gestion de l'environnement sont détaillés dans ce mémoire.

Mots clés: Cloud Computing, virtualisation, cloud privé, centre de données, noeud de calcul, OpenStack, Eucalyptus, OpenNebula.

Abstract

Finding the suitable Cloud Computing solution for a specific environment and implement is an important step in the process of Cloud Computing. whatever the solution of Cloud Computing chosen, this one have its advantages and its disadvantages. The important is to choose the solution that suits for the environment in wich we want to implement this solution. Through the literature review, some key to understanding the concept of Cloud Computing issues were discussed, in particular Open Source solutions the most coveted including the OpenStack solution chosen to be implemented. The installation process of the solution Open Source OpenStack so the configuration of thus component and the implementation of Dashboard as a management tool of the environment are detailed throughout the memory.

Keywords: Cloud Computing, virtualization, private cloud, datacenter, calcul node, OpenStack, Eucalyptus, OpenNebula.