

Rapport projet de fin d'étude Licence Informatique SMI

Mise en place d'un système Cloud Computing privé



Encadré par :

Mr Abdeslam El FERGOUGUI

Réalisé par :

Mekdad Yassine

Jury :

Mr Abdelbaki ELBELRHITI ELALAOUI

Mr Mehdi ELISMAILI ALAOUI

Remerciements

Au terme de ce travail, nous saisissons cette occasion pour exprimer nos vifs remerciements à toute personne ayant contribué, de près ou de loin, à la réalisation de ce travail.

*Avant d'entamer ce rapport, Nous tenons à remercier dans un premier temps, **Mr EL FERGOUGUI Abdeslam**, notre professeur encadrant qui n'a pas cessé de nous encourager pendant la durée du projet, ainsi pour sa générosité en matière de formation et d'encadrement. Nous le remercions également pour la confiance qu'il nous a témoignée.*

*Nous exprimons également notre gratitude aux membres du jury **Mr A.ELBELRHITI EL ALAOUI** et **Mr M.ELISMAILI ALAOUI** qui nous ont honorés en acceptant de juger ce modeste travail.*

*Nous tenons à remercier nos professeurs de **la Faculté des Sciences de Meknès**, et le corps professoral de la filière **Sciences mathématiques et informatique**, de nous avoir incités à travailler en mettant à notre disposition leurs expériences et leurs compétences.*

Sommaire

| | |
|--|-----------|
| Introduction | 4 |
| Chapitre I : Les fondamentaux du Cloud Computing | 6 |
| 1. Introduction | 6 |
| 2. Technologies habilitantes. | 11 |
| 3. Les traits du Cloud Computing. | 12 |
| 4. Les plates-formes du Cloud Computing | 13 |
| 5. Les défis du Cloud Computing | 14 |
| 6. Cloud Computing dans le future. | 15 |
| Chapitre II : Sécurité du Cloud Computing. | 17 |
| 1. Les problèmes de sécurité en Cloud Computing | 17 |
| 2. Architecture du Cloud. | 19 |
| 3. Les exigences de sécurité dans le Cloud. | 20 |
| 4. La surveillance de sécurité du Cloud. | 24 |
| Chapitre III : Les systèmes libres (Open) du Cloud Computing | 26 |
| 1. CloudStack | 27 |
| 2. Eucalyptus Cloud. | 29 |
| 3. OpenNebula | 32 |
| 4. OpenStack | 33 |
| Chapitre IV : Etude d'un Système OpenSource du CC (OpenStack) | 38 |
| 1. Etude générale du système..... | 38 |
| 1.1 Présentation : | 38 |
| 1.2 Architecture : | 42 |
| 1.3 Installation : | 48 |
| 2. Test du bon fonctionnement..... | 50 |
| 2.1 Mode graphique : | 50 |
| 2.2 Mode commande : | 54 |
| Conclusion..... | 56 |
| Annexes | 57 |
| Bibliographie | 59 |
| Webographie | 60 |

Introduction

Cloud Computing est devenu une excellente solution pour fournir un flexible service à la demande, et une infrastructure informatique dynamique évolutive pour de nombreuses applications. Cloud Computing présente également des importantes tendances technologiques, et il est déjà évident qu'il est en train de transformer les processus de technologie de l'information et le marché informatique.

Le Cloud Computing, ou informatique en nuage, est, outre ses nouvelles spécificités techniques, un nouveau concept. Cette nouvelle façon de penser et de concevoir le rapport homme/machine s'inscrit dans un cycle bien plus large : pendant longtemps la machine n'était qu'une simple interface de visualisation de l'information et tout son traitement était internalisé (l'ère du Minitel), puis les machines ont embarqué de la puissance de calcul et de la mémoire et les calculs ont alors été externalisés et effectués en local ([voir Figure 1.1](#)). Aujourd'hui, nous nous dirigeons, avec l'émergence des tablettes tactiles, vers un retour à la notion d'interface de travail. Logiciels et données sont dans la majeure partie hébergés dans des serveurs privés ou communautaires à travers le monde.

Les capacités informatiques concernées par le Cloud Computing sont variées : capacité de calcul, espace de stockage, bande passante ou encore logiciels. Les applications proposées en mode Cloud Computing ne se trouvent plus forcément sur un serveur informatique hébergé chez l'utilisateur mais dans un « Cloud » formé de l'interconnexion de serveurs géographiquement distincts réalisée au niveau de fermes de serveurs géantes (également appelées Data Center). Ceci est rendu possible par le procédé de [virtualisation](#) qui consiste à faire fonctionner plusieurs systèmes d'exploitation ainsi que leurs applications associées sur un seul serveur physique. La

virtualisation permet ainsi de recréer plusieurs ordinateurs virtuels sur une seule et même machine physique.

Notre projet consiste à établir la mise en place d'un système Cloud computing privé au sein de la faculté dans le but de faciliter le travail pour les étudiants, les enseignants et tous les employés qui appartiennent à notre faculté des sciences.

Le présent travail s'articule autour de quatre chapitres : Dans [le premier chapitre](#), nous expliquons quelques notions fondamentales et généralités à propos du Cloud Computing, Les services offerts ainsi les différents types du Cloud computing, n'oubliant pas les défis de cette technologie. Dans [le deuxième chapitre](#) nous abordons la relation entre le cloud computing et la sécurité informatique qui représente le thème de recherche le plus important, nous introduisons aussi les problèmes de sécurité liés au cloud et les exigences de sécurité. Dans [le troisième chapitre](#), nous menons une étude exhaustive sur les alternatives open-sources des plateformes du Cloud Computing, Ceci nous a permis d'avoir et de présenter une idée riche sur les techniques disponibles pour la création d'un environnement du Cloud et le choix de notre plateforme. Et enfin dans [le quatrième chapitre](#) nous abordons une étude générale de notre solution contenant une présentation puis son architecture et l'installation. Ensuite nous faisons le test du bon fonctionnement avec les deux modes (graphique et lignes de commandes).

Chapitre I : Les fondamentaux du Cloud Computing

I.1 Introduction

Cloud computing peut être définie comme un nouveau style de l'informatique dans lequel les ressources de manière dynamique évolutive et souvent virtualisées sont fournies comme un service sur Internet. Cloud computing est devenu une tendance technologique majeure, et de nombreux experts s'attendent à ce que le cloud computing va remodeler les technologies de l'information (IT) et les processus du marché de l'IT. Grâce à la technologie de cloud computing, les utilisateurs utilisent une variété de dispositifs, y compris les PC, ordinateurs portables, smartphones, PDA (*Personal Digital Assistant*) et aux programmes d'accès, de stockage, et les plates-formes de développement d'applications sur Internet, via les services offerts par les fournisseurs de Cloud computing. Les avantages de la technologie de Cloud computing comprennent des économies de coûts, la haute disponibilité et l'évolutivité facile. Lorsqu'on parle des nouvelles technologies et de l'Internet, beaucoup d'entre nous ont déjà entendu ces termes de [cloud computing](#), [virtualisation](#) et [dématérialisation](#). Nul doute que la future génération d'Internet (plus connue par Web 2.0) aura un effet particulier sur la manière avec laquelle nous interagissons et communiquons entre nous, aussi bien pour les particuliers que pour les professionnels.

Ce concept qu'on appelle "cloud computing" peut changer notre façon de travailler et commence déjà à s'installer. Tout d'abord n'essayons pas de lui trouver une traduction en français, car personne n'oserait vendre de "l'informatique dans les nuages". En quelques mots, c'est la capacité d'utiliser des ressources et des outils via Internet, sans en avoir la possession ou en être à proximité, il suffit simplement de pouvoir y accéder.

Phases

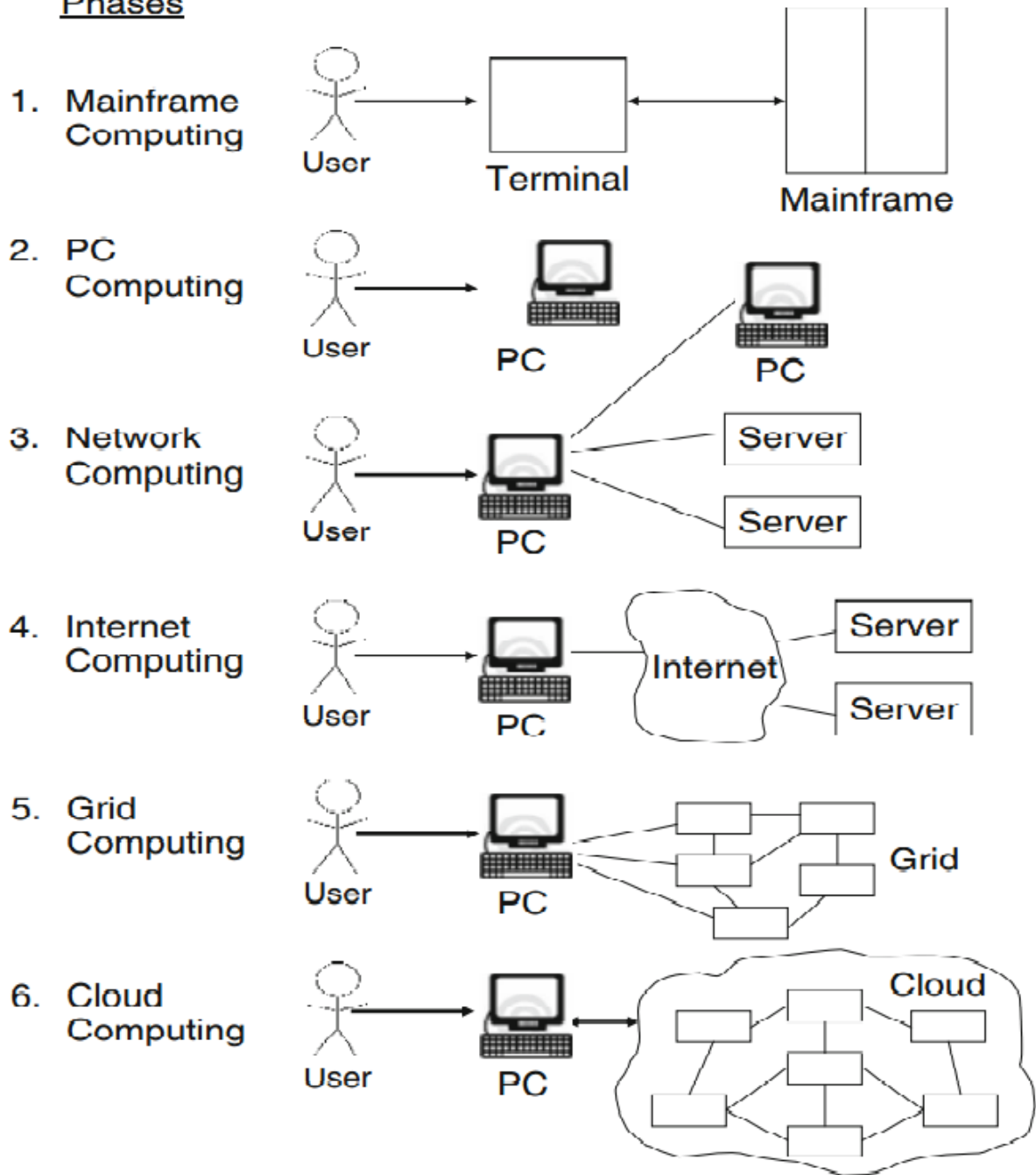


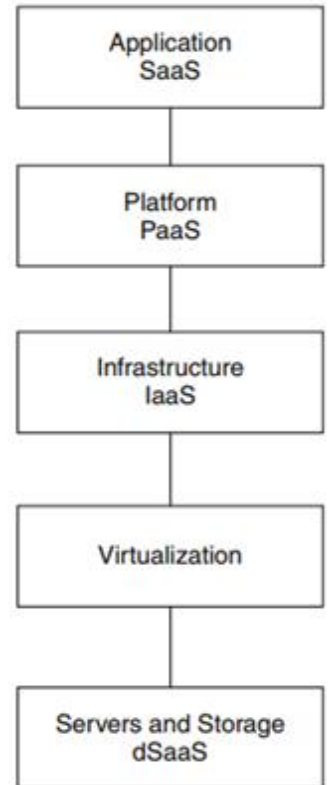
Figure 1.1 : Six paradigmes informatiques - du mainframe à Internet informatique, à la grille informatique et le Cloud Computing (adapté de Voas et Zhang (2009))

I.1.1-Les couches du Cloud Computing

Cloud computing peut être considéré comme un ensemble de services, ce qui peut être présenté comme une architecture de cloud computing en couches (voir figure 1.2).

Figure : 1.2

Architecture en couches du Cloud Computing



Cloud computing, SaaS, PaaS, IaaS ...

1. **Infrastructure as a Service (IaaS)**, l'accès de l'utilisateur à une infrastructure matérielle sur sa demande.
2. **Platform as a Service (PaaS)**, un environnement de développement via une infrastructure Cloud.
3. **Software as a Service (SaaS)**, une application informatique que l'utilisateur utilise en ligne et sans installation locale.

Infrastructure as a Service (IaaS), c'est un service qui libère l'entreprise de la nécessité de posséder (gérer, maintenir et contrôler, ...) ses propres serveurs et autres infrastructures de traitement de données. Il suffira d'un ordinateur ou un portable connecté à Internet pour lui garantir sa fonctionnalité commerciale. Cet accès est une forme d'abonnement auprès d'entités qui commercialisent ces services de virtualisation des matériels. La réduction des coûts d'acquisition, de maintenance et de recyclage des équipements est importante pour les entreprises. Le risque des pannes et de l'interruption d'activité est aussi minimisé, vu la très haute fiabilité de ce genre de services en ligne. Au final, c'est un gain de temps et d'argent considérable pour les clients de cette forme de cloud computing.

Platform as a Service (PaaS) est aussi appelé "*cloudware*". Cet autre service de virtualisation concerne la mise à disposition au profit des créateurs d'applications et de services web, d'outils de développement en ligne. Tout comme l'IaaS, on ne s'intéresse pas au fonctionnement interne de la "machine", c'est une solution qui permet aux utilisateurs de réduire leurs coûts et d'accéder à la demande, aux outils dont ils auront besoin. Les créateurs d'application pourront utiliser un environnement de développement en ligne sans se préoccuper des serveurs, du matériel, des licences de logiciels ou des bases de données.

Le service de Software as a Service (SaaS) est défini comme un abonnement d'accès à des applications par Internet. L'accès se fait en ligne et ne nécessite pas d'installation sur l'ordinateur local. Beaucoup d'entre nous ont sans doute utilisé ces sites où on peut envoyer et partager photos et vidéos avec ses amis et sa famille. Ces solutions sont de plus en plus populaires, car les utilisateurs n'ont pas besoin d'installer et mettre à jour un logiciel chez eux, il leur suffit de se connecter et de l'utiliser quand ils ont en besoin.

Les données-Storage-as-a-Service (dSaaS) fournissent le stockage que le consommateur a utilisé, y compris les exigences de bande passante pour le stockage.

I.1.2 - Les types du Cloud Computing

Il existe trois types du Cloud Computing (voir figure 1.3):

- Cloud public
- Cloud privé
- Cloud hybride

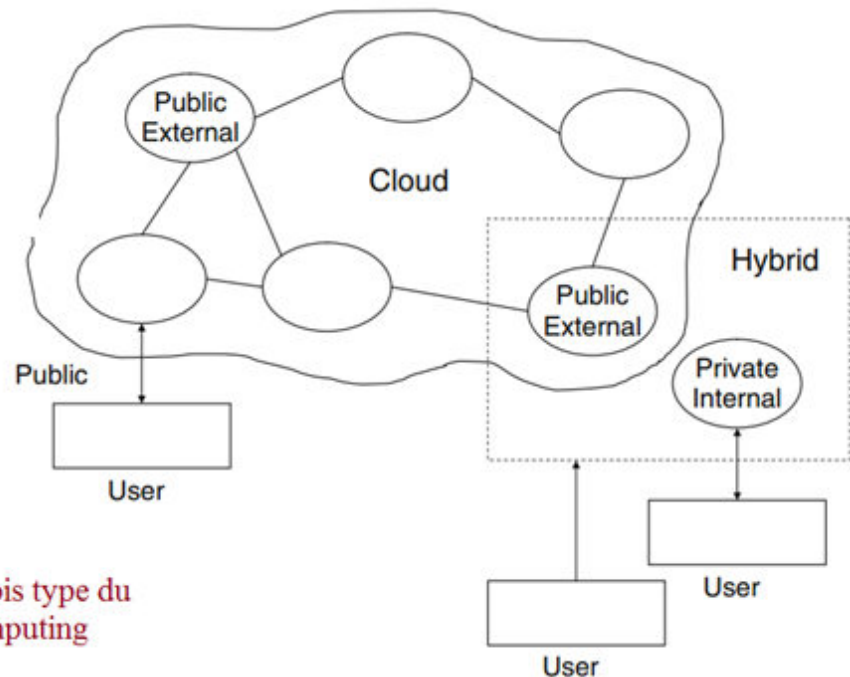


Figure 1.3 : Trois type du Cloud Computing

Dans le cloud public (ou cloud externe) les ressources informatiques sont provisionnées dynamiquement sur Internet via des applications Web ou des services Web à partir d'un fournisseur tiers hors site. Les Clouds publics sont gérés par des tiers, et les demandes de différents clients sont susceptibles d'être mélangés sur les serveurs, systèmes de stockage et les réseaux du Cloud.

Le cloud privé (ou cloud interne) se réfère au Cloud Computing sur les réseaux privés. Les clouds privés sont construits à l'usage exclusif d'un client, offrant un contrôle total sur les données, la sécurité et la qualité de service. Les clouds privés peuvent être construits et gérés par la propre organisation informatique d'une entreprise ou par un fournisseur de Cloud.

Un environnement de Cloud hybride combine plusieurs modèles de Cloud public et privé. Les Clouds hybrides introduisent la complexité de déterminer comment distribuer des applications sur un Cloud à la fois public et privé.

I.2 Technologie habilitantes

Technologies clés qui ont permis à l'informatique en nuage sont décrits dans la présente section; ils comprennent la virtualisation, le service Web et l'architecture orientée services, les flux de services et les flux de travail, et le Web 2.0 et mashup.

I.2.1 - La virtualisation

L'avantage du cloud computing est la capacité de virtualiser et partager des ressources entre les différentes applications avec l'objectif d'une meilleure utilisation des serveurs. Dans le Cloud, les serveurs peuvent être partagés, ou virtualisés, des systèmes d'exploitation et les applications qui en découlent en moins de serveurs (dans l'exemple spécifique de deux serveurs).

Les technologies de virtualisation incluent des techniques de machines virtuelles telles que VMware et Xen, et des réseaux virtuels, tels que le VPN. Les machines virtuelles fournissent des IT infrastructures virtualisées à la demande, tandis que les utilisateurs de soutien des réseaux virtuels avec un environnement de réseau personnalisé d'accès aux ressources de Cloud Computing.

I.2.2 - Le service Web et l'architecture orientée services

Le service Web et l'architecture orientée services (SOA) ne sont pas de nouveaux concepts; mais ils représentent les technologies de base pour le Cloud Computing. Les services Cloud sont généralement conçus comme des services Web, qui suivent des normes de l'industrie, y compris WSDL, SOAP, UDDI. Une architecture orientée service organise et gère les services Web dans les Clouds (VOUK, 2008). Une SOA comprend également un ensemble de services de Cloud Computing, qui sont disponibles sur différentes plates-formes distribuées.

I.2.3 - Les flux de services et les flux de travail

La notion de flux de services et flux de travail se réfère à une vision intégrée des activités de services basés prévue dans les Clouds. Les flux de travail sont devenus l'un des domaines de recherche importants dans le domaine des systèmes (VOUK, 2008) la base de données et d'information.

I.2.4 - Le Web 2.0 et Mashup

Web 2.0 est un nouveau concept qui fait référence à l'utilisation de la technologie Web et la conception Web pour améliorer la créativité, le partage d'informations et la collaboration entre les utilisateurs (Wang, Tao, & Kunze, 2008). D'autre part, Mashup est une application Web qui combine des données à partir de plus d'une source dans un outil de stockage intégré unique. Ces deux technologies sont très bénéfiques pour le Cloud Computing.

I.3 Les traits du Cloud Computing

Le Cloud Computing apporte un certain nombre de nouvelles fonctionnalités par rapport aux autres paradigmes d'informatique. Ils sont brièvement décrits dans cette section :

- ✓ Services d'évolutivité et à la demande

Cloud Computing offre des ressources et des services pour les utilisateurs à la demande. Les ressources s'adaptent sur plusieurs centres de données.

- ✓ Interface utilisateur -centrique

Les Interfaces des Clouds sont des locations indépendantes et peuvent être bien accessibles par des interfaces établies telles que les services Web et les navigateurs Internet.

- ✓ Qualité de service garantie (QoS)

Cloud Computing peut garantir la qualité de service pour les utilisateurs en termes de matériel / CPU performances, largeur de bande, et la capacité de mémoire.

- ✓ Système autonome

Les systèmes de cloud computing sont des systèmes autonomes gérés de manière transparente pour les utilisateurs. Cependant, les logiciels et les données à l'intérieur des Clouds peuvent être automatiquement reconfigurés et consolidés à une simple plateforme en fonction des besoins de l'utilisateur.

- ✓ Prix

Cloud computing ne nécessite pas d'investissements -. Aucune dépense en capital n'est requise. Les utilisateurs paient pour les services et les capacités dont ils ont besoin.

I.4 Les plates-formes du Cloud Computing

Cloud computing a un grand potentiel commercial. Selon le cabinet d'études de marché IDC (International Data Corporation), les dépenses mondiales dans les services de Cloud public (IaaS, PaaS et SaaS) devraient atteindre **47,2 milliards de dollars**. Et pour le cabinet, la croissance sur ce marché est là pour durer : 23,5% par an en moyenne entre 2013 et 2017.

I.4.1 - Etablissement des prix

Prix des plates-formes et services Cloud repose sur trois dimensions clés :

- **Le stockage** : qui est généralement mesurée par la quantité quotidienne moyenne de données stockées en Go sur une période mensuelle.
- **La bande passante** : est mesuré par le calcul de la quantité totale de données transférées dans et hors de service de plate-forme à travers transaction et le traitement par lots. En général, le transfert de données entre les services au sein de la même plate-forme est disponible gratuitement dans de nombreuses plates-formes.
- **Temps de calcul** : est mesurée en unités de temps nécessaires pour exécuter une instance, ou de l'application, ou la machine à des demandes de service.

En résumé, en analysant le coût du Cloud Computing, en fonction des caractéristiques de l'application du coût de déploiement d'une application peut varier en fonction de la plate-forme sélectionnée.

I.4.2 - Les composants du Cloud Computing et ses fournisseurs

Les principaux éléments constitutifs de plates-formes de cloud computing comprennent le matériel informatique, le stockage, l'infrastructure, logiciels, systèmes d'exploitation, et la plate-forme de virtualisation. Les principaux fournisseurs offrant les composants du Cloud Computing sont présentés dans le tableau.

| Cloud computing components | Vendors |
|----------------------------|---|
| Computer hardware | Dell, HP, IBM, Sun |
| Storage | Sun, EMC, IBM |
| Infrastructure | Cisco, Juniper Networks, Brocade Communication |
| Computer software | 3tera. Eucalyptus. G-Eclipse. Hadoop |
| Operating systems | Solaris, AIX, Linux (Red Hat, Ubuntu) |
| Platform virtualization | Citrix, VMWare, IBM, Xen, Linux KVM, Microsoft, Sun xVM |

I.5 Les défis du Cloud Computing

En résumé, le nouveau paradigme du cloud computing offre un certain nombre de bénéfices et avantages par rapport aux paradigmes informatiques précédents et de nombreuses organisations l'adoptent. Cependant, il ya encore un certain nombre de défis, qui sont actuellement traitée par les chercheurs et les praticiens dans le domaine (Leavitt, 2009). Ils sont présentés brièvement ci-dessous.

I.5.1 - La performance

Le problème majeur de la performance peut être pour certaines applications de transaction orientée intensives et autres de données intensives, dans lequel le Cloud computing peut manquer une performance adéquate. En outre, les utilisateurs qui sont à une grande distance de fournisseurs du Cloud peuvent éprouver une latence élevée et des retards.

I.5.2 - Sécurité et confidentialité

Les entreprises sont toujours préoccupées par la sécurité lors de l'utilisation du cloud computing. Les clients sont préoccupés par la vulnérabilité aux attaques, lorsque les informations et les ressources informatiques critiques sont à l'extérieur du pare-feu. La solution pour la sécurité suppose que les fournisseurs du cloud computing suivent les pratiques de sécurité standard, comme décrit dans [le chapitre II](#)

I.5.3 - Le contrôle

Certains départements d'IT sont concernés parce que les fournisseurs du Cloud Computing ont le contrôle total des plates-formes. Les fournisseurs du Cloud computing ne conçoivent généralement pas des plates-formes pour les entreprises spécifiques et leurs pratiques commerciales.

I.5.4 - Coûts de bande passante

Avec le Cloud Computing, les entreprises peuvent économiser de l'argent sur le matériel et le logiciel. Cependant ils pourraient encourir la hausse des charges de bande passante réseau. Coût de la bande passante peut être faible pour des plus petites applications basées sur Internet, qui ne sont pas de données intensives, mais il pourrait s'augmenter de manière significative pour les applications de données intensives.

I.6 Cloud Computing dans le future

En résumé, le cloud computing est certainement un type de paradigme informatique / architecture qui restera pour une longue période à venir. Dans un proche avenir, le cloud computing peut émerger dans différentes directions. Un scénario possible pour l'avenir est que l'entreprise peut utiliser un cloud hybride distribuée comme illustré sur La [Figure 1.4](#) ci-dessous :

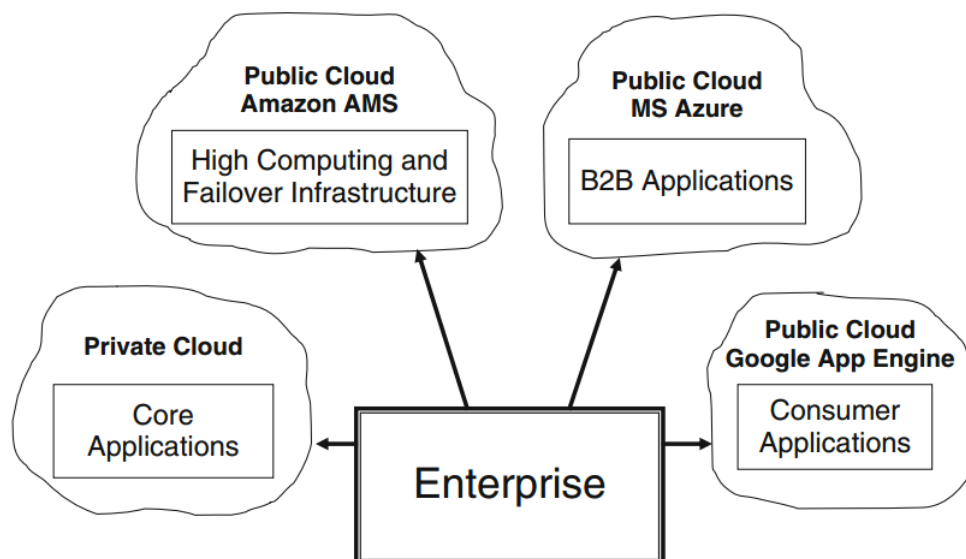


Figure 1.4 : L'architecture distribuée d'hybride Cloud (adapté par Lakshmanan (2009))

Selon ce scénario, l'entreprise va utiliser les applications de base sur son Cloud privé, tandis que d'autres applications seront distribuées sur plusieurs Cloud privés et publics qui sont optimisés pour des applications spécifiques.

Chapitre II : La sécurité du Cloud Computing

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

II.1 Les problèmes de sécurité en Cloud Computing

Pour commencer, nous allons rappeler quelques problèmes de sécurité en Cloud Computing :

- **Disponibilité du réseau**: La valeur du cloud computing ne peut être réalisé que lorsque la connectivité réseau et la bande passante répondent aux minimum des besoins. Le Cloud doit être disponible chaque fois que nous en avons besoin. Si ce n'est pas le cas, les conséquences ne sont pas différentes d'une situation de déni de service.
- **Reprise après le problème et continuité des activités** : Les locataires et les utilisateurs doivent avoir confiance que leurs activités et leurs services continueront si l'environnement de production du fournisseur de cloud est soumis à une catastrophe.
- **Les incidents de sécurité** : Les locataires et les utilisateurs doivent être informés de manière appropriée par le fournisseur en cas d'incident. Les locataires ou les utilisateurs peuvent avoir besoin du soutien du fournisseur pour répondre à la vérification ou de l'évaluation des résultats. En outre, un fournisseur peut ne pas avoir la possibilité d'offrir un soutien suffisant aux locataires ou aux utilisateurs pour résoudre les enquêtes.
- **Perte de contrôle physique** : Depuis les locataires et les utilisateurs perdent le contrôle physique sur leurs données et applications, il en résulte une série de préoccupations :

- Confidentialité et données : Avec les clouds publics, les données peuvent ne pas rester dans le même système, ce qui soulève plusieurs questions juridiques.
- Données de l'utilisateur ou de l'organisation peuvent être mélangés de différentes manières avec des données appartenant à d'autres ...
- Un administrateur locataire a limité la portée de contrôle et de responsabilité au sein d'une infrastructure-as-a-Service (IaaS) mise en œuvre en commun, et encore moins avec une plate-forme-comme-un service (PaaS) un. Les locataires doivent avoir confiance que le fournisseur offrira un contrôle approprié, tout en reconnaissant que les locataires devront simplement adapter leurs attentes pour un niveau de contrôle qui est raisonnable dans ces modèles.
- **Nouveaux risques et nouvelles vulnérabilités** : Il est à craindre que le Cloud computing apporte de nouvelles classes de risques et des vulnérabilités Bien que tous les logiciels, le matériel et l'équipement de réseau soient soumis aux déterrements à cause de nouvelles vulnérabilités, en appliquant une sécurité multicouche et les processus opérationnels bien conçus, un Cloud peut être protégé contre les types les plus courants d'attaque, même si certaines de ses composantes sont intrinsèquement vulnérables.

Dans le domaine du Cloud Computing, les craintes concernant les pertes de données et aux failles de sécurité sont nombreuses. Elles sont motivées par la virtualisation de serveur et l'hébergement des données à distance. L'idée de ne plus maintenir physiquement le matériel constitue pour une entreprise le principal frein à l'utilisation du Cloud Computing. Pour cela, les fournisseurs ont mis en place diverses mesures visant à garantir aux utilisateurs un haut degré de sécurité des données en Cloud Computing. Le chiffrement des données, la surveillance logicielle et la sécurisation des lieux de stockage en font notamment partie.

II.2 Architecture du cloud

Un but pour l'architecture est qu'elle doit être adaptée pour répondre aux besoins. Cette partie examine les exigences architecturales clés pour une mise en œuvre typique du Cloud.

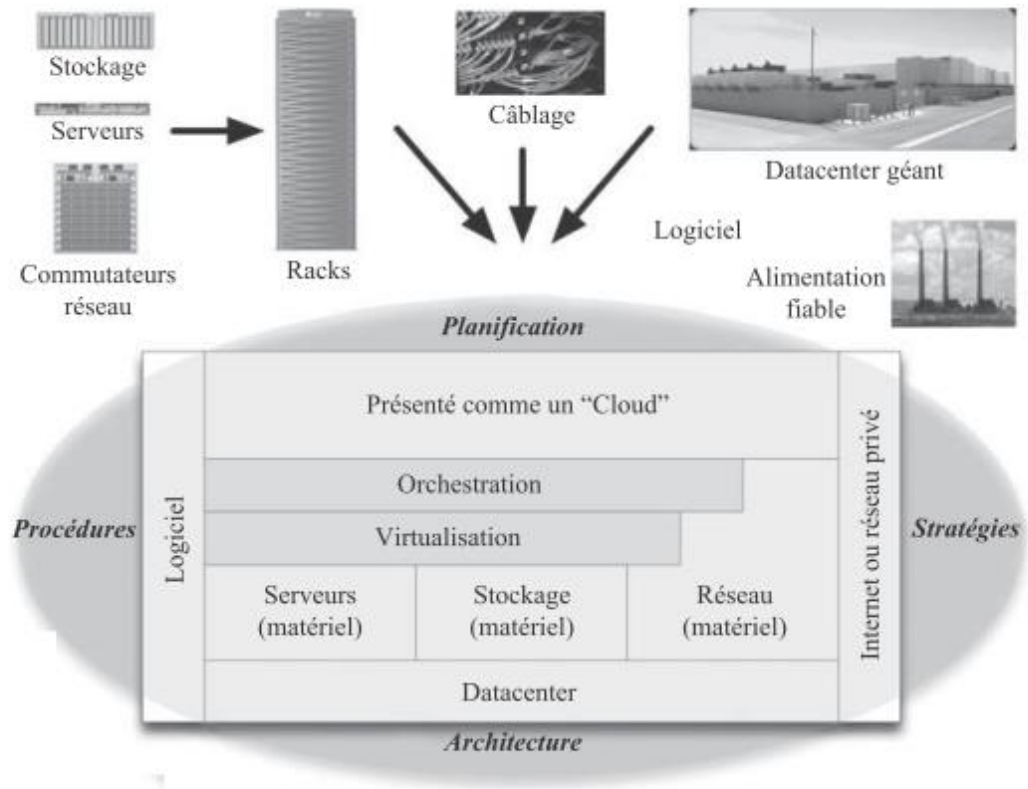


Figure 2.1 : Architecture et implémentation d'un Cloud

L'exploitation sécurisée et efficace d'un Cloud implique une planification préalable importante. À un niveau élevé, nous avons un Data center et des connexions Internet redondantes qui donnent accès à l'entrée d'un Cloud. Cette entrée représente la partie technologique d'un périmètre de sécurité des informations constituée de périphériques réseau qui apportent des communications sécurisées.

À l'intérieur de ce périmètre, nous trouvons de nombreux équipements placés en racks et câblés conformément à des modèles définis. Une infrastructure est également nécessaire pour gérer le Cloud et ses ressources pendant son fonctionnement. En allant plus loin, chaque composant (serveurs, stockage et réseau) demande un certain niveau

de configuration. Cette vue d'ensemble est celle de nombreux composants organisés en partie selon des modèles visuellement évidents. Lors de la conception ou de la planification d'un système complexe, il est important de voir plus loin et d'envisager les processus et les procédures qui seront nécessaires à son exploitation. Bien qu'il soit possible de construire un petit Cloud sans passer trop de temps en planification, tout Cloud de taille moins modeste exige une planification et une conception importantes. Tout manque de planification conduira inévitablement à des coûts récurrents plus élevés en raison de l'inefficacité de la conception et des procédures, ainsi que d'une exploitation décalée par rapport aux besoins de gestion d'un Cloud fortement dynamique. Une meilleure approche implique une architecture prudente qui tient compte d'un inévitable besoin d'évolution et conserve la souplesse nécessaire à des adaptations (voir Figure 2.1).

II.3 Les exigences de sécurité dans un Cloud

La sécurité physique :

La portée des questions en matière de sécurité physique est important, et il s'agit d'une série de mesures pour éviter, prévenir, détecter et répondre aux accès non autorisés à l'installation ou à des ressources ou de l'information dans l'établissement. La sécurité physique pour un établissement doit lui-même être considéré comme un système de protection, avec les éléments de sécurité individuels se complètent mutuellement dans une défense à multiples facettes et en couches. Ces éléments comprennent les aspects de **la conception de l'environnement, le contrôle d'accès** (y compris mécanique, électronique et procédurale), de **surveillances** (y compris la vidéo, thermique, la proximité avec des capteurs environnementaux), **l'identification des personnes et des contrôles d'accès, détection d'intrusion et en conjonction avec les systèmes d'intervention** (lumières, des portes et des zones de verrouillage).

II.3.1 Exigences de sécurité pour l'architecture :

Ces exigences sont :

- Coûts et ressources.
- Fiabilité.
- Performances.
- Le trio de la sécurité (La confidentialité, l'intégrité et disponibilité).
- Contraintes légales et réglementaires

II.3.2 Exigences générales sur la sécurité de l'infrastructure :

En plus des contraintes de sécurité du Cloud décrites précédemment dans cette section, il existe de nombreuses exigences plus générales pour [la sécurité de l'infrastructure](#) :

- Exploiter les meilleures pratiques des fournisseurs, car elles sont issues de l'expérience. Même si une bonne pratique n'est pas applicable, la connaître peut déjà se révéler bénéfique.

- Par défaut, les machines virtuelles doivent être endurcies et minimisées.

- Les ports ouverts doivent se limiter au strict minimum pour la mise à disposition et l'allocation initiale pour un client.

- La mise en œuvre des moyens qui permettent d'assurer la continuité de l'activité conformément aux accords de niveaux de service.

- Vérifier que les liens physiques qui entrent dans le bâtiment (et, à partir de là, dans l'infrastructure du Cloud) ne sont pas sujets à un point de défaillance unique en cas d'événement catastrophique.

- Vérifier que le bâtiment dispose d'une source d'alimentation suffisante et que sa distribution permette à l'infrastructure de secours d'entrer en action lorsque l'électricité est coupée dans une partie du bâtiment.

- Vérifiez que les adresses IP retirées internes au Cloud, comme celles affectées aux machines virtuelles d'un précédent locataire, soient suffisamment anciennes avant de les recycler chez un autre utilisateur. Cela permettra d'éviter que ce nouvel utilisateur n'accède aux ressources de l'utilisateur précédent.

- Attendez-vous à une innovation et à des évolutions permanentes dans l'informatique en nuage et dans les technologies sous-jacentes. Prévoyez de pouvoir

modifier, adapter ou étendre l'infrastructure d'une manière que vous n'aurez pas totalement anticipée.

II.3.3 Exigences de la gestion des identités :

Les identités constituent un élément clé de la sécurité. Ces informations doivent être correctes et disponibles à tous les éléments du Cloud qui ont des besoins de validation des accès.

Voici les exigences de base :

- Des contrôles doivent être mis en œuvre pour protéger la confidentialité, l'intégrité et la disponibilité des informations d'identité.
- Un système de gestion des identités doit être mis en place pour prendre en charge les besoins d'authentification du personnel du Cloud.
- Le système de gestion des identités doit pouvoir prendre en charge les besoins d'authentification à grande échelle des locataires et des utilisateurs du Cloud.
- L'identité des utilisateurs doit être vérifiée au moment de leur enregistrement conformément à la stratégie de sécurité et aux contraintes légales.
- Lorsque des identités sont supprimées, l'historique des utilisateurs doit être conservé pour d'éventuels besoins d'enquête ultérieure.
- Lorsque des identités d'utilisateurs sont retirées ou recyclées, vous devez vérifier que l'accès accordé à un nouvel utilisateur ne concerne pas les données ou les environnements des anciens utilisateurs, ni d'autres ressources d'information privées. Pour cela, il faut qu'au niveau approprié du système de gestion les identités des utilisateurs ne soient jamais réellement réutilisées, ce qui évitera tout conflit ou confusion.
- Des moyens doivent être mis en place pour que le personnel du fournisseur du Cloud puisse vérifier les déclarations d'identité des clients.

II.3.4 Exigences sur la gestion des accès :

Le contrôle des accès se base sur les informations d'identité pour permettre et contraindre l'accès à un Cloud en fonctionnement et à l'infrastructure sous-jacente.

Voici quelques exigences :

- De manière générale, le personnel du Cloud doit avoir un accès limité aux données des clients. Il peut avoir besoin d'un accès à l'hyperviseur d'une machine allouée à un client ou aux périphériques de stockage qui héberge les machines virtuelles ou les données du client, mais un tel accès doit être sévèrement encadré et se limiter à des opérations précises parfaitement définies dans la politique de sécurité.
- Interdisez l'usage de comptes partagés (comme celui de l'administrateur) et favorisez l'utilisation de sudo ou équivalent pour donner des privilèges consignés et uniquement aux membres du rôle approprié.
- Mettez en place une liste blanche d'adresses IP pour tous les contrôles ou les accès à distance du personnel d'exploitation. Lorsqu'une telle liste ne peut pas être établie, imposez un accès au travers de mécanismes supplémentaires, comme des passages renforcés au travers de proxies ou de passerelles.

II.3.5 Exigences sur la gestion des clés :

Dans un Cloud, le chiffrement constitue l'un des principaux moyens de protéger les données lorsqu'elles sont stockées et au cours des phases de stockage et de traitement. Voici les exigences quant à la gestion des clés :

- ✚ Des contrôles adaptés doivent être mis en place de manière à limiter l'accès au système de génération des clés sur lequel le fournisseur du Cloud garde le contrôle.

- ✚ Vérifiez que le niveau racine et les clés de signature sont gérés de manière appropriée.
- ✚ Des procédures doivent permettre le rétablissement lorsque des clés ont été compromises.
- ✚ Protégez et chiffrez les données des clients et les images des machines virtuelles à toutes les phases appropriées de leur cycle de vie.

II.4 La surveillance de la sécurité du Cloud

La surveillance de la sécurité se base sur la surveillance de la sécurité réseau (par une inspection du trafic avec snort ou autres) et sur des données environnementales (voir la partie “Sécurité physique”).

La surveillance de la sécurité doit être un service durci et hautement disponible auquel il est possible d’accéder en interne ou à distance de manière sécurisée.

Elle doit comprendre les éléments suivants :

- ✚ La génération d’alertes fondée sur la reconnaissance automatique de l’existence ou de la détection d’un événement de sécurité ou d’une situation critique.
- ✚ L’envoi d’alertes critiques par différents moyens pour que l’équipe de sécurité et la direction en soient informées promptement.
- ✚ Les moyens donnés au personnel de sécurité pour enquêter et suivre le déroulement d’un incident ou simplement pour examiner les journaux de manière à améliorer les mécanismes d’alerte ou à identifier manuellement des incidents de sécurité.
- ✚ Mettez en place un système de détection des intrusions ou d’une anomalie au niveau du Cloud et envisagez de le fournir en tant que service pour les locataires ou les utilisateurs

- ✚ Envisagez de fournir sous forme de PaaS (Platform-as-a-Service) ou d'IaaS (Infrastructure-as-a-Service) une fonction qui permette aux clients d'implémenter la détection des intrusions/anomalies et qui les autorise à envoyer des jeux d'événements ou des alertes appropriés au système de surveillance de la sécurité mis en place par le fournisseur du Cloud
- ✚ Vérifiez que la surveillance de la sécurité est implémentée de manière fiable et correcte même en cas de dysfonctionnement dans le processus de génération des événements et de leur collecte. Les journaux de sécurité doivent être conservés de manière compatible avec la loi, la réglementation applicable et la stratégie de sécurité.

Chapitre III : Les systèmes libres (Open) du Cloud Computing

Le *Cloud Computing* peut permettre d'effectuer des économies, notamment grâce à la mutualisation des services sur un grand nombre de clients. Certains analystes indiquent que 20 à 25 % d'économies pourraient être réalisées par les gouvernements sur leur budget informatique s'ils migraient vers le *Cloud Computing*. Comme pour la virtualisation, l'informatique dans le nuage peut être aussi intéressante pour le client grâce à son évolutivité. En effet, le coût dépend de la durée de l'utilisation du service rendu et ne nécessite aucun investissement préalable (homme ou machine). L'« élasticité » du nuage permet de fournir des services évolutifs et peut permettre de supporter des montées en charge. Inversement, le fournisseur a la maîtrise sur les investissements, est maître des tarifs et du catalogue des offres, et peut se rémunérer d'autant plus facilement que les clients sont captifs.

Tout comme système d'exploitation pour un ordinateur de bureau ou serveur, ce dernier gère les ressources qui lui sont associées, les systèmes Cloud gèrent les ressources associées avec le Cloud. La majorité des ressources de cloud computing sont **Compute, Storage & Network**. **Compute** comprend des serveurs et des hyperviseurs associés aux autres serveurs qui permettent la création des machines virtuelles. **Storage** comprend le stockage local, réseau de stockage et le stockage d'objets. **Network** comprend des vlans, pare-feu, les équilibreurs de charge, etc.... Les systèmes Cloud sont également chargés de donner les caractéristiques des Clouds comme l'élasticité, le modèle de provisionnement en libre-service, etc.... Ils sont également responsables d'autres éléments comme la gestion de l'image, l'authentification, la sécurité, la facturation, etc....

Dans ce chapitre on va citer quelques systèmes du Cloud Computing qui occupent une très grande place dans le monde de la technologie d'information ainsi un énorme nombre d'utilisateurs à travers le monde entier.

III.1 CloudStack

[Apache CloudStack](#) est un logiciel open source conçu pour déployer et gérer de grands réseaux de machines virtuelles, comme une infrastructure hautement évolutive et disponible en tant que service (IaaS). CloudStack est utilisé par un certain nombre de fournisseurs de services pour offrir des services de cloud public, et par de nombreuses entreprises afin de fournir une offre Cloud (privé) sur les locaux ou dans le cadre d'une solution de Cloud hybride.

[CloudStack](#) est une solution clé en main qui comprend l'ensemble de caractéristiques, la plupart des organisations veulent avec un Cloud IaaS: orchestration des ordinateurs, Network-as-a-Service, l'utilisateur et la gestion de compte, une API native et ouverte, la comptabilité des ressources, et une Interface utilisateur de première classe (UI) ([voir Figure 3.1](#)).

[CloudStack](#) prend actuellement en charge les hyperviseurs les plus populaires: VMware, KVM, XenServer, Xen Cloud Platform (XCP) et Hyper-V.

Les utilisateurs peuvent gérer leur Cloud avec une interface facile à utiliser Web et des outils de ligne de commande. En outre, CloudStack fournit une API qui est compatible avec [AWS EC2](#) et [S3](#) pour les organisations qui souhaitent déployer des Cloud hybrides.

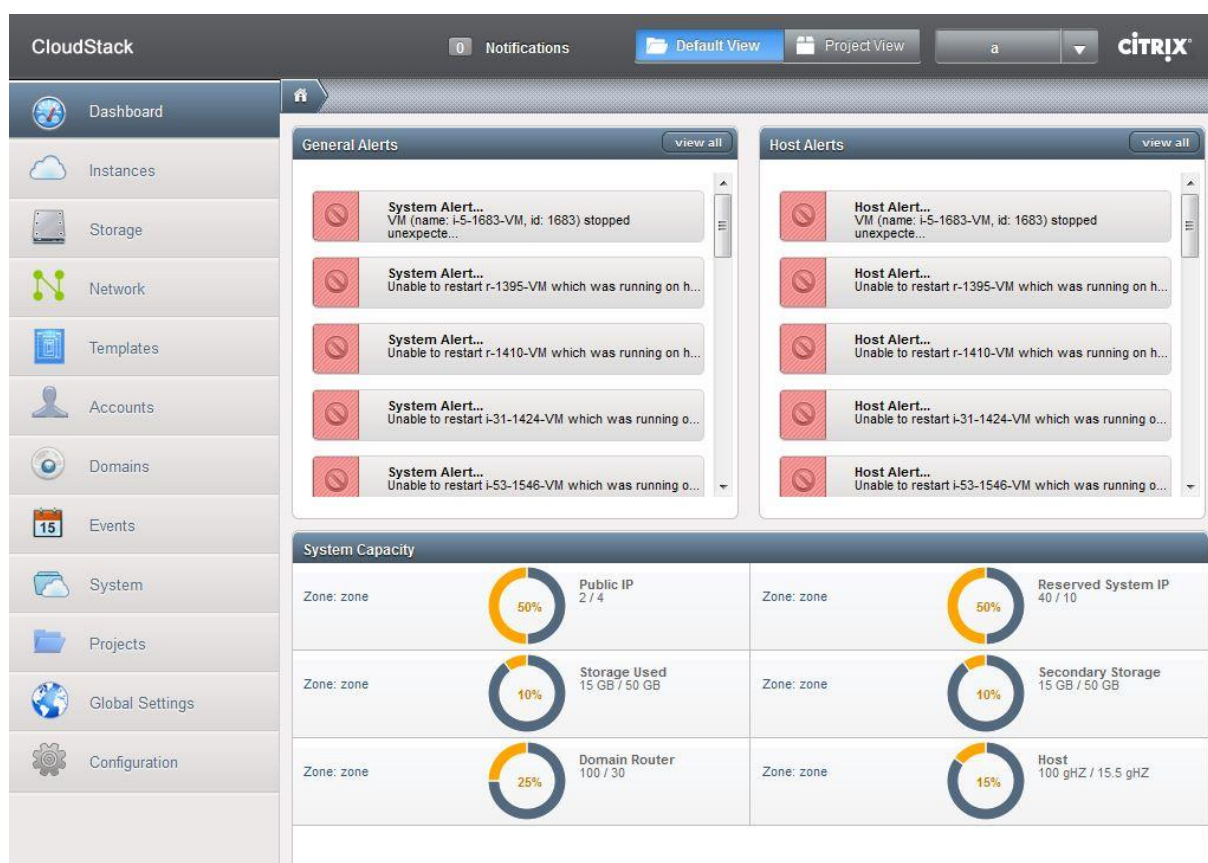


Figure 3.1 : Le tableau de bord de CloudStack

Interopérabilité API EC2

L'interopérabilité entre les infrastructures permet d'éviter la multiplicité des interfaces, les API permettant aux CloudStack server de communiquer avec des solutions externes (voir Figure 3.2).

En tant qu'utilisateur final, accédez au portail, à la ligne de commande et au service WEB par divers moyens.

Créez ensuite de manière programmatique des machines virtuelles, des projets, réseaux et applications.

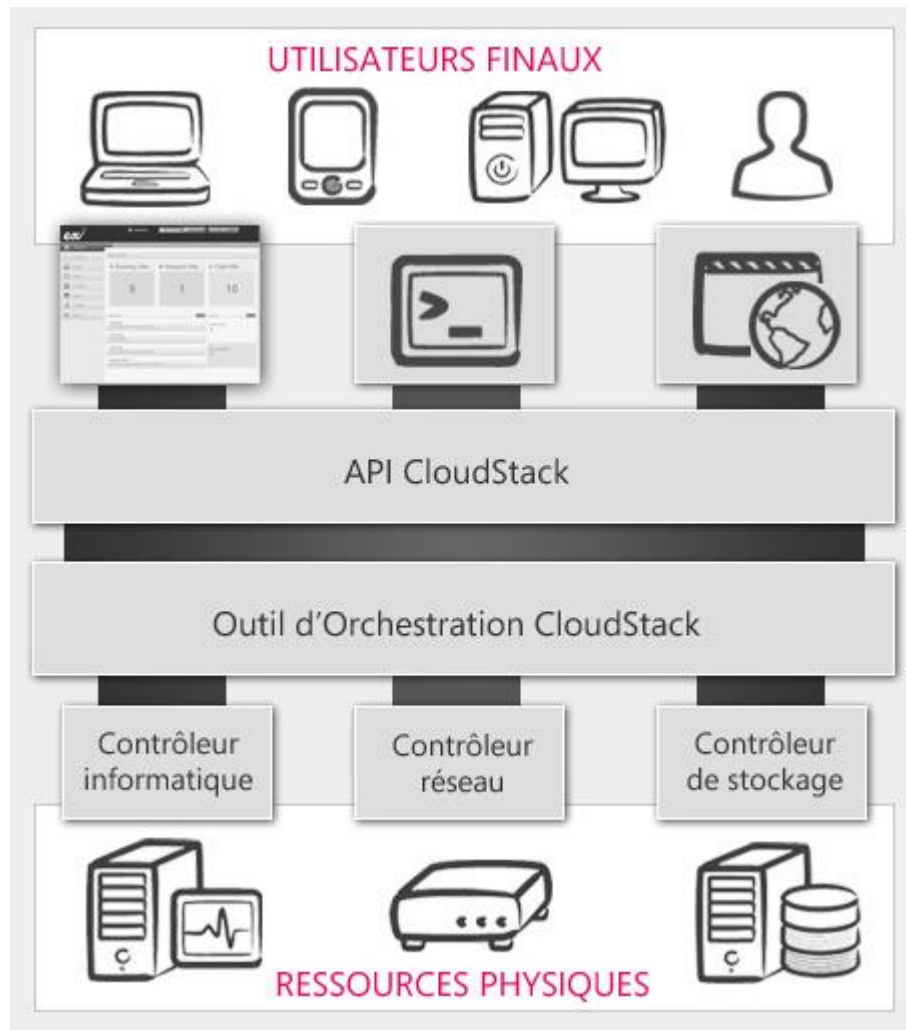


Figure 3.2 : Interopérabilité avec les API

III.2 Eucalyptus Cloud

Eucalyptus est un logiciel médiateur (anglais *middleware*) open-source pour l'implémentation du Cloud Computing sur une grappe de serveurs. Son nom fait référence à l'acronyme anglais « Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems » qui peut se traduire en « Utilitaire d'Architecture informatique élastique pour relier vos programmes à des systèmes fonctionnels ». **Eucalyptus** est compatible avec Amazon Web Services. Il est intégré dans la distribution Gnu/Linux Ubuntu 9.04 en tant qu'outils de « cloud computing ». *Eucalyptus* peut s'installer facilement sur la majorité des distributions Gnu/linux: *Debian*, *CentOS*, ...

Fonctionnalités :

- Compatible avec [Amazon Web Services API](#)
- Installation et déploiement avec l'outil de gestion de grappe de serveurs [Rocks Linux](#)
- Communication sécurisée entre les processus interne via [SOAP](#) (*Simple Object Access Protocol*) et [WS Security](#) (*Web Services Security*)
- Outils d'administration basique
- Capacité à configurer de multiples grappes de serveurs comme un seul « cloud »

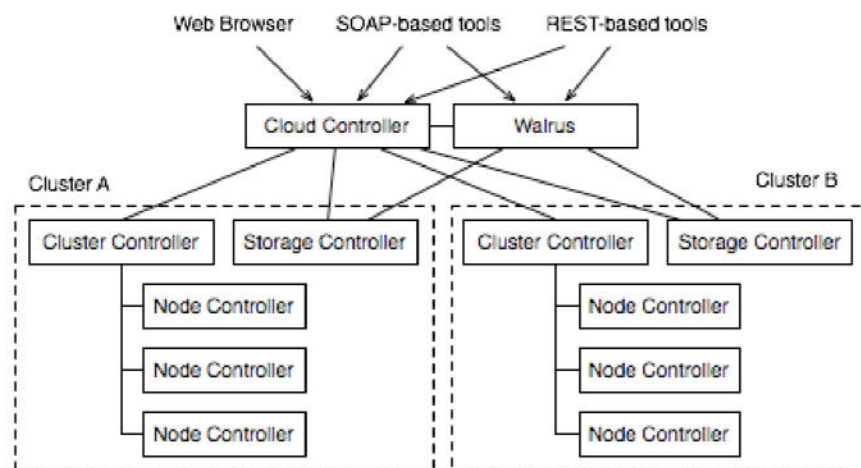


Figure 3.3 : L'architecture d'Eucalyptus Cloud

Architecture :

L'architecture d'Eucalyptus est composée de cinq éléments de haut-niveau : (voir Figure 3.3)

- [Cloud Controller](#) (CLC, contrôleur du Cloud) ;
- [Cluster Controller](#) (CC, contrôleur de grappe) ;
- [Walrus](#) (il implémente la scalabilité du stockage et comporte une interface pour ajouter ou récupérer des paquets et objets). Il fournit un mécanisme de persistance du stockage et de contrôle d'accès aux images de machines de la machine virtuelle et aux données de l'utilisateur) ;
- [Storage Controller](#) (SC, contrôleur de stockage) ;

- **Node Controller** (NC, contrôleur de nœud).

Chaque composant de haut-niveau du système a sa propre interface web et est implémenté comme un service web indépendant. Cela a deux principaux avantages :

- Chaque service web expose une API bien définie et indépendante du langage, sous la forme d'un document WSDL (WEB Services Description language) contenant à la fois les opérations que le service peut effectuer et les structures de donnée d'entrée/sortie.
- Eucalyptus, utilise avec avantage les fonctionnalités existantes dans les services web telles que les polices de sécurité (WSS) pour la communication sécurisée entre les composants, et se base sur les logiciels de services web se conformant aux standards industriels.



Figure 3.4 : Tableau de bord d'Eucalyptus Cloud

III.3 OpenNebula

OpenNebula à la différence des solutions de Cloud Computing classiques, fournit une boîte à outils complète permettant de gérer de façon centralisée une infrastructure virtuelle hétérogène.

L'outil est compatible avec les hyperviseurs classiques : VMware, Xen, KVM. OpenNebula opère comme un ordonnanceur des couches de stockage, réseau, supervision et de sécurité.

C'est une solution adaptée à la conversion d'une infrastructure virtuelle en Plateforme IaaS. Cette fonction d'orchestration centralisée, d'environnements hybrides est le cœur de l'outil. Ce projet initié en 2005 a livré sa première version en 2008 et reste depuis actif. De nombreuses releases ont permis d'obtenir aujourd'hui des évolutions fonctionnelles importantes sur le support des nœuds de stockage, la haute disponibilité des environnements et l'ergonomie des interfaces d'administration.

OpenNebula est distribuée sous licence [Apache 2.0](#).

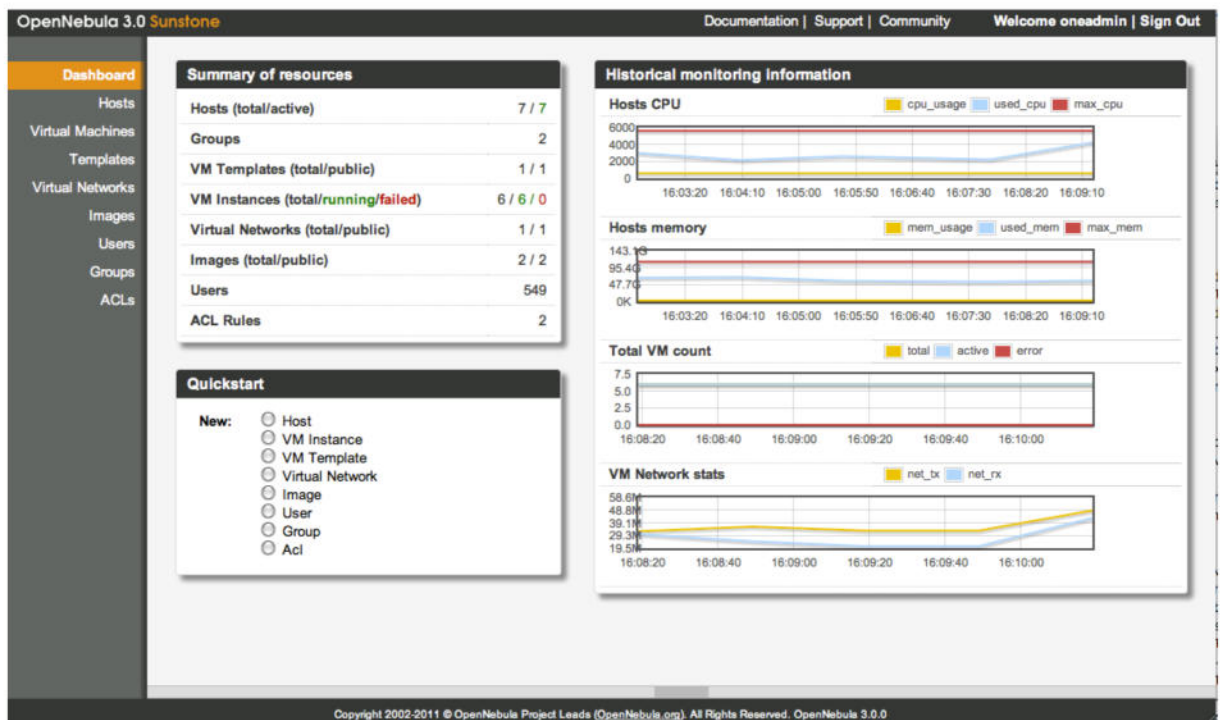


Figure 3.5 : Tableau de bord de OpenNebula

OpenNebula est utilisé par les fournisseurs d'hébergement, les opérateurs de télécommunications, fournisseurs de services IT, centres de supercalculateurs, les laboratoires de recherche et des projets de recherche internationaux. D'autres solutions de Cloud Computing utilisent **OpenNebula** comme le moteur de Cloud ou d'un service du noyau.

III.4 OpenStack

OpenStack est un logiciel open source pour construire des clouds publics et privés. Il connaît maintenant un succès mondial et il est développé et soutenu par des milliers de personnes dans le monde et soutenu par les principaux acteurs dans l'espace du Cloud aujourd'hui. Ce projet est spécialement conçu pour implémenter un cloud privé à partir d'OpenStack.

La fondation OpenStack est une organisation non-commerciale qui a pour but de promouvoir le projet OpenStack ainsi que de protéger et d'aider les développeurs et toute la communauté OpenStack.

Beaucoup d'entreprises ont rejoint la fondation OpenStack. Parmi celles-ci on retrouve : Canonical, Red Hat, SUSE Linux, AT&T, Cisco, Dell, HP, IBM, Yahoo! et Oracle.

C'est un logiciel libre distribué selon les termes de la licence Apache, et il est initié par Rackspace et La Nasa.

Le projet OpenStack

- **OpenStack Compute** : provisionner et contrôler un large réseau de machines.
- **OpenStack Object Storage** : créer une plateforme de stockage de plusieurs Petabytes hautement disponible à l'aide de serveurs standards
- **OpenStack Image Service** : gérer et organiser un large catalogue d'images de machines

Les versions

| Nom | Date | Composants inclus |
|----------|-------------------|--|
| Austin | 21 octobre 2010 | Nova, Swift |
| Bexar | 3 février 2011 | Nova, Glance, Swift |
| Cactus | 15 avril 2011 | Nova, Glance, Swift |
| Diablo | 22 septembre 2011 | Nova, Glance, Swift |
| Essex | 5 avril 2012 | Nova, Glance, Swift, Horizon, Keystone |
| Folsom | 27 septembre 2012 | Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder |
| Grizzly | 4 avril 2013 | Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder |
| Havana | 22 octobre 2013 | Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer |
| Icehouse | 17 avril 2014 | Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove |

Tab 3.1 : Les versions d'OpenStack

Les composants

OpenStack possède une architecture modulaire qui comprend de nombreux composants :

Voici la liste des composants dits intégrés à OpenStack.

- **Compute** : Nova (application)
- **Object Storage** : Swift (stockage d'objet)
- **Image Service** : Glance (service d'image)
- **Dashboard** : Horizon (interface Web de paramétrage et gestion)
- **Identity** : Keystone (gestion de l'identité)
- **Network** : Neutron (auparavant nommé Quantum) (gestion des réseaux à la demande)
- **Storage** : Cinder (service de disques persistants pour les machines virtuelles)
- **Orchestration** : Heat (service d'orchestration à base de templates)
- **Telemetry** : Ceilometer (service de métrologie notamment pour la facturation)
- **Trove**: Service de Base de données à la demande (DBaaS)

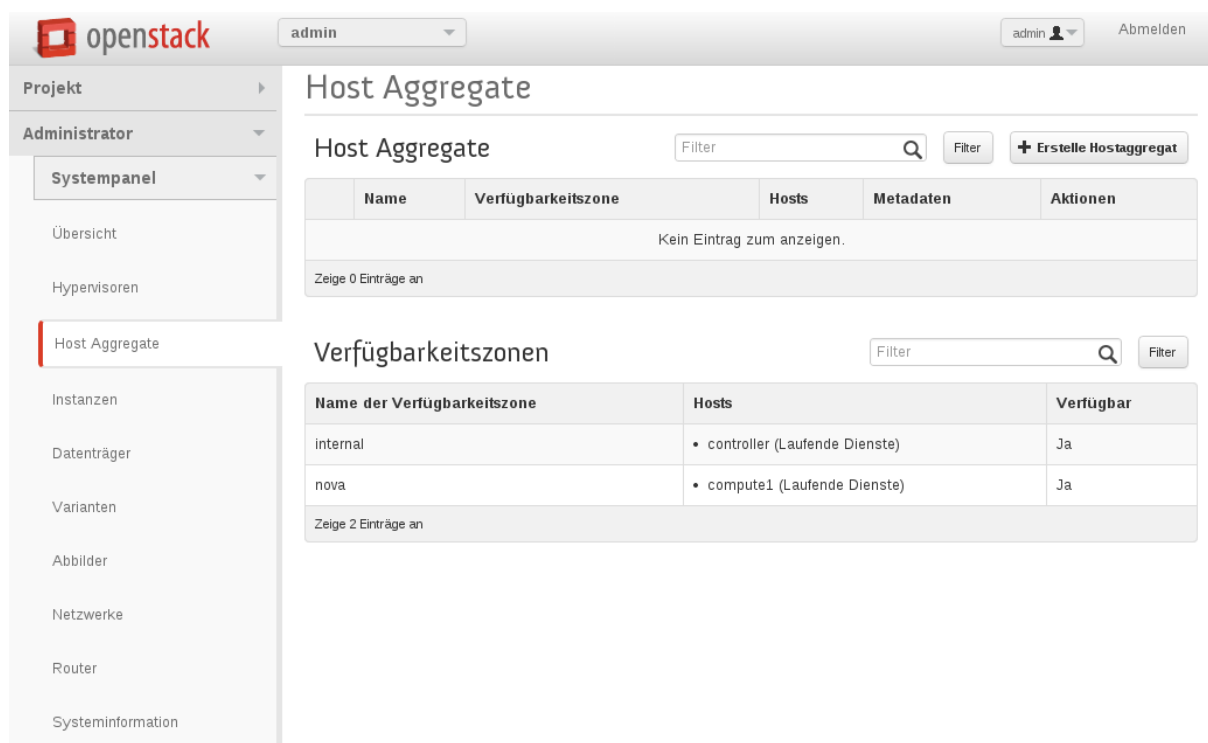


Figure 3.6 : Tableau de bord d'OpenStack

OpenStack offre de nombreux et séduisants avantages : la capacité à monter massivement en taille, d'être interopérable, et d'être agressif sur prix. Que l'on soit une grande entreprise à la recherche d'un Framework pour créer sa plateforme cloud, ou une petite entreprise qui doit affronter les géants du cloud, OpenStack va proposer de nouvelles solutions pré-packagées qui rendront la solution plus accessible, sur le mode Cloud In the Box. Avec son architecture et son design qui offrent la possibilité de démarrer petit et de progressivement aller massivement à l'échelle. Les acteurs du marché doivent trouver des critères de différenciation, jouer sur l'innovation, être agressifs sur les prix, ce qui est une vraie problématique, mais qui peut être en partie résolue en adoptant une solution open source haut de gamme, innovante, et sécurisée. Quant à l'interopérabilité, elle est une des clés pour réussir, avec en plus la capacité d'évoluer vers un cloud hybride donnant le choix de la plateforme de cloud publique.

Comparaison entre les systèmes vus précédemment

Du point de vue de l'architecture, **CloudStack** est une application monolithique, alors que Eucalyptus et OpenStack est divisé en plusieurs morceaux d'applications plus petites. Par exemple, OpenStack est divisé en Compute (Nova), le stockage d'objets (Swift), réseau (Quantum), l'authentification (Keystone), la gestion de l'image (glance), front-end (Horizon).

Eucalyptus essaie d'imiter AWS API autant que possible. Amazon et Eucalyptus ont un accord de partenariat ainsi. Les gens qui vont vers Eucalyptus ont l'avantage de se déplacer à des Clouds hybrides facilement que même api / cli peut travailler à la fois dans le cloud privé Eucalyptus et AWS cloud public. CloudStack a aussi une couche de conversion pour convertir AWS API à CloudStack API.

OpenStack a la plus grande dynamique actuellement en raison du grand nombre de participation des fournisseurs.

Tout Cloud OS Open source peut être utilisé pour construire nuage à la fois public et privé.

| | OPENSTACK | CLOUDSTACK | EUCALYPTUS | OPENNEBULA |
|-------------|--------------------------------|--------------------------------|-----------------------------|--------------------------------|
| Source Code | Fully open-source, Apache v2.0 | Fully open-source, Apache v2.0 | Fully open-source, GPL v3.0 | Fully open-source, Apache v2.0 |

Tab 3.2 : Licences de différents Cloud OS

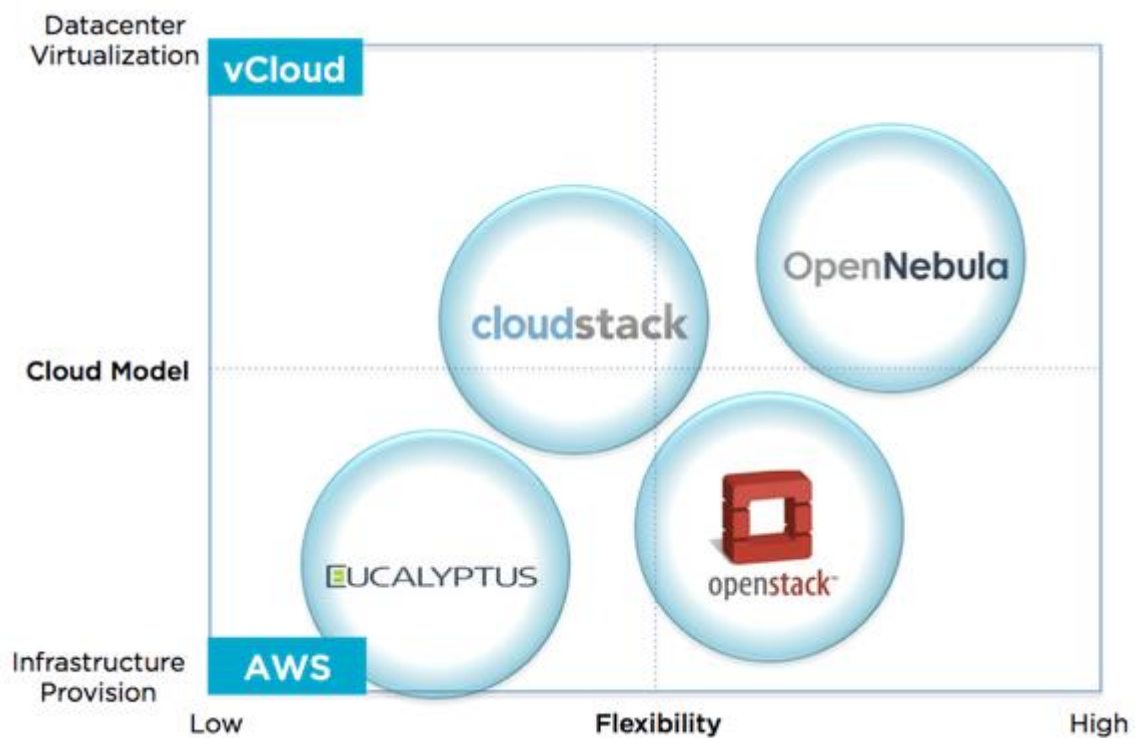


Figure 3.7 : Comparaison selon certains critères

| | CloudStack | Eucalyptus | OpenStack |
|-------------------|-------------|-------------|------------|
| Architecture | Monolithic | 5 part, AWS | Fragments |
| Installation | Medium | Medium | Difficult |
| Administration | UI, EC2 CLI | EC2 CLI | Multi CLI |
| Security | Baseline | Registered | Keystone |
| High Availability | LB multi | 2x failover | Swift only |

Tab 3.3 : Récapitulatif de la comparaison des différents Cloud OS

Chapitre IV : Etude d'un système Open Source de Cloud Computing (OpenStack)

IV.1 Etude générale du système

IV.1.1 - Présentation

OpenStack est un système d'exploitation Open Source destiné aux Clouds Publics et Privés. **OpenStack** est un projet informatique de service d'infrastructure (Infrastructure as a Service (IaaS)) du domaine du cloud computing, mené par la Fondation OpenStack.

La fondation OpenStack est une organisation non-commerciale qui a pour but de promouvoir le projet OpenStack ainsi que de protéger et d'aider les développeurs et toute la communauté OpenStack. La communauté autour de ce projet a rapidement grandi et compte aujourd'hui 2100 développeurs et plus de 150 entreprises. Beaucoup d'entreprises ont rejoint la fondation OpenStack. Parmi celles-ci on retrouve : Canonical, Red Hat, SUSE Linux, AT&T, Cisco, Dell, HP, IBM, Yahoo!. Oracle.

OpenStack est un logiciel libre distribué selon les termes de la licence Apache. Aussi c'est un logiciel d'infrastructure dont on entend beaucoup parler, OpenStack à l'ambition de transformer un Datacenter virtualisé en une offre de cloud opérationnelle. Comme il s'agit d'un projet Open Source, au même titre que Linux, beaucoup voient en lui un système qui pourrait se répandre rapidement et, de fait, imposer un standard dans le cloud, permettre l'interopérabilité.



Figure 4.1 : Informations concernant OpenStack

Historique

2010 : lancement

- En juillet 2010, Rackspace Hosting et la NASA ont lancé conjointement un nouveau projet open source dans le domaine du [cloud computing](#) sous le nom d'OpenStack.
- L'objectif du projet OpenStack est de permettre à toute organisation de créer et d'offrir des services de cloud computing en utilisant du matériel standard.
- La première version livrée par la communauté, dont le surnom est Austin, fut disponible seulement quatre mois après;
- Il est prévu de livrer régulièrement des mises à jour logicielles à quelques mois d'intervalle.

Jusqu'en 2012: de nombreux membres rejoignent le projet

Au début 2012, on compte plus de 150 entreprises ayant rejoint le projet, OpenStack incorpore du code source de [Nebula \(computing Platform\)](#) ainsi que de [Rackspace Cloud](#)

OpenStack, qu'est-ce que c'est pour les informaticiens?



OpenStack est une pile logicielle qui contrôle le fonctionnement d'un cloud. Pour ce faire, elle s'installe au-dessus de la couche de virtualisation des serveurs. En principe, lorsqu'un informaticien souhaite déployer un cloud dans un Datacenter vierge, il commence par installer vSphere de VMware, Windows Server de Microsoft, Enterprise Virtualization de Red Hat ou l'une des distributions Linux Server de Canonical (Ubuntu) ou de Novell (Suse) pour exécuter des machines virtuelles. Et c'est ensuite qu'il a besoin d'un logiciel comme OpenStack pour contrôler et définir quelles machines virtuelles seront disponibles pour quelles fonctions, avec quelles caractéristiques, quelles configurations réseau, quel OS, quel stockage, etc.

Du coup, lorsqu'il sera question d'exécuter dans le cloud un ERP, par exemple, il suffira de définir dans OpenStack que le logiciel a besoin de telles ressources et qu'il est accessible à tels utilisateurs. OpenStack s'occupera alors de tout mettre en place et lancera autant d'instances d'ERP que nécessaire suivant la charge de travail, quitte à relancer tout seul de nouvelles en cas de plantage.

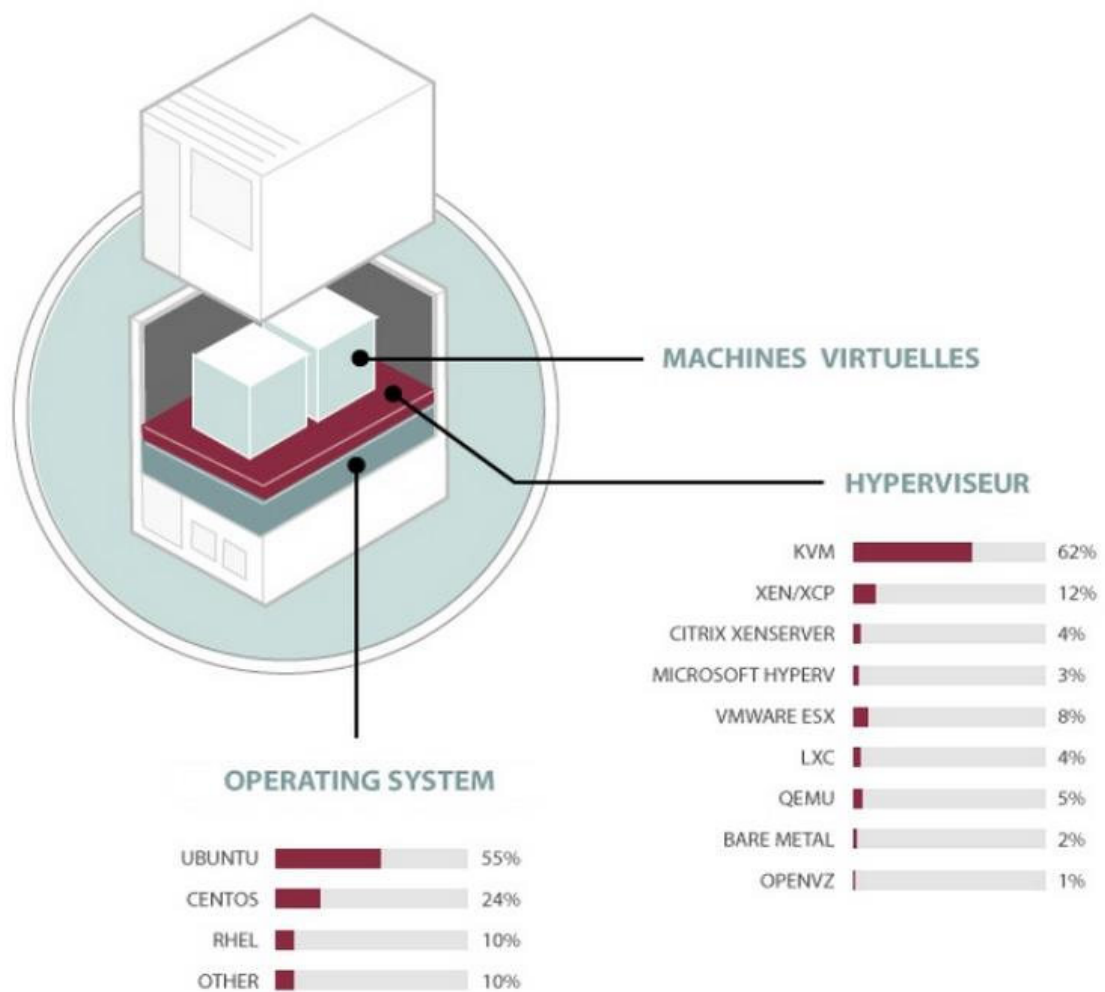


Figure 4.2 : Principe de fonctionnement d'un système Cloud OS

Projet Open Source, au même titre que Linux, OpenStack s'installe sur une machine physique du centre de données par le biais d'une distribution. Il en existe près d'une vingtaine. Red Hat propose Enterprise Linux OpenStack Platform. Canonical et Novell livrent plutôt OpenStack comme un paquet optionnel de leur Linux. Il s'installe alors après la couche de virtualisation maison ou à part, pour compléter tout autre système de virtualisation concurrent. Pour les plus aguerris, il est également possible d'installer le logiciel OpenStack à la main, par-dessus un serveur Linux ou Windows (voir Figure 4.2).

OpenStack revendique le qualificatif de « standard », là où toutes les autres solutions sont propriétaires. Cela signifie trois choses :

- 1. Ne plus être lié à un fournisseur**
- 2. Espérer avoir le système le plus répandu**
- 3. Pour profiter d'un écosystème d'applications**

Cette standardisation intéressera également un éditeur de logiciel SaaS qui veut diffuser son application sur plusieurs clouds, tantôt pour qu'elle soit intégrée à différents portails (et même à des portails d'entreprises, pourquoi pas), tantôt pour toucher des clients européens qui ne veulent pas s'abonner à un cloud public américain comme Amazon EC2, Microsoft Azure ou Google. Il est à noter que les développeurs d'OpenStack l'ont conçu pour qu'il soit compatible avec le format de templates utilisé par Amazon. Il existe donc bien un intérêt à concevoir dès à présent, des templates dans le format d'OpenStack.

IV .1.2 - Architecture

OpenStack est un ensemble de neuf modules écrits en Python. Ils sont chacun dédiés à une fonction précise (gestion des machines virtuelles, gestion du réseau, gestion du stockage, gestion des utilisateurs, supervision de l'ensemble...) et ont chacun une base de données SQL attitrée dans laquelle ils écrivent et lisent les informations qui leur permettent de travailler. Ils communiquent entre eux et avec l'extérieur au moyen d'un jeu de commandes propres (l'API) et selon le protocole MQ.

Toutes les autres technologies sont laissées à l'appréciation de l'exploitant du cloud. Les modules OpenStack peuvent être exécutés par un système Linux ou Windows, les machines virtuelles gérées fonctionneront par-dessus un hyperviseur VMware, Microsoft (Hyper-V), Linux (KVM) ou autre (hyperviseur Xen, partitions LPAR, etc.), la base de données de chaque module peut-être de n'importe quel marque. Les développeurs d'OpenStack utilisent MySQL, mais le système fonctionne aussi bien avec PostgreSQL, Oracle ou SQL Server de Microsoft. Quant au protocole de communication MQ, il peut être lancé par les services RabbitMQ (recommandé), ou Apache Qpid ou encore Zero MQ (voir Figure 4.3).

Selon l'usage que l'on veut faire d'OpenStack, il n'est pas nécessaire d'installer tous les modules. Chaque module dispose d'un nom de code auquel les documentations font référence.

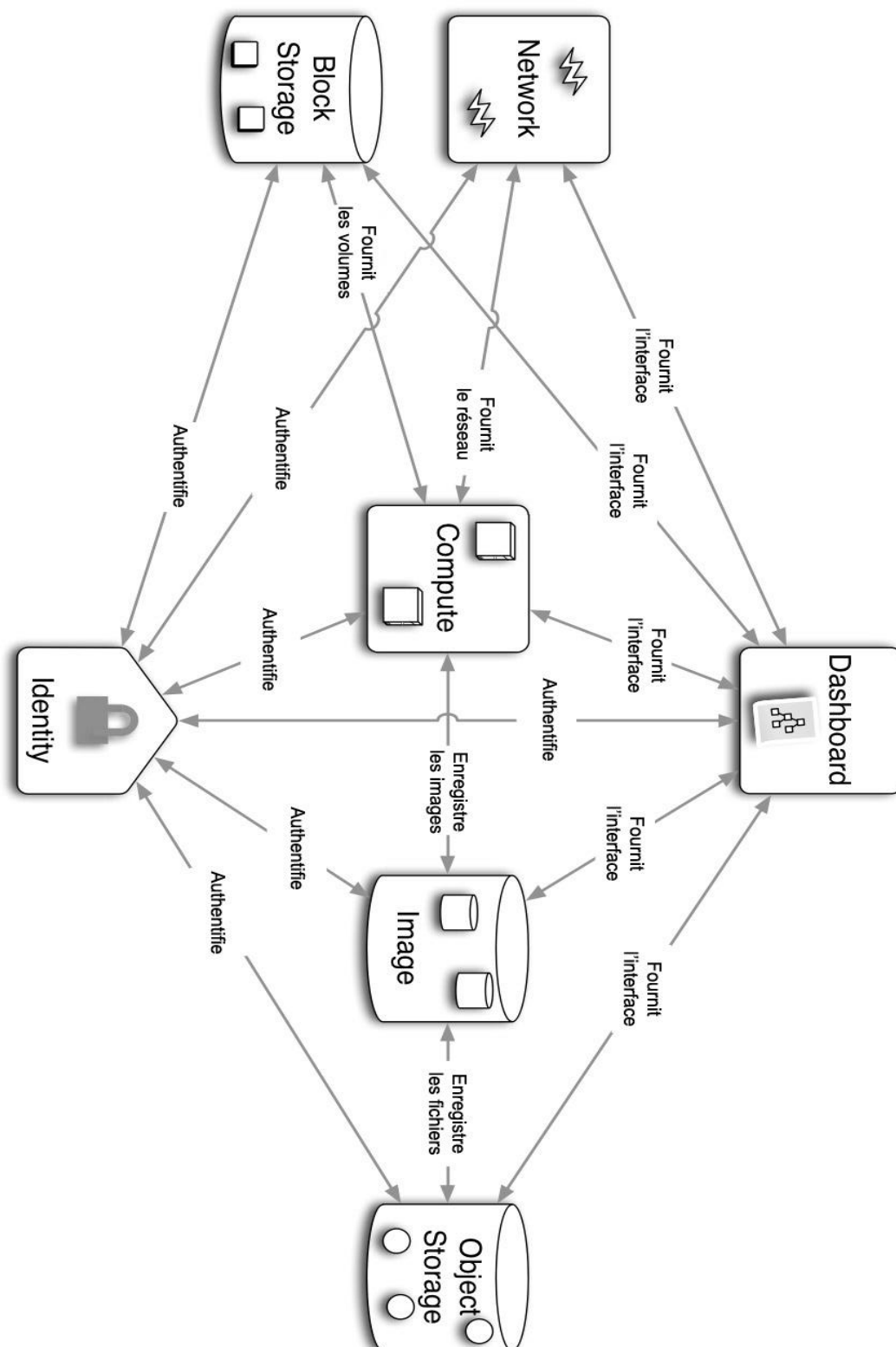


Figure 4.3 : Architecture d'OpenStack

Dashboard : l'interface graphique (Voir Figure 3.6)



Le module Dashboard, dit « Horizon » est l'application web qui permet à l'utilisateur d'interagir avec les services OpenStack et qui sert à les surveiller. On utilise typiquement le Dashboard pour lancer des machines virtuelles, ou un groupe de machines virtuelles, attribuer des adresses IP ou encore contrôler les accès. Le Dashboard peut recevoir des extensions pour ajouter des onglets à son interface, par exemple pour ajouter un système de facturation. On peut remplacer le Dashboard par n'importe quelle interface graphique web déjà écrite pour piloter le cloud EC2 d'Amazon. Fonctionnent Euca2ools (interface en lignes de commande), Hybridfox (plug-in dans Firefox), boto (interface pour envoyer des commandes en Python ou connecter des interfaces écrites en Python), fog (idem mais en Ruby) et PHP-opencloud (idem mais en PHP).

Orchestration : pour organiser le Cloud en templates



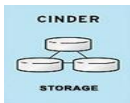
Le module Orchestration, dit « Heat », tient à jour des groupes de machines virtuelles qui doivent fonctionner ensemble, ou templates. Lorsque l'utilisateur veut par exemple lancer plusieurs instances d'un site e-commerce, le module Orchestration indiquera au reste d'OpenStack qu'il faut lancer tel serveur web, tel serveur d'applications et tel serveur de base de données. Il y a la possibilité d'indiquer, via le protocole REST (c'est à dire via le web) des machines virtuelles qui sont hébergées dans un cloud compatible. Typiquement, dans un autre cloud OpenStack ou dans le cloud EC2 d'Amazon.

Compute : pour lancer les machines virtuelles



Le module Compute, dit « Nova », met en route et éteint des machines virtuelles quel que soit leur format (VMware, Hyper-V, KVM, Xen, etc.). Il peut d'ailleurs fonctionner avec plusieurs hyperviseurs en même temps, ce qui permet d'avoir un cloud OpenStack compatible avec différentes technologies de virtualisations incompatibles entre elles. Compute peut également envoyer des messages aux machines virtuelles, typiquement l'identifiant et le mot de passe qui leur permet de démarrer jusqu'au bout.

Object Storage et Block Storage : dévolus au stockage



En matière de stockage, OpenStack dispose de deux modules. Soit il gère des images disque standard (c'est le mode Block, traité par le module « Cinder »), soit il gère les volumes de données au format S3, d'Amazon (c'est mode objet, traité par le module « Swift »). Par ailleurs, il peut y avoir du stockage dans la VM elle-même, mais ce contenu disparaît à l'extinction de la VM.

En mode Block, le module Cinder se base sur un pilote de disque virtuel, du genre iSCSI, pour associer une image disque à une machine virtuelle. Cette image subsiste tant qu'on ne l'efface pas. De plus, Cinder fonctionne un peu comme le RAID ce qui lui permet d'aller lire et écrire sur un autre disque si celui d'origine tombe en panne.

En mode Objet, l'espace de stockage est partagé entre plusieurs machines virtuelles et elles y accèdent avec des commandes REST. A noter que le mode objet revient à un accès en HTTP. De même, ce qui est physiquement stocké peut l'être sur n'importe quel système de

fichier. En pratique, on ne peut pas copier un volume Objet d'un cloud à l'autre. Il faut copier les fichiers. Les volumes de données objets compatibles sont S3, GridFS, Rados Block Device et n'importe quel système de fichiers local.

Image : pour tenir à jour les images disque



Le module Image, dit « Glance » référence l'emplacement de toutes les images disques par le biais d'un chemin d'accès dans Block Storage ou Object Storage. Glance définit également si une image disque est censée accompagner une machine virtuelle en particulier (les données d'un serveur par exemple) ou fait figure de disque générique pour démarrer un serveur en particulier (typiquement avec une copie de Windows ou de Linux dessus). Via REST, il est possible de demander à Glance de copier une image disque d'un cloud OpenStack à l'autre.

Networking : pour établir les plages d'adresses IP



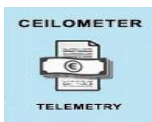
Le module Networking, dit « Neutron » (et initialement « Quantum ») configure le réseau selon différents schémas. Il peut attribuer des adresses IP fixes, définir un serveur DHCP qui en attribuera dynamiquement aux machines virtuelles, faire office d'équilibreur de charge ou encore mettre sur un même réseau privé des machines virtuelles qui sont hébergées sur des clouds physiquement séparés. Il accepte la connexion de services supplémentaires par plugin, par exemple pour doter le réseau d'un pare-feu qui détecte les attaques.

Identity : pour référencer les utilisateurs



Le module Identity, dit « Keystone » tient à jour la liste des utilisateurs en leur faisant correspondre les services auxquels ils ont accès. Ces services peuvent être propres à OpenStack (par exemple pour paramétrer le réseau) ou des templates en particulier (par exemple pour les utilisateurs abonnés à une application SaaS). Keystone peut s'interfacer avec un annuaire d'entreprise, de type LDAP ou Active Directory, et ainsi paramétrer automatiquement les utilisateurs.

Telemetry : pour savoir tout ce qu'il se passe



Le module Telemetry, dit « Ceilometer » récupère tous les messages issus de tous les fichiers log du cloud et les enregistre comme des métriques dans sa base de données. Ces métriques doivent, à terme, servir aux services de facturation, en particulier pour évaluer le prix qu'un utilisateur doit payer au regard de sa consommation du cloud. Dernier module à avoir fait son apparition, Ceilometer ne dispose pas encore de beaucoup de plug-ins qui en tiennent compte.

Trove : la base de données comme service (DBaaS) (Nouvel module sortie le 17 avril 2014 mais en cours de développement)

Depuis la sortie de OpenStack Icehouse on a inclut le module Trove qui est une base de données comme service pour OpenStack. Elle est désignée pour s'exécuter sur [OpenStack](#), dans le but de permettre aux utilisateurs d'utiliser rapidement et facilement les caractéristiques d'une base de données relationnelle sans la charge de traitement des tâches administratives complexes. Les utilisateurs de Cloud et les administrateurs de base de

données peuvent fournir et gérer des instances de bases de données multiples si nécessaire. Initialement, le service mettra l'accent sur la fourniture de l'isolement des ressources à haute performance tout en automatisant les tâches administratives complexes, y compris le déploiement, la configuration, la correction, les sauvegardes, les restaurations, et le suivi.

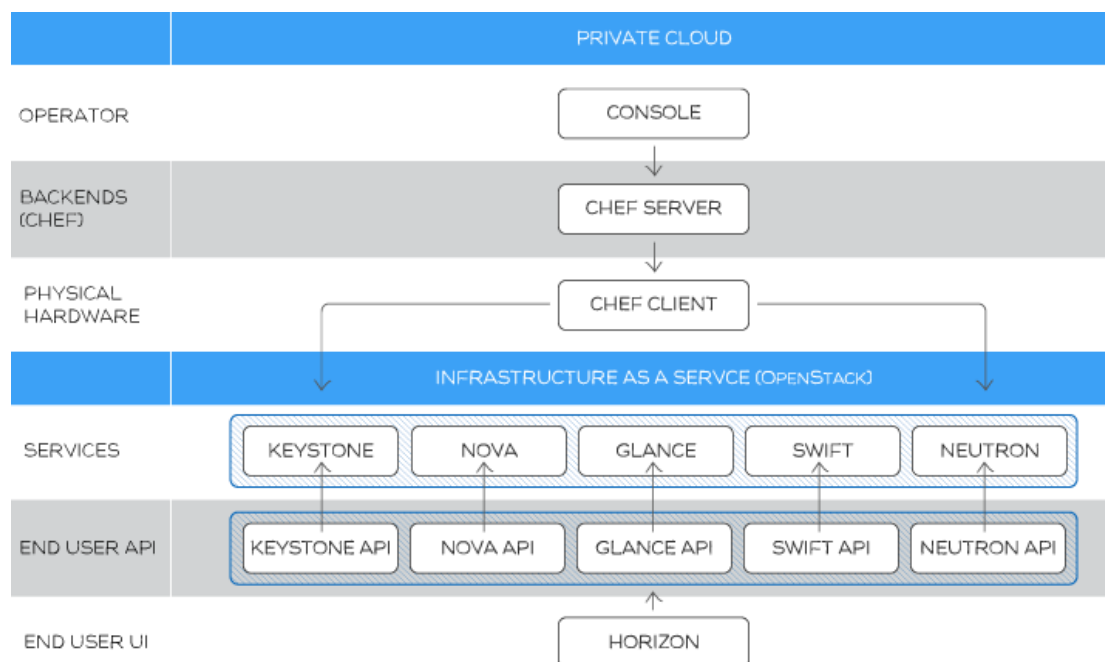


Figure 4.2 : Les composants d'Openstack

IV.1.3 - Installation

Le moyen le plus simple et le meilleur pour commencer votre voyage dans le développement OpenStack serait par Devstack. Donc, on va suivre à travers la configuration de base d'OpenStack des outils qui sont utiles pendant l'utilisation de Devstack.

Il est recommandé d'utiliser la machine virtuelle à installer Devstack que vous pouvez facilement vous débarrasser de celui-ci sans avoir beaucoup à perdre, en cas de collisions. En outre, il ferait gagner du temps si vous gardez un aperçu de la configuration Devstack sur la machine virtuelle de sorte que dans l'avenir, vous pourriez économiser le temps d'installation. Commençons par l'installation

On utilise 64 bits Ubuntu 12.04 pour la configuration. (Plus d'infos sur le OS voir [ANNEXES](#))

Installez Virtual Box

Créer une nouvelle machine virtuelle avec un minimum de 2 Go de RAM (nécessaire pour Devstack pour bien fonctionner) et installer Ubuntu 12.04 64 bits.

Exécuter la machine virtuelle

Ouvrez la console

Exécuter une mise à jour

```
$ Sudo apt-get update
```

Installez git

```
$ Sudo apt-get install git
```

Devstack est un script Shell utilisé pour déployer un environnement de développement complet de OpenStack. Le script Devstack installe toutes les dépendances nécessaires à l'exécution OpenStack il n'y a donc pas besoin d'installer quoi que ce soit séparément. C'est un bon endroit pour commencer pour les débutants à partir avec le développement d'OpenStack.

Ouvrez la console et cloner le repo de Devstack qui contient un script qui va installer OpenStack. Vous devez cloner ce repo où vous voulez que votre Devstack de résider. Cette commande va chercher le repo de Devstack sur votre machine locale.

```
$ git clone https://github.com/openstack-dev/devstack.git
```

Maintenant, vous trouverez un répertoire de Devstack où vous avez cloné le repo

```
$ cd devstack
```

Exécutez le script d'installation OpenStack

```
$ ./stack.sh
```

Le script va prendre un certain temps à construire. Il vous sera demandé d'entrer les mots de passe pour les services par défaut activée dans le script.

Tapez OpenStack comme mot de passe si vous voulez

Après il est fait, il va retourner une URL pour horizon. Ouvrez ce lien dans votre navigateur et vous pouvez accéder au tableau de bord OpenStack!

Par défaut, vous obtenez deux pseudos admin et démo. Le mot de passe sera le mot de passe que vous définissez

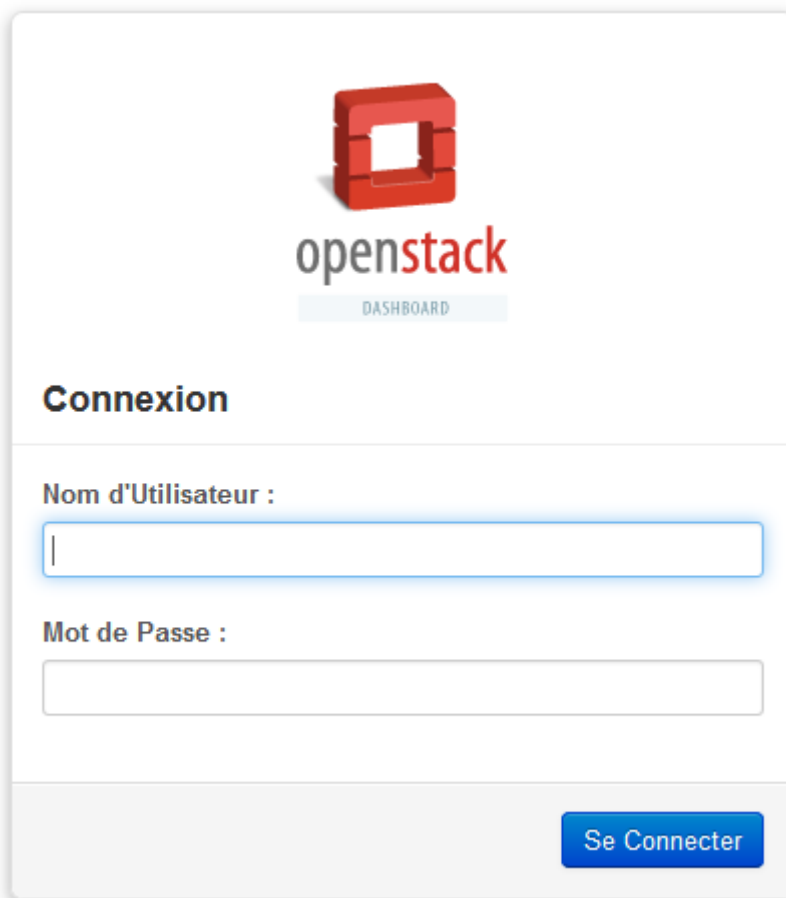
Ici, vous êtes, terminé avec la configuration de base pour OpenStack!

(Rappel - Ne pas oublier de prendre un snap-shoot de la machine virtuelle une fois que vous avez atteint ce point!)

IV.2 Test du bon fonctionnement

IV.2.1 Mode graphique

Afin d'avoir exécuté le script `./stack.sh` vous pouvez maintenant accéder au Dashboard d'openstack suivant l'adresse ip de votre machine, puis vous obtiendrez l'interface suivante :



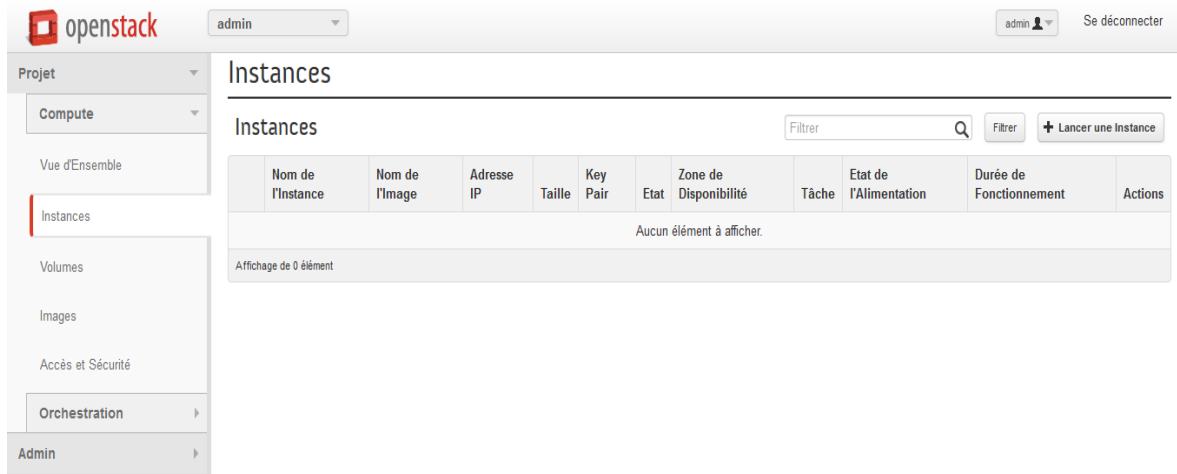
The image shows the OpenStack Dashboard login page. At the top, there is the OpenStack logo, which consists of a red 3D cube with a white square in the center, and the text "openstack" in a sans-serif font, with "open" in black and "stack" in red. Below the logo is a light blue button labeled "DASHBOARD". Below this, the word "Connexion" is displayed in bold. There are two input fields: "Nom d'Utilisateur :" followed by a text input field, and "Mot de Passe :" followed by a password input field. At the bottom right, there is a blue button labeled "Se Connecter".

Figure 4.3 : Connexion au tableau de bord

Votre login est par défaut admin (comme administrateur) ou demo (administrateur d'un projet)

Le mot de passe est le mot de passe que vous avez entré

Après l'authentification vous aurez l'interface comme cela :



Vous pouvez maintenant créer une instance

Rq : par défaut deux images sont téléchargées (cirros && Fedora) donc créons une instance pour cirros

Lancer une Instance

Détails *

Accès et Sécurité *

Post-Création

Advanced Options

Zone de Disponibilité :

nova

Nom de l'Instance : *

test_cirros

Type d'instance : *

m1.micro

Nombre d'Instances : *

1

Source de l'instance de démarrage. *

Boot from image

Nom de l'Image :

cirros-0.3.2-x86_64-uec (24,0 Mo)

Spécifier les détails de démarrage d'une instance.

Le tableau ci-dessous montre les ressources liées aux quotas et utilisées par ce projet.

Détails du Type d'Instance

| | |
|-----------------|----------|
| Nom | m1.micro |
| VCPUs | 1 |
| Disque racine | 0 Go |
| Disque éphémère | 0 Go |
| Disque Total | 0 Go |
| RAM | 128 Mo |

Limites du Projet

Nombre d'Instances

0 sur 10 utilisés

Nombre de VCPUs

0 sur 20 utilisés

RAM Totale

0 sur 51 200 Mo utilisés

Annuler

Lancer

Après avoir cliqué sur Lancer vous aurez ceci :

Instances

| Instances | | | | | | | | | | | |
|------------------------|-------------------|-------------------------|------------|--|-----------------------|--------|-------------------------------|-------|------------------------|-------------------------|----------------------------|
| Filtrer | | | Filtrer | | + Lancer une Instance | | Redémarrage à Chaud Instances | | Terminer Instances | | |
| | Nom de l'Instance | Nom de l'Image | Adresse IP | Taille | Key Pair | Etat | Zone de Disponibilité | Tâche | Etat de l'Alimentation | Durée de Fonctionnement | Actions |
| | test_cirros | cirros-0.3.2-x86_64-uec | 10.0.0.2 | m1.micro 128Mo RAM 1 VCPU 0Bytes Disques | - | Active | nova | None | Running | 0 minute | Créer un Instantané Plus ▾ |
| Affichage de 1 élément | | | | | | | | | | | |

Vous pouvez maintenant tester le bon fonctionnement de votre instance en cliquant sur test_cirros :

52

Console de l'Instance

Si la console ne répond plus aux entrées clavier, cliquez sur la barre d'état grise ci-dessous. [Cliquez ici pour ne voir que la console](#)
To exit the fullscreen mode, click the browser's back button.

```
Connected (unencrypted) to: QEMU (instance-00000003) Send CtrlAltDel
[ 5.223233] Registering the dns_resolver key type
[ 5.241448] scsi 1:0:1:0: CD-ROM QEMU QEMU DVD-ROM 2.0. PQ
: 0 ANSI: 5
[ 5.276195] sr0: scsi3-mmc drive: 4x/4x cd/rw xa/form2 tray
[ 5.277403] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 5.387946] sr 1:0:1:0: Attached scsi generic sg0 type 5
[ 5.469460] registered taskstats version 1
[ 6.466163] Freeing initrd memory: 3640k freed
[ 6.878320] Magic number: 2:555:866
[ 6.881525] rtc_cmos 00:01: setting system clock to 2014-06-12 03:53:53 UTC (
1402545233)
[ 6.883098] powernow-k8: Processor cpuid 663 not supported
[ 6.887376] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[ 6.888385] EDD information not available.
[ 6.947074] Freeing unused kernel memory: 924k freed
[ 7.050569] Write protecting the kernel read-only data: 12288k
[ 7.183759] Freeing unused kernel memory: 1608k freed
[ 7.276251] Freeing unused kernel memory: 1188k freed

further output written to /dev/ttyS0
[ 16.913757] hrtimer: interrupt took 149685295 ns

login as 'cirros' user. default password: 'cubswin:)'. use 'sudo' for root.
test-cirros login:
```

Puis entrez le login + mot de passe défini par défaut :

```
Connected (unencrypted) to: QEMU (instance-00000003)
: 0 ANSI: 5
[ 5.276195] sr0: scsi3-mmc drive: 4x/4x cd/rw xa/form2 tray
[ 5.277403] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 5.387946] sr 1:0:1:0: Attached scsi generic sg0 type 5
[ 5.469460] registered taskstats version 1
[ 6.466163] Freeing initrd memory: 3640k freed
[ 6.878320] Magic number: 2:555:866
[ 6.881525] rtc_cmos 00:01: setting system clock to 2014-06-12 03:53:53 UTC (
1402545233)
[ 6.883098] powernow-k8: Processor cpuid 663 not supported
[ 6.887376] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[ 6.888385] EDD information not available.
[ 6.947074] Freeing unused kernel memory: 924k freed
[ 7.050569] Write protecting the kernel read-only data: 12288k
[ 7.183759] Freeing unused kernel memory: 1608k freed
[ 7.276251] Freeing unused kernel memory: 1188k freed

further output written to /dev/ttyS0
[ 16.913757] hrtimer: interrupt took 149685295 ns

login as 'cirros' user. default password: 'cubswin:)'. use 'sudo' for root.
test-cirros login: cirros
Password:
$ _
```

Et voilà vous êtes connecté à votre instance :

```
login as 'cirros' user. default password: 'cubswin:)'. use 'sudo' for root.
test-cirros login: cirros
Password:
$ uname -a
Linux test-cirros 3.2.0-60-virtual #91-Ubuntu SMP Wed Feb 19 04:13:28 UTC 2014 x
86_64 GNU/Linux
$
```

Vous pouvez manipuler votre instance comme vous voulez.

IV.2.1 Mode commandes

On peut aussi procéder d'une autre manière de travailler à travers les lignes de commandes. Avant tout il faut savoir que les utilisateurs ne peuvent pas être créés sans qu'ils appartiennent à un projet donc le projet doit être créé en premier

```
$ keystone tenant-create --name test --description "description projet" --enabled true
```

Puis on affiche la liste des projets : **\$ keystone tenant-list**

Les rôles sont les permissions qu'on accorde aux utilisateurs dans un projet (on possède de 2 rôles admin et Member) on les crée

```
$ keystone role-create --name admin && keystone role-create --name Member
```

On affiche la liste des rôles : **\$ keystone role-list**

Puis on crée un utilisateur suivant un projet et avec un rôle bien définie :

On a besoin de l'id du projet :

```
$ TENANT_ID=$(keystone tenant-list \ | awk '/project_name/{print $2}')
```

Puis on crée l'utilisateur

```
$ keystone user-create --name admin --tenant_id $TENANT_ID --pass $Password --email root@localhost --enabled true
```

On ajoute le rôle

```
$ keystone user-role-add --user $USER_ID --role $ROLE_ID --tenant_id $TENANT_ID
```

Avec **\$ USER_ID= (keystone user-list awk '/admin/{print \$2}')**

ET **\$ ROLE_ID= (keystone role-list awk '/admin/{print \$2}')**

On peut afficher la liste de tous ce qu'on a fait avec:

```
$keystone tenant-list
```

```
$ keystone role-list
```

```
$ keystone user-list
```

La possibilité d'upload d'une image cloud est aussi intéressante avec :

```
$ wget {le lien de l'image}  
$ glance image-create --name='nom de l'image' --disk-format= 'le format du disk' --  
container-format=bare
```

Puis on affiche les images qu'on a `$glance image-list`

On peut supprimer l'image `$ glance image-delete $IMAGE_ID`

Avec `$IMAGE_ID` est l'id de l'image facile à la trouver comme précédemment

On peut afficher les détails de l'image

```
$ glance image-show $IMAGE_ID
```

Pour savoir si les machines sont en cours d'exécution, on tape :

```
$ sudo nova-manage service list
```

Pour exécuter une instance Cloud, on affiche d'abord les images qu'on a :

```
$ nova image-list
```

Puis on démarre notre instance :

```
$ nova boot MyInstance --image $IMAGE_ID --flavor 2 --key_name demo
```

Afficher la list des computes :

```
$ nova list
```

Pour se connecter à la machine (ex Ubuntu) avec une clé `demo.pem`

```
$ ssh -i demo.pem ubuntu@Adresse_Ip_Hote
```

Pour supprimer une instance : `$ nova delete MyInstance`

Dans ce chapitre, nous avons fait une simulation de mise en place notre plateforme pour construire un cloud privé, à travers cette installation nous avons pu dégager l'intérêt de cloud, et ce que peut nous apporter de bénéfices et augmentation de performances, mais la mise en place de cette technologie nécessite un matériel puissant qui supporte la virtualisation et les différentes de technologies utilisées dans le Cloud.

Conclusion

Ce rapport est le fruit de deux mois et demi de travail et de recherches, une période durant laquelle on a mené une étude de recherche sur les systèmes du Cloud Computing.

Nous avons donné une idée générale sur le Cloud Computing, son architecture et ses différents services. Nous avons fait par la suite une étude sur différentes solutions Open Source du Cloud. Ceci nous a permis d'avoir une idée riche sur les techniques de virtualisation ainsi que les différentes solutions disponibles du Cloud Computing et surtout de bien maîtriser le concept du Cloud.

Lors de l'élaboration de ce projet nous avons rencontré des problèmes concernant:

- La non disponibilité des ressources matérielles nécessaire pour mettre en place le Cloud privé.
- La rareté des informations concernant le sujet parce que cette technologie est assez récente.
- Le débit de connexion qui n'est pas performant à la faculté des sciences ...

La difficulté du travail et les contraintes confrontées dont on a pu trouver, avec la contribution de notre encadrant, des solutions alternatives afin de les résoudre.

Annexes

Qu'est-ce que les scripts font?

Maintenant, si vous jetez un œil à votre répertoire devstack vous trouverez des scripts très utiles. Ici, je vais vous dire comment ces scripts sont utilisés.

stack.sh

Nous avons d'abord exécuté ce script pour installer Openstack. Ce script vous permet de spécifier les options de configuration de ce référentiels git à utiliser, les services activés, la configuration du réseau et les différents mots de passe. Vous devez exécuter ce script à chaque fois que vous faites un changement à la configuration de OpenStack ou d'activer le nouveau service.

unstack.sh

Arrête tous les services démarrés par stack.sh (surtout) MySQL et Rabbit sont laissés en marche comme OpenStack rafraîchit de code ne les obligent pas à être renouvelées. Notez que les modifications apportées au code devraient demeurer intacts après l'exécution unstack.sh sauf mention contraire.

rejoindre-stack.sh

Ce script rejoint un écran existant, ou recrée une session d'écran d'une exécution précédente de stack.sh. Cela signifie que si vous avez précédemment exécuté stack.sh et vous souhaitez rejoindre la session que vous exécutez rejoindre-stack.sh. Faire cela restaure toutes les données des sessions précédentes. Donc n'ayez pas peur de perdre vos données après le redémarrage de votre machine.

Il suffit de lancer `./ unstack.sh`

Ensuite, redémarrez votre machine

Lorsque vous voulez vous joindre à la session de nouveau fonctionner. / `rejoin-stack.sh`

Et vous avez toutes vos données à la place! Vous pouvez vérifier en accédant au tableau de bord sur la même URL que précédemment.

run_test.sh

Cet outil est exécuté sur l'ensemble du projet en utilisant un script run_test.sh. Cela vous donnera des informations sur les espaces blancs errants.

clean.sh

Il est utilisé si vous voulez effacer tous les fichiers liés OpenStack de votre système.

exercise.sh

Ce script exécute tous les exemples présents dans le répertoire et les rapports sur les résultats devstack / exercices. Le répertoire des exercices contient les scripts de test utilisées pour valider et démontrer certaines fonctions d'OpenStack. Ces scripts savent comment sortir tôt ou sauter des services qui ne sont pas activés.

Concernant le système d'exploitation qu'on s'est basé est Ubuntu 12.04 mais vous pouvez aussi utilisez Ubuntu openstack qui est la dernière version d'Ubuntu 14.04 dédié pour installer Ubuntu en parallèle des composants nécessaires au déploiement d'openstack à l'aide de l'outil JuJu, qui vous donne aussi la possibilité d'effectuer le Maas (Metal as a Service)

On peut faire un script simple pour installer openstack qui est le suivant :

```
#!/ Bash/bin  
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get dist-upgrade  
sudo apt-get install git  
git clone https://github.com/openstack-dev/devstack.git  
cd devstack  
./stack.sh
```

Et voilà Openstack qui s'installe d'une durée de 1h30min vers 2h00min

Bibliographie

Zaigham Mahmood, Richard Hill-Cloud Computing for Enterprise Architectures –Spr-2011

Kan Yang, Xiaohua Jia (auth.)-Security for Cloud Storage Systems-Springer New York
2014

Siani Pearson (auth.), Siani Pearson, George Yee (eds.)-Privacy and Security 2013

Nick Antonopoulos, Lee Gillam-Cloud Computing_ Principles, Systems and Applications
2010

Handbook of Cloud computing - Borko Furht·Armando Escalante - 2010

Graham Speake, Patrick Foxhoven-Securing the Cloud_ Cloud Computer Security
Techniques and Tactics-Elsevier Science Technology - 2011

Implementing and Developing Cloud Computing Applications - 2011

Kevin_Jackson,_Cody_Bunch_OpenStack_Cloud_Computing_Cookbook_Second_Edition_
2013

Webographie

<http://cloudcomputing.fr/>

<http://pro.clubic.com/it-business/cloud-computing/article-689272-4-openstacktechnologie-standardiser-cloud.html>

http://fr.wikipedia.org/wiki/S%C3%A9curit%C3%A9_du_cloud

http://nauges.typepad.com/my_weblog/2013/06/r%C3%A9concilier-cloud-computing-et-s%C3%A9curit%C3%A9-.html

https://www.gemalto.com/france/identite/inspiration/cloud_computing.html

<https://www.cloudwatt.com/fr/le-cloud/qui-sommes-nous/open-source.html>

<http://www.discoposse.com/index.php/2014/01/26/openstack-havana-all-in-one-lab-on-vmware-workstation/>

<http://blogs.vmware.com/workstation/2013/04/create-an-openstack-lab-in-vmware-workstation.html>

<http://uksysadmin.wordpress.com/2011/02/17/running-openstack-under-virtualbox-a-complete-guide/>

<https://wiki.openstack.org/wiki/DevStackVirtualbox>

<http://www.ubuntu.com/download/cloud/install-ubuntu-cloud>

<http://serverfault.com/questions/409216/what-is-the-correct-network-configuration-for-a-devstack-vm-virtualbox/409331#409331>

<http://www.infoq.com/fr/news/2014/01/openstack-icehouse>

<http://decrypt.ysance.com/2013/07/introduction-a-openstack-2-sur-2-installation-devstack-et-utilisation-via-console-web-horizon/>

<https://wiki.openstack.org/wiki/ReleaseNotes/Icehouse>

<https://gist.github.com/11044642.git>

<http://debian.netside.net/xps13-openstack.html>

<http://vimeo.com/87528023>

<http://cloudsecurity.org/>

<http://fr.slideshare.net/pygnosis/open-stack-en10min>

<https://wiki.openstack.org/wiki/DevStackVirtualbox>

<https://gist.github.com/max-lobur/7786074>

<https://github.com/StackGeek/openstackgeek/tree/master/grizzly>