Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: *Géni Logiciel (G.L)*

Thème

Déploiement d'un Cloud privé sous openstack et intégration d'un système d'identification centralisé (LDAP)

Réalisé par :

- **Bali Hicham**
- **Malti Mohammed Nourreddine**

Présenté le ... Juillet 2019 devant le jury composé de :

- ***Mme. Abdeldjalil Hanane*** *(Président)*
- ***Mme. Labraoui Nabila*** *(Encadreur)*
- ***Mme. Malti Djawida*** *(Examinateur)*

Année universitaire : 2018-2019

Résumé

Le changement rapide de la technologie et l'augmentation immense des données, obligent les moyennes et grandes entreprises à se mettre à jour avec la nouveauté informatique car un petit retard causera une grande perte de temps et par la suite des pertes financières difficiles à gérer. Une de ses nouveautés est le domaine du Cloud Computing qui vient de franchir les hauts niveaux de la gestion, automatisation et rapidité à résoudre différentes tâches, et qui représente une tendance consommée par les entreprises.

Notre travail consiste à faire une migration d'un simple système d'entreprise à un Cloud privé qui contiendra une bonne partie de l'ancien système mais avec un nouveau mécanisme. L'importante tâche dans ce travail est de centralisé l'identification tout en gardant tous les utilisateurs existants et avec un seul accès à toutes les applications et systèmes intégrés dans l'entreprise.

Abstract

The fast change of technologies and the immense increase of data, oblige the medium and big company to be upgraded with the novelty of computer science. Small delay may cause a great loss of time and then, a financial losses that are difficult to manage. One of those innovations is cloud computing that has just crossed the high level of management, automation and speed to solve different tasks, and which represents a trend consumed by the companies.

Our job is to migrate a simple enterprise system to a private cloud that will contain the main part of the old system but with a new mechanism. The important task in this work is to centralize the identification while keeping all existing users and with only one access to all applications and systems integrated into the company.

ملخص

التغيير السريع للتكنولوجيات والزيادة الهائلة في البيانات ، ألزم الشركات المتوسطة والكبيرة بالترقية مع حداثة التكنولوجيا ، لأن أي تأخير بسيط قد يؤدي إلى ضياع كبير للوقت ثم إلى خسائر مالية في وقت لاحق ومن الصعب تقبل ذلك ، أحد هذه الابتكارات هو الحوسبة السحابية التي تجاوزت مستوى عالٍ من الإدارة والسرعة و الحل الأوتوماتيكي للمشاكل المختلفة ، والتي تمثل اتجاهًا تستهلكه الشركات حاليا.

تتمثل مهمتنا في الانتقال من نظام مؤسسة بسيط إلى سحابة خاصة تحتوي على جزء جيد من النظام القديم ولكن مع وجود آلية جديدة ، تتمثل المهمة في هذا العمل في جعل عملية تحديد الهوية مركزية مع الاحتفاظ بجميع المستخدمين الحاليين مع تطبيق واحد فقط الوصول إلى جميع التطبيقات ونظام متكامل في الشركة.

Remerciements

Au nom d'*Allah le Miséricordieux le Très Miséricordieux. Certes, la louange est à Allah, de qui nous implorons aide et repentance.*

Nous remercions Allah le tout puissant de nous avoir guider et aider à la réussite de ce projet avec

Courage et patience.

En préambule à ce mémoire, nous souhaitons adresser nos remerciements les plus Sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de

ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

On tient à remercier sincèrement Notre encadreur Mme. Labraoui Nabila,

Pour sa contribution, et pour son temps précieux qui nous as accordés.

On tient aussi à remercier sincèrement Mr. Bouida Djelloul et Mme. Labraoui Nabila

Pour leurs contributions et soutiens tout au long du projet

qui s'ont toujours montré à l'écoute et l'aide tout au long de la période de ce projet.

On tient à saluer les membres du jury Mme. Abdeldjalil Hanane et Mme. Malti Djawida qui nous ont honorés par leur présence et d'avoir

accepté d'évaluer notre travail.

On n'oublie surtout pas nos parents pour leurs contributions, leurs soutiens et leur patience.

Merci à toutes et à tous.

Hicham Bali

Dédicaces

C'est avec profonde gratitude et sincères mots, que je remercie le bon Dieu, le tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement

A mes chers parents, qui m'ont aidé, de près et de loin.

A mes chers frères et sœur, qui m'ont donné le courage

A tous mes amis .

A toute la familles Bali et Malti.

A mon binôme Malti Mohammed

J'espère qu'un jour, je pourrai leur rendre un peu de ce qu'ils ont fait pour moi, que dieu leurs prête bonheur et longue vie

Malti Mohammed Nourreddine

Dédicaces

Tout d'abord je tiens à remercier Dieu le tout puissant de m'avoir donné la santé, la volonté, le courage et de m'avoir fourni sa bénédiction ;

Je dédie ce modeste travail ;

A mes chers parents.

A mes deux frère et ma sœur ,

A tous mes amis .

A toute la famille Malti et Bali.

A mon binôme Bali Hicham.

Sommaire

Table Des Figures

Introduction Générale

Au cours des dernières années, l'informatique en nuage ou Le Cloud Computing est devenue l'une des tendances les plus chaudes dans le domaine des TIC, qui suscite beaucoup d'attention de la part des chercheurs des domaines industriels et académiques. Ce dernier est censé faire évoluer la manière dont différentes technologies collaborent pour modifier l'approche de l'organisation en matière de construction et de gestion de son infrastructure informatique ainsi que ses services informatiques, le Cloud Computing peut augmenter la productivité des entreprises et leur permettre de se concentrer sur l'augmentation des profits et la réduction des coûts.

Comme d'autres avancées scientifiques et technologiques, en particulier celles qui évoluent à partir de technologies existantes, le Cloud Computing ne réside pas dans la technologie elle-même, mais dans les changements opérationnels intervenus lors de son déploiement et de son application. Le Cloud Computing consiste en une interconnexion et une coopération de ressources informatiques, situées dans diverses structures internes, externes ou mixtes et dont le mode d'accès est basé sur les protocoles et standards Internet. Le Cloud Computing est devenu le sujet le plus débattu aujourd'hui dans le secteur des technologies de l'information. Le consensus qui se dégage est que le Cloud Computing jouera un rôle de plus en plus important dans les opérations informatiques des entreprises au cours des années à venir. C'est pour cela que ce travail de mastère s'intéresse au passage au Cloud Computing.

Ce présent mémoire est constitué de trois chapitres :

Le chapitre 1 constitue un aperçu sur le Cloud Computing en abordant les principes fondamentaux du Cloud Computing, les considérations d'architecture, la sélection de services de technologie Cloud, les avantages et inconvénients du Cloud Computing, et enfin le problème de sécurité du Cloud.

Le chapitre 2 introduit Openstack avec tous ses composants et fonctionnalités.

Le chapitre 3 traite le concept et la solution technologique pour gérer l'authentification dans Openstack et comment déployer une infrastructure efficace et manipuler le service de l'identité pour lui permettre de communiquer avec un serveur d'authentification centralisé.

CHAPITRE I :

Cloud Computing

Introduction

Une adaptation rapide est la clé du succès pour la survie de toute entreprise dans le contexte économique dynamique actuel. Si l'on exploite une entreprise rentable aujourd'hui, cela ne signifie pas que le modèle commercial actuel fournira la même croissance à l'avenir. En plus de s'adapter à l'évolution de la réglementation gouvernementale, les entreprises doivent explorer et mettre en œuvre de nouveaux domaines pour faire face aux tendances informatiques actuelles.

Nous allons voir un aperçu du Cloud Computing, qui fournit une infrastructure informatique efficace aux entreprises d'aujourd'hui, quel que soit le lieu ou le moment.

De nombreuses entreprises ont déjà basculé leurs ressources informatiques vers le Cloud car, selon elles, ce modèle constitue un moyen plus rentable et plus efficace pour servir leurs clients, leurs partenaires et leurs fournisseurs. En revanche, d'autres entreprises examinent ce modèle avec plus de prudence en ce qui concerne la sécurité de leurs processus métier et de leurs actifs intellectuels. Le principal avantage du Cloud Computing réside dans le fait qu'il élimine bon nombre des contraintes complexes rencontrées dans l'environnement informatique traditionnel, notamment les coûts, l'espace, le temps et l'énergie.

Nous allons voir dans ce chapitre l'évolution des Datacenters, jusqu'à l'arrivé du Cloud Computing, en abordant les principes fondamentaux du Cloud Computing, les considérations d'architecture, la sélection de services de technologie Cloud et les avantages et inconvénients du Cloud Computing.

1.1 Définitions

1.1.1 La virtualisation

C'est un cadre ou une méthodologie permettant de diviser les ressources d'un ordinateur en plusieurs environnements d'exécution en appliquant une ou plusieurs technologies telles que le partitionnement matériel et logiciel, le partage de temps, la simulation partielle ou complète de la machine [2].

Définition de NIST:"*La virtualisation est la simulation du logiciel et / ou du matériel sur lequel d'autres logiciels sont exécutés*"

1.1.2 Data center

Un data center ou centre de données est un site physique regroupant des installations informatiques (serveurs, routeurs, commutateurs, disques durs, …) chargées de stocker et de distribuer des données à travers un réseau interne ou *via* un accès Internet [3].

1.1.2 Système d'exploitation

C'est un ensemble de programmes spécialisés qui permet l'utilisation des ressources matérielles d'un ou plusieurs ordinateurs. Il assure le démarrage (*Boot*) de l'ordinateur et l'exécution des logiciels applicatifs. Il remplit deux fonctions majeures : d'une part, la gestion des ressources matérielles (la mémoire, le processeur et les périphériques), en répartissant leur utilisation entre les différents logiciels ; d'autre part, la fourniture de services aux applications [4].

1.1.3 Infrastructure hyper convergée

L'hyper-convergence consiste à concevoir des architectures de Systèmes d'Informations modulaires et évolutives intégrant au sein d'un même nœud le traitement, le stockage, le réseau et la virtualisation. Les ressources de calcul, de stockage et de réseau sont dissociées de l'infrastructure car celle-ci va être déterminée par une couche logiciel [5].

1.1.4 L'automatisation informatique

L'automatisation informatique, aussi appelée automatisation de l'infrastructure, consiste à utiliser des logiciels pour créer des instructions et des processus reproductibles dans le but de remplacer ou de réduire l'interaction humaine avec les systèmes informatiques [6].

1.1.5 Softwares defined

Les softwares defined représentent une nouvelle classe de produits pour lesquels le logiciel est la cible principale et est utilisé pour fournir la solution plutôt que le matériel.

1.2 L'évolution du Datacenter

Le Datacenter a considérablement évolué au cours des dernières décennies. Les sections suivantes examineront chaque époque en détail.

1.2.1 L'ère de l'ordinateur central

Le mainframe a régné pendant de nombreuses années et a jeté les bases de notre situation actuelle. Il a permis aux entreprises de tirer parti des caractéristiques clés telles que processeur, mémoire principale et stockage convergés de manière native, redondance interne d'ingénierie.

Mais l'ordinateur central a également introduit des problèmes comme les coûts élevés d'acquisition d'infrastructures et la complexité inhérente, et aussi un manque de flexibilité et des environnements hautement cloisonnés [10].

1.2.2 Le passage aux serveurs autonomes

Avec les ordinateurs centraux, il était très difficile pour les entreprises de tirer parti de ces fonctionnalités, ce qui a en partie conduit à l'introduction de serveurs autonomes. Les principales caractéristiques des serveurs autonomes inclus les CPUs, les mémoire principale et stockage DAS (direct-attached storage), plus de flexibilité et un accès sur le réseau, ce qui signifie que les applications sont passées du client léger (traitement sur le serveur) au client lourd (traitement côté utilisateur / client).

Mais par contre, ces serveurs autonomes ont introduit plus de problèmes tels que l'utilisation faible ou inégale des ressources, le serveur est devenu un point de défaillance unique pour le calcul et le stockage [10][17].

1.2.3 Stockage centralisé

Les entreprises ont toujours besoin de gagner de l'argent et les données sont un élément clé de ce puzzle. Avec le stockage DAS (Direct-Attached Storage), les entreprises avaient besoin de plus d'espace que celui disponible localement, ou de la haute disponibilité des données (HA), pour lesquelles une défaillance du serveur n'entraînerait aucune indisponibilité des données.

Le stockage centralisé a remplacé à la fois l'ordinateur central et le serveur autonome par des pools de stockage plus importants, partageables, offrant également une protection des données.

Avec le stockage centralisé, ils ont gagné une meilleure utilisation du stockage, en éliminant le risque de perte de serveur causant la perte de données [19].

1.2.4 La virtualisation

Historique

Une bonne part des travaux sur la virtualisation fut développée au centre scientifique de Cambridge d'IBM en collaboration avec le MIT, où fut mis au point le système expérimental CP/CMS, devenant ensuite le produit (nommé Hyper viseur) VM/CMS. Par la suite, les mainframes (serveurs IBM) ont été capables de virtualiser leurs systèmes d'exploitation avec des technologies spécifiques et propriétaires, à la fois logicielles et matérielles.

Dans la deuxième moitié des années 1980 et au début des années 1990, on a créé des embryons de virtualisation sur des ordinateurs personnels. Ces solutions pouvaient être soit purement logicielles, soit couplées à du matériel additionnel (ajout de processeur, carte réseau, etc.). Et c'est sur des ordinateurs Amiga équipé de processeur hétérogène comme le 80386 et 80486, 68xxx, et PPC qu'il était possible de lancer d'autres OS comme un Windows, Mac OS, voire des solutions Linux. Le tout en multitâche sous AmigaOS. Pour les PC, il y avait des émulateurs comme le SideCar et PC Task. Sur Macintosh, Emplant et ShapeShifter.

Dans la seconde moitié des années 1990, les émulateurs sur x86 des vieilles machines des années 1980 ont connu un énorme succès, notamment les ordinateurs Atari, Amiga, Amstrad.

La société VMware développa et popularisa à la fin des années 1990 et au début des années 2000 un système propriétaire de virtualisation logicielle des architectures de type x86 pour les architectures de type x86. Les logiciels libres Xen, KVM, QEMU, Bochs, Linux-VServer, Virtual Box et les logiciels propriétaires mais gratuits VirtualPC, Virtual Server et VMware Server ont achevé la popularisation de la virtualisation dans le monde x86 [18].

Définition et caractéristiques

La virtualisation est caractérisée par sa facilitation de la restauration, les réductions des coûts de matériel, de maintenance, énergétiques et de refroidissement; sa rapidité pour déployer un nouveau serveur, et un monitoring simplifié! Elle permet aussi d'avoir facilement un environnement de test, par des captures instantanées.

Enfin, la virtualisation est avantageuse pour sa continuité de service « haute disponibilité » Car pour une machine physique, une machine virtuel n'est qu'une suite de fichiers, donc facile à déplacer [18][19].

1.2.5 Cloud Computing

Sa première énonciation date de 1960 (John McCarthy), mais sa réelle mise en application a commencé au début des années 2000. Salesforce.com fut le premier hébergeur de Cloud en 1999, suivi en 2002 par Amazon. Le Cloud Computing met en œuvre l'idée de l'informatique utilitaire du type service public, proposée par John McCarthy en 1961 qui suggère que la technologie informatique partagée pourrait construire un bel avenir dans lequel la puissance de calcul et même les applications spécifiques pourraient être vendues comme un service public. L'apparition du Cloud Computing vient d'une évolution de certaines technologies telles que la virtualisation du matériel informatique, les services web, ou l'architecture orientée services SOA (Service Oriented Architecture). La virtualisation a été la première pierre de l'ère du Cloud Computing. En effet, cette notion permet d'optimiser les ressources matérielles en les partageant entre plusieurs environnements dans le but de pouvoir exécuter plusieurs systèmes « virtuels » sur une seule ressource physique et fournir une couche supplémentaire d'abstraction du matériel. Le Cloud Computing est donc la juxtaposition de ces technologies pour passer à la vitesse supérieure sur l'exploitation de données à travers Internet [12].

1.3 Le Cloud Computing

1.3.1 Définitions

Le Cloud Computing est un terme général employé pour désigner la livraison de ressources et de services à la demande par internet, Il répond à des besoins de plus en plus complexes. Et malgré cela, il n'y a pas eu une définition normalisée, chaque entreprise et institution a fait sa propre définition.

National Institute of Standards and Technology(NIST):"Le Cloud Computing est un modèle permettant un accès réseau omniprésent, pratique et sur demande à un pool partagé de ressources configurables (réseaux, serveurs, stockage,

applications et services) pouvant être rapidement provisionnées et libérées avec un effort de gestion minimal.” [1].

Tout le monde est mis d’accord, que le Cloud Computing n’est pas une technologie, mais il repose sur diverses technologies: systèmes d’exploitation, virtualisation, outils de gestion et d’automatisation.

1.3.2 Différence entre Cloud Computing et virtualisation

Bien que la virtualisation contribue à la création de Cloud, elle ne peut pas être assimilée au Cloud Computing. Il est facile de confondre ces deux concepts, notamment parce qu'ils reposent sur le même principe, à savoir isoler les ressources du matériel afin de créer un environnement optimal [12].

Toutefois, il ne s'agit pas de la même chose, La virtualisation est une technologie qui sépare les fonctions du matériel, par contre le cloud computing s'apparente davantage à une solution qui repose sur cette séparation.

1.3.3 Services fournis par les clouds

IaaS, *abréviation de Infrastructure as a Service*, fournit au client des moyens de calcul et de stockage, des équilibreurs de charge et des systèmes d'exploitation, les capacités réseau et d’autres ressources indispensables (pare-feu, cache...).Le client accède aux ressources via une interface web ou via des api, voir figure 1.1 [20]. Amazon EC2 et Google Compute Engine sont des exemples [12].

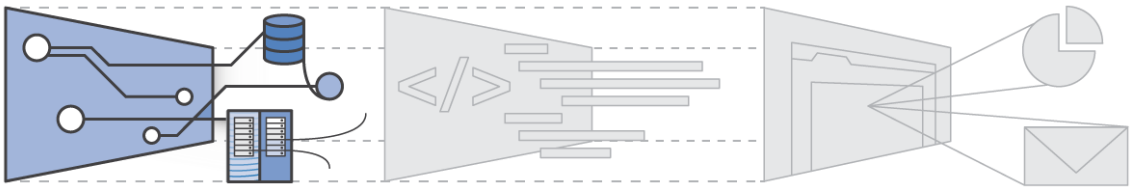

FIGURE 1.1 CLOUD COMPUTING IAAS MODEL 1

PaaS, *qui signifie Platform as a Service*, désigne une plateforme Cloud de développement et de déploiement logiciel, qui fournit l'ensemble du matériel et des logiciels gérés par le fournisseur de service, tels que les librairie, les bases de données,

ainsi que le système d'exploitation, le middleware et l'environnement d'exécution, voir la figure 1.2 [20]. Openshift et Google App Engine sont des exemples.[12]

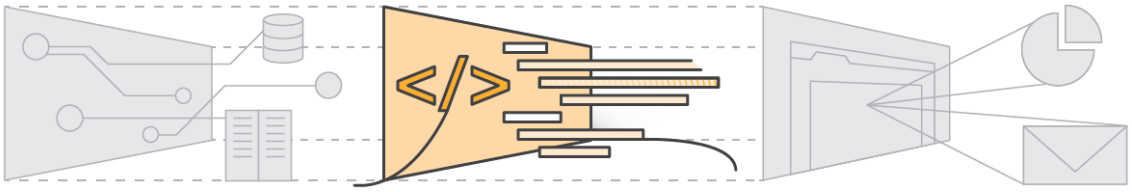

FIGURE 1.2 CLOUD COMPUTING PAAS MODEL

SaaS, *abréviation de Software as a Service*, offre des logiciels et des applications via internet, tels que les emails, les outils de collaboration et les jeux vidéo. Les utilisateurs s'abonnent à un logiciel et y accèdent par le web ou par les API du fournisseur. Et sur ce, le client utilise efficacement les ressources en minimisant les coûts initiaux et de maintenance, voir la figure 1.3 [20]. Salesforce(CRM),Gmail, Trello ou Facebook en sont des exemples [12].

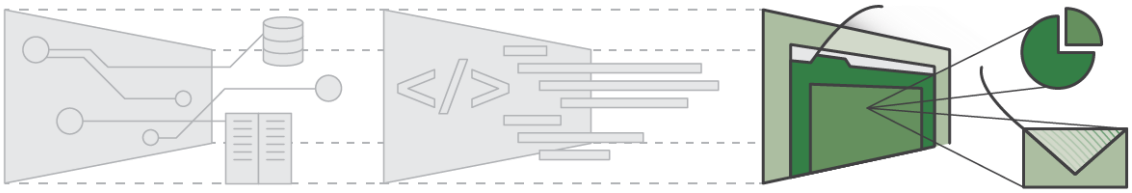

FIGURE 1.3 CLOUD COMPUTING SAAS MODEL

XaaS, Les services SaaS, PaaS et IaaS énumérés ci-dessus sont les formes les plus courantes de XaaS, les ressources fournies faisant référence respectivement à un logiciel, une plate-forme et une infrastructure. Tout en tant que service (XaaS) fait référence à la diversité croissante des services fournis sur Internet plutôt que localement ou sur site [12].

1.3.4 Les modèles du Cloud

Cloud privé (On promise): Un Cloud privé est un pool de ressources virtuelles, provenant de systèmes dédiés à leurs utilisateurs et gérés par ceux-ci, qu'il est possible d'approvisionner et d'allouer automatiquement via une interface en libre-service. Donc les ressource allouées, appartiennent aux client [12].

Cloud public: Un Cloud public est un pool de ressources virtuelles, créées à partir de matériel détenu et géré par une entreprise tierce, qui sont automatiquement approvisionnées et allouées à différents clients via une interface en libre-service. Cette méthode simple permet de faire évoluer les charges de travail soumises à des fluctuations imprévues de la demande [12].

Cloud hybride: Un Cloud hybride est la combinaison d'un ou plusieurs environnements de Cloud public et privé. Il s'agit d'un pool de ressources virtuelles développées en partie à partir de matériel possédé et géré par une entreprise tierce et en partie à partir de matériel appartenant à l'entreprise qui utilise le Cloud [12].

Cloud communautaire: Le Cloud communautaire, c'est la possibilité pour plusieurs entités ou membres d'organisations ayant les mêmes besoins d'utiliser une seule et unique solution Cloud**.** Il permet à plusieurs entreprises de partager l'ensemble des ressources et d'avoir accès aux mêmes données [12].

1.3.5 Solutions Cloud

Il existe plusieurs solutions Cloud, celles pour un Cloud public, ou privé.

Cloud privé: Pour faire un Cloud privé, il y a des solutions open source et d'autres propriétaires. Voici quelques exemples:

- **Solution open source:**

✓ OpenNebula
✓ Eucalyptus
✓ OpenStack
✓ Cloudstack

- S**olution propriétaire:**

✓ VMware VCloud
✓ Cisco CloudCenter
✓ Microsoft Azure Stack

Cloud public: Pour un Cloud public, le client a énormément de choix. Les fournisseurs leaders du marché sont**:** Amazon, Google,Microsoft Azure et IBM.

1.3.6 Avantages et bénéfices du Cloud Computing

- **Pour les clients**

Les clients peuvent gagner énormément d'avantage en utilisant le Cloud, principalement en terme d'argent, ils vont remplacez les coûts en capitaux par

des coûts variables plutôt que d'investir massivement dans des centres de données et des serveurs avant de savoir comment vont les utiliser, ils ne vont payer que lorsqu'ils consomment des ressources. Aussi, ils Cessent de deviner les capacité nécessaires des besoins en termes de ressources d'infrastructure. Avant de déployer une application, lorsqu'ils doivent décider des capacités à allouer, ils se retrouvent bien souvent à court ou, au contraire, avec des ressources inutilisées qui coûtent cher. Grâce au Cloud Computing, il ne faut plus se soucier de cet aspect. Qu'elles soient faibles ou importantes, le client peut accéder à toutes les ressources qu'il souhaite, et les augmenter ou les réduire en fonction de ces besoins en à peine quelques minutes. Plus encore, le client bénéfice d'une vitesse et d'une souplesse accrue dans un environnement de Cloud Computing, il suffit d'un clic pour obtenir de nouvelles ressources informatiques. Ainsi, le temps nécessaire pour rendre ces ressources disponibles pour les développeurs passe de quelques semaines à quelques minutes à peine. L'organisation voit ainsi sa souplesse augmenter considérablement, car le coût et les temps nécessaires pour expérimenter et développer sont fortement réduits. En outre, on ne va plus dépensez l'argent pour le fonctionnement et la maintenance de centres de données, et le client peut établir sa connexion de n'importe où et avoir accès à ses données immédiatement, sans passer par la mise en place d'un VPN (réseau privé virtuel) dans l'entreprise.

- **Pour les fournisseurs**

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

1.3.7 Les inconvénients du Cloud

Le mode « pay-as-you-go » est pratique, mais il peut s'avérer onéreux si celui ci n'est pas maîtrisé en interne, laissé en libre service sans contrôle des coûts, il peut s'avérer exorbitant. De plus, Le Cloud utilise de manière intensive le transfert de données, il faut avoir une connexion très performante. Plusieurs cas peuvent faire que le cloud sera inadapté à une entreprise, par exemple, Si la connexion ne dispose pas d'un débit garanti, une coupure peut survenir, privant l'entreprise de tous les accès au cloud, et donc à toutes les applications et les données. En outre, la plateforme cloud, si elle est externe (non installée sur le réseau interne ou avec une ouverture extérieure) doit être suffisamment sécurisée pour éviter le risque d'intrusion, de vol des données par piratage. L'autre risque est qu'un utilisateur oublie de se déconnecter sur un appareil accessible par des éléments externes à l'organisation. Il faut dans ce cas prévoir une déconnexion automatique en cas de non-activité du compte et bien segmenter les droits d'utilisateurs afin que ces derniers ne puissent accéder qu'aux données des projets dans lesquels ils sont impliqués. Plus généralement, une clause de confidentialité et la confiance dans son personnel sont primordiales pour que les données ne fuitent pas de manière volontaire.

1.3.8 La sécurité dans le Cloud

La sécurité est la préoccupation la plus importante et la plus ancienne dans l'informatique des nuages (le Cloud), le premier souci est que, le principe fondamental du Cloud public est la multi-location des ressources. Certains pensent que louer les ressources à plusieurs clients, voir plusieurs organisations, peut faire du Cloud public une cible plus attrayante pour les cybercriminels et les collecteurs de données.

Lorsqu'une entreprise commence à planifier une transition ou un déploiement dans le Cloud, certaines considérations spécifiques à la sécurité doivent être abordées. Ces considérations doivent faire partie du processus global de planification dans le Cloud et non pas simplement un audit de sécurité ou une évaluation une fois que tout est déployé.

La première étape à prendre en compte concerne les systèmes informatiques, les applications et les données qui doivent rester dans un centre de données de l'entreprise.

Pas toutes les charges de travail sont des candidats idéaux pour la transition vers le Cloud en raison de préoccupations liées à la sensibilité des données et à la criticité des missions. Les experts en sécurité doivent travailler avec les propriétaires d'applications et d'entreprises pour déterminer les applications et les données qui peuvent être facilement déplacé vers un Cloud afin d'avoir une liste de priorités avec des notations spécifiques sur les classifications de sensibilité, de réglementation ou de sécurité souhaitées.

Disposer de ces informations est essentiel pour les unités commerciales informatiques et financières de toute organisation pour aider à déterminer le type de cloud à utiliser ou à déployer, à calculer la capacité initiale de l'infrastructure, des modèles de retour sur investissement et de modèle financier, ainsi qu'à établir des décisions globales de gouvernance opérationnelle.

La prochaine étape consiste à examiner le ou les modèles de cloud à acquérir ou à déployer. Bien que les services de cloud public offrent une sécurité solide de l'infrastructure, ils n'ont souvent pas le niveau de sécurité ou de personnalisation dont l'organisation pourrait avoir besoin. Un cloud privé peut être fortement personnalisé pour répondre aux exigences de sécurité ou de fonctionnalités, mais on doit tout de même contrôler les coûts ,la dérivé de contenu et la construction excessive du cloud initial.

Dans un Cloud hybride ou multifournisseurs, la sécurité devient encore plus compliquée lors de l'évaluation, de la sélection et de l'utilisation de services cloud qui ne se trouvent pas tous au même niveau de sécurité ou d'accréditation. Si une entreprise sait dès le départ qu'elle est susceptible de former un cloud hybride avec plusieurs fournisseurs de cloud, elle doit envisager de définir une posture de sécurité minimale acceptable que tous les fournisseurs doivent respecter. Ensuite, évaluez et sélectionnez certains fournisseurs de cloud offrant une conformité renforcée en matière de sécurité pour des applications ou des charges de travail critiques, permettant ainsi au système de gestion de cloud hybride de fournir des services aux fournisseurs de cloud appropriés. Notez que le système de gestion en cloud hybride ou un courtier en cloud peut avoir accès aux données stockées sur chacun des multiples fournisseurs de cloud. Par conséquent, le courtier en cloud (Broker, l'intermédiaire entre le client et le fournisseur)

ou le fournisseur hybride doit être accrédité à un niveau égal aux exigences de sécurité de votre fournisseur [7,8].

Conclusion

Le cloud computing a atteint un point de basculement où il a passé la phase de battage publicitaire et est entré dans une phase où les entreprises commencent à accepter le fait que le Cloud est réel et qu'il est là pour rester. Comme toute autre idée ou technologie, il n'ya pas de solution miracle. Les entreprises qui auront du succès dans le Cloud sont celles qui comprennent les différences entre le service Cloud et les modèles de déploiement et qui font les bons choix en fonction des besoins de leur entreprise. Ils doivent comprendre les exigences techniques pour la création de services de Cloud et mettre en œuvre une architecture qui répond à chaque exigence. Ces entreprises doivent également faire face au changement organisationnel et lutter contre la résistance, les lacunes en matière de compétences, les nouveaux processus, etc. Comme pour toute autre transformation à laquelle nous sommes confrontés au fil des ans, tout se résume aux personnes, aux processus et à la technologie.

Dans le chapitre suivant, nous abordons un système de gestion de Cloud open source avec tous ses composants et fonctionnalités.

CHAPITRE II :

OPENSTACK

Introduction

2.1 Openstack

Openstack est né d'une globale collaboration entre développeurs et des experts du Cloud computing afin de produire l'omniprésence des plateformes Cloud pour le public. Le projet vise à fournir des solutions pour tous les types de Cloud en étant simple à mettre en œuvre, extrêmement évolutive et riche en fonctionnalités.

Openstack [9] est une IAAS (**Infrastructures as a service**) plateforme de Cloud Computing open source qui supporte tous types d'environnements Cloud. Il contrôle plusieurs Sources de stockage informatique et de ressources réseau dans un Data Center.

Openstack transforme tous cet ensemble de ressources en périphériques de stockage et de réseau au sein d'un ou de plusieurs Data Center en source de ressources. Ces ressources peuvent être gérés et consommés à partir d'un seul outil nommé Openstack.

La fondation Openstack a décidé de définir logiquement le projet en deux services , Les services de base et les services optionnel.

2.2 Composants et Fonctionnalités

Le projet Openstack comprend plusieurs sous-projets interdépendants qui aident à gérer différents aspects des ressources matérielles, y compris le service de calcul, le stockage, les réseaux et autres services associés, chacun offrant son propre ensemble d'API pour faciliter l'intégration [11].

Tous ces sous-projets ou service sont classés dans deux type de service , service de base ou optionnel.

Les *services de base* sont les plus importants pour déployer un Cloud sous Openstack. Ces services sont : **Nova**, **Glance**, **Keystone**, **Cinder** ou **Swift** et **Neutron**.

Les *services optionnels* sont ceux qui peuvent être choisis ou pas dans un Cloud Openstack, cela reste en fonction des cas d'utilisation. Ces services sont : **Horizon**, **Ceilometer**, **Heat**, **Sahara**, **Trove**, **Manilla**, **ironic**...etc.

La figure 2.1 [9] Illustre l'architecture conceptuelle OpenStack, avec des interactions entre ses composants logiciels. En tant que plateforme cloud dédiée à IaaS, OpenStack a des machines virtuelles en son centre, fournies par le module Nova. Les VM sont

entourées par d'autres services, y compris la connectivité réseau gérée par Neutron, système de gestion images stockées par Glance, services de stockage fournis par Swift et Cinder. Keystone est responsable de l'authentification de l'ensemble du système OpenStack, Horizon fournit une interface de gestion Web à tous les autres services.

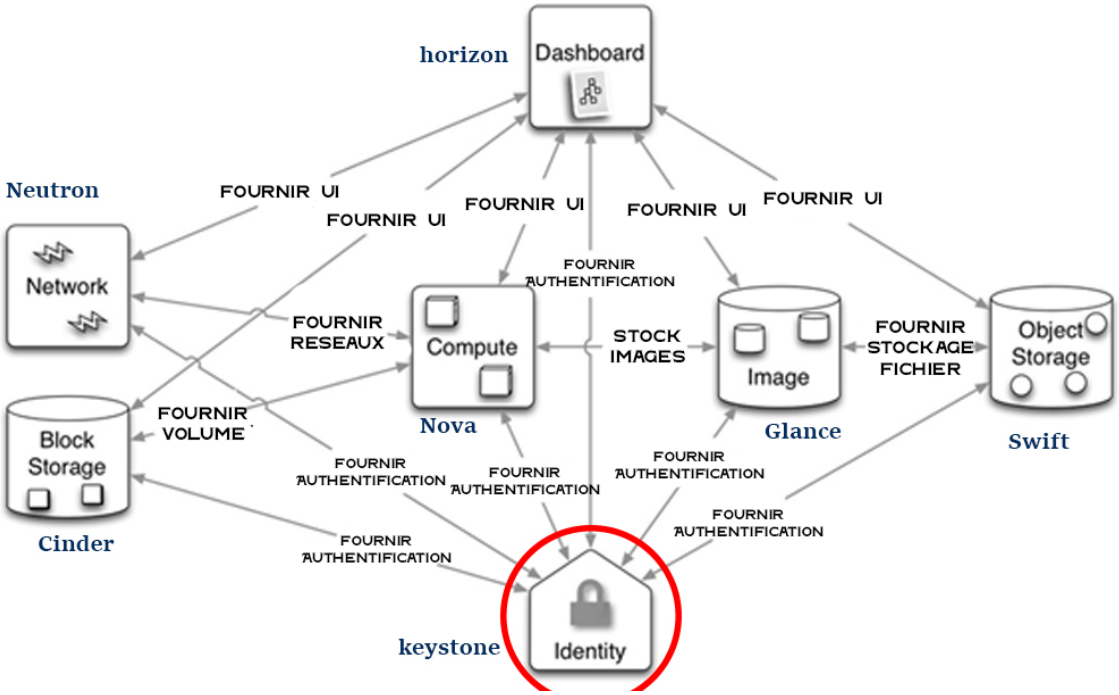

FIGURE 2.1 ARCHITECTURE OPENSTACK

2.2.1 Service de calcul (Nova)

Appelé aussi le service de calcul (compute), le projet le plus important dans la base de Openstack, il représente l'hyperviseur pour le monde extérieur.

Nova est conçu pour gérer et automatiser les ressources de calcul, compatible avec plusieurs technologies de virtualisation disponible ainsi que les configuration informatique haut performance.

Il touche la base avec presque tous les composants de openstack comme service d'imagerie, stockage, réseaux…

Nova fournit une gestion de cycle de vie des instances avec son API riche, une instance signifie une machine virtuelle. Il est responsable de la création, la planification, la suppression et la sélection de la taille et des caractéristique, par exemple le nombre de processeur virtuels ou les giga octets de Ram ainsi de suite.

Étant donné que les machines virtuelles sont créées par l'hyperviseur choisi, cela

fait de nova le gestionnaire d'hyperviseur, il gérera ces machines virtuelles sur demande à travers l'hyperviseur.

Nova fonctionne comme un ensemble de démons sur des serveurs Linux existants pour fournir ce service.

2.2.2 Service de stockage

OpenStack prend également en charge deux types de stockage, à savoir Object stockage et stockage en bloc.

Cinder (Block Storage) : Créer et fournir des périphériques de bloc de virtualisation persistants indépendamment de toute instance particulière, les volumes peuvent être attachés à une seule instance à la fois mais peuvent être détachés ou rattachés à une autre instance tout en conservant toutes les données, à la manière d'un lecteur USB, le stockage de volume est indépendant de tout cas particulier et c'est la persistance.

Les volumes sont créés par l'utilisateur et respectent les limites de quota et de disponibilité et peuvent être de n'importe quelle taille. Le stockage de blocs est implémenté dans Openstack par le projet de stockage de blocs de Cinder.

Swift (Object Storage) : Swift fonctionne de manière très simple et c'est ce qui le rend si puissant et évolutif, contrairement à Cinder qui fonctionne au niveau du bloc , les objets et les fichiers sont stockés, répliqués, et distribués sur plusieurs serveurs du cluster.

Il s'agit d'un stockage de fichiers très basique, mais il peut être très puissant, il fournit une devise à coût élevé, ce qui signifie qu'il peut servir une tonne d'utilisateurs en même temps.

Swift fournit une API pour transmettre le contenu.

2.2.3 Service de réseau

Neutron est le projet sous Openstack qui fourni la connectivité réseau en tant que service pour les instances en marche sur un hyperviseur. Il abstrait les port réseaux et sous-réseaux pour les rendre programmable à l'aide des API [11].

Il a une architecture modulaire à déployer soit de manière centralisée ou distribuée selon le besoins. Ce service fonctionne en permettant aux utilisateurs de créer leur propre réseau virtuel isolé puis l'attacher à des interfaces.

Ces réseaux pourraient rester isolés ou être connectés au reste du monde en fonction des besoins, La connectivité entre les réseaux internes est réalisée en créant des routeurs virtuels pour créer des racines (root) entre eux. Même un routeur virtuel peut être connecté au monde extérieur (public).

Un réseau et une adresse IP flottante pourraient être attribués à une instance pour fournir un accès externe.

Neutron est le responsable de mettre toute la configuration nécessaire pour lier les réseau et les mettre en place, il suffit de le programmer via des API.

2.2.4 Service de gestion d'image (Le Projet Glance)

Glance est utilisé pour stocker des images de disques des machines virtuelles. C'est à Glance que nous stockons ces images prédéfinies qui aident à gérer les instances de machines virtuelles.

C'est un service très stable et pas si complexe, Glance est composé de plusieurs composants qui effectuent les tâches de demande pour stocker et récupérer les images et les métadonnées associées demandées par un client.

Lorsque vous ajoutez une image à Glance, vous devez spécifier le format de disque d'image de la machine virtuels et le format de conteneur. Glance prend en charge une grande variété de formats d'image, notamment: RAW, VHD, VDI, VMDK, OVF, QCOW2, etc.

2.2.5 Service de tableau de bord (Horizon)

Horizon est la structure qui fournit l'interface Web permettant une gestion facile des instances et autres configuration dans Openstack.Il permet aux utilisateurs d'accéder à des machines virtuelles, manipuler leurs réseaux privés et d'autres ressources OpenStack via l'interface utilisateur graphique basée sur le Web (écrite en Python en utilisant Django).Cela en fait l'un des composants les plus populaires de openstack , car c'est une plateforme idéale pour tout utilisateur effectuant leur service librement [11].

La figure 2.2 illustre le service Dashboard, présentant différente fonctions pour manipuler le service d'authentification.

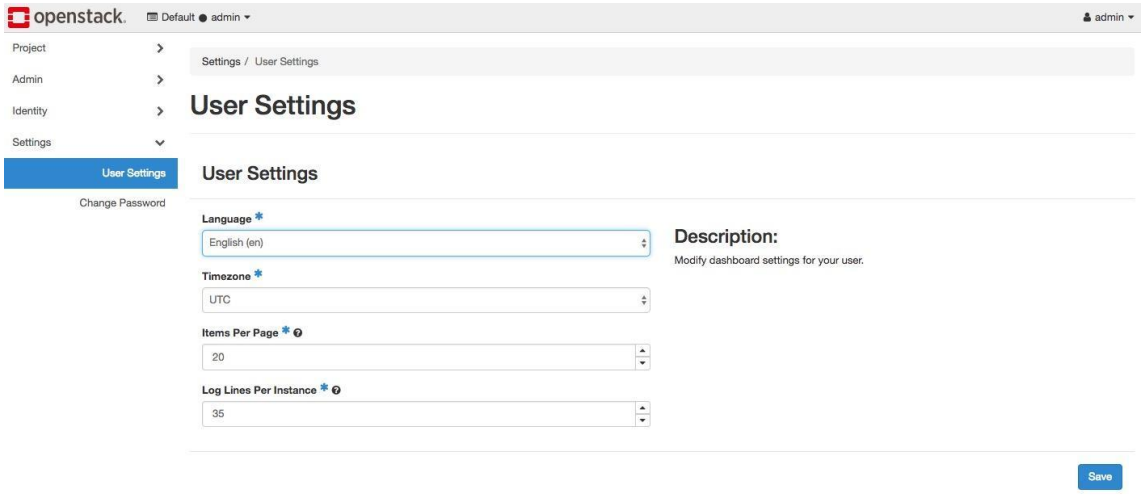

FIGURE 2.2 OPENSTACK DASHBOARD GESTION DES UTILISATEURS

2.3 Service de communication et d'intégration

OpenStack est un système distribué, composé de plusieurs projets et modules. Chaque module est conçu avec le "Shared Nothing Architecture" et est fonctionnellement indépendant de l'autre. Un module comme Nova ou Neutron est composé de plusieurs composants qui fournissent ensemble ses fonctionnalités.

Comme avec tout autre système distribué, la fonctionnalité de OpenStack dans son ensemble dépend fortement de la manière dont ses services internes sont intégrés [11], ce qui dépend à son tour de la capacité de ses modules et composants à communiquer entre eux. Il existe 3 mécanismes principaux permettant la communication et l'intégration des services d'OpenStack: API RESTful, Appel de procédure distante et RabbitMQ.

2.3.1 Restful API

REST[13] est un style architectural et une approche aux communications souvent utilisées dans le développement de services Web. Chacun des modules de base d'OpenStack exposent une ou plusieurs interfaces RESTful pour interagir avec le monde extérieur. En utilisant les API RESTful, OpenStack offre un accès aux utilisateurs de différents moyens, soit par l'interface de ligne de commande (CLI), cURL ou via le client REST. Chacun de ces projets dispose d'un service API en tant que point de terminaison auquel le client peut accéder (par exemple, openstack-nova-api, openstack-glance-api) pour pouvoir accepter la demande REST de son clients,

utilisateurs ou autres modules. En tant que telle, l'API RESTful est un moyen efficace de laisser différents modules OpenStack communiquer.

2.3.2 Remote Procedure Call

L'appel de procédure distante, ou RPC, permet une communication inter-processus qui permet aux clients de déclencher l'exécution de "routines" dans un emplacement distant. Les modules OpenStack comme Nova (nova-compute, nova-api, nova-scheduler), Neutron (serveur de neutrons, neutron-openvswith-agent) ou Cinder (cinder-scheduler, cinder-volume) utilise beaucoup le protocole RPC pour sa communication intra-module. Par exemple, après Neutron le serveur de neutrons reçoit une demande (RESTful) pour créer un nouveau réseau, il demande le plug-in disponible (par exemple, ml2plugin) pour envoyer à son tour un appel RPC à l'agent correspondant (par exemple neutron-openvswith-agent).

2.3.3 RabbitMQ

Les appels RPC reposent sur un canal ou sur un mécanisme de messagerie par lequel ils sont acheminés vers les processus consommateurs (c'est-à-dire les consommateurs). Les requêtes RPC sont regroupées dans des messages qui sont envoyés à un courtier de messages qui les transmet ensuite aux consommateurs.C'est là qu'un courtier de messagerie tel que RabbitMQ entre en jeu. RabbitMQ est une implémentation open-source de la norme AMQP (Advanced Message Queue Protocol). L'AMQP est conçu pour faciliter le courtage de messages entre diff érents processus, applications du même système, voire entre systèmes communiquant en transmettant des messages. Dans la plate-forme OpenStack, l'AMQP est utilisé pour établir un mécanisme de communication interne efficace entre les composants du même module OpenStack, par exemple Nova, Neutron ou Cinder.

2.4 KVM

Kernel-based Virtual Machine, ou KVM, est une solution de virtualisation complète pour Linux et est fourni avec le noyau Linux depuis la version 2.6.20 du noyau. KVM est activé par l'exécution du matérielle d'émulation basée sur QEMU avec le mode d'accélération KVM activé. KVM est un mode de fonctionnement spécial de QEMU qui utilise la fonctionnalité de virtualisation assistée par le processeur(Hardware

Virtual Machine ou HVM) pour effectuer la virtualisation matérielle via ses modules de noyau spécifiques au processeur. KVM est parmi plusieurs plates-formes d'hyperviseur compatibles avec OpenStack.

2.5 Service d’identité (keystone)

Comme nous connaissons tous le moyen le plus simple d'authentifier, un utilisateur doit demander des informations d'identification puis vérifier leurs existence sur la base de donnée, ce qui est très courant et utiliser sur tout type d'authentification sur les appareils personnels ou autre.

Quand cela vient à beaucoup de dispositif distincts travaillant ensemble comme sur openstack nous devons reconsidérer cette approches.
Le problème principale est l'incapacité d'utiliser une identité d'un utilisateur unique autorisé partout.

keystone est le projet openstack qui fournit une authentification centralisé aux utilisateur et les projets, il fournit un répertoire central des services et des utilisateurs ainsi que leurs rôles et autorisation, à chaque action,on doit s'authentifier à partir de keystone (vérification du jeton).

La figure 2.3 représente un diagramme de séquence qui montre les interactions entre les services et le rôle important de Keystone qui vérifie à chaque fois le jeton d’authentification.

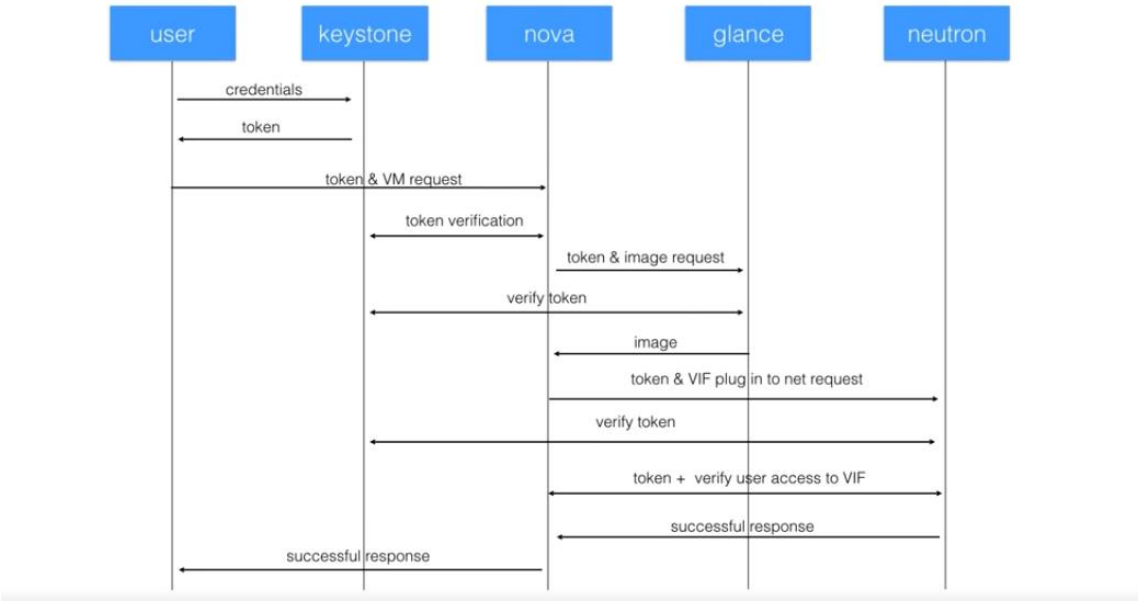

FIGURE 2.3 DIAGRAMME DE SEQUENCE , SERVICE INTERACTION

Dans Keystone, nous avons les concepts de locataires, de rôles et d'utilisateurs. Un client hébergé est comme un projet et contient des ressources telles que des utilisateurs, des images et des instances, ainsi que des réseaux qui ne sont connus que de ce projet particulier. Un utilisateur peut appartenir à un ou plusieurs locataires et peut basculer entre ces projets pour accéder à ces ressources. Différents rôles peuvent être attribués aux utilisateurs d'un locataire. Dans le scénario le plus élémentaire, un utilisateur peut se voir attribuer le rôle d'administrateur ou simplement être membre. Lorsqu'un utilisateur dispose de privilèges d'administrateur au sein d'un client hébergé, il peut utiliser des fonctionnalités pouvant affecter le client hébergé (telles que la modification de réseaux externes), tandis qu'un utilisateur normal se voit attribuer le rôle de membre, généralement affecté à l'exécution de rôles liés à l'utilisateur, tels que la multiplication des instances et la création de volumes.

Conclusion

Openstack est un logiciel open source de premier plan utilisé pour exécuter des clouds privés. Sa popularité a augmenté de façon exponentielle depuis sa fondation par Rackspace et la NASA. Le résultat de cette communauté engagée est stupéfiant, ce qui permet à de nombreuses nouvelles fonctionnalités de se retrouver dans Openstack à chaque version. Le projet a maintenant une taille telle que personne ne peut vraiment

connaître les détails de chaque service. Lors d'un travail sur un projet aussi complexe, il est inévitable de rencontrer des problèmes, des bugs, des erreurs, des problèmes et tout simplement de vieux problèmes. Dans ce chapitre, nous avons appris les caractéristiques de conception d'Openstack et ses composants essentiels.

Le chapitre suivant traite plus en détail le concept et la solution technologique pour gérer l'authentification dans Openstack. Ainsi, nous allons voir comment déployer une infrastructure efficace et manipuler le service de l'identité pour lui permettre de communiquer avec un serveur d'authentification centralisé.

CHAPITRE III :

Déploiement de OPENSTACK et intégration avec LDAP

Introduction

Nous avons vu que pour satisfaire la demande toujours plus grande en migration vers le Cloud Computing, tout en réalisant des économies d'échelle, on réduisant les coûts liés à l'approvisionnement électrique du matériel et aux besoins de refroidissement, Dans les moyennes et grandes entreprises, sachant que ces types d'entreprises possèdent des quantités d'informations massive. Cette migration qui représente un point sensible et des fois effrayant a cause de la sensibilité et l'importance des données rend cette opération un challenge de réussir cette migration tout en respectant les normes et on gardant une partie importante du système existant.

En parlant des systèmes existants, le système d'authentification fait son apparence en premier degré. L'indisponibilité de ce système peut causer des dégâts techniques ou failles au niveau de sécurité qui entraine par la suite des pertes financières.

Le fait qu'une entreprise possède au minimum, deux systèmes qui peuvent interagir entre eux, il est préférable de centraliser le système d'authentification, pour unifier la gestion des authentifications et des autorisations. Cela permet également de centraliser la gestion de la politique de sécurité.

Pour répondre à ces problèmes, nous avons réalisé dans notre travail de fin d'étude, une intégration d'un annuaire LDAP existant avec le service identité openstack (keystone). Un annuaire LDAP est avant tout un annuaire. Il permet donc d'obtenir des informations sur une personne enregistrée comme son adresse email, son numéro de téléphone, son service, ou n'importe quel autre renseignement que l'on aura juger bon de stocker dans la base. La recherche peut se faire selon de multiples critères.

Une telle intégration permet:

- **Unification et la Centralisation:** de nombreuse applications et systèmes sont capables d'interroger un même annuaire LDAP.
- **La fiabilité:** Des mécanismes de réplication (en cours de standardisation) entre des annuaires maîtres et des réplicas permettent d'assurer une bonne fiabilité au système.
- **La sécurité :** Les annuaires supportent pour la plupart des mécanismes de chiffrage des connexions (SSL TLS).

Les droits d'accès aux différentes données de l'annuaire peuvent être précisés finement grâce à des ACL. Le support de nombreux environnements de développement.

3.1 Keystone

3.1.1 De quoi est il composé ?

Keystone est composé de plusieurs services, que nous détaillerons dans ce qui suit. La figure 3.1[21] détaille l'architecture de Keystone.

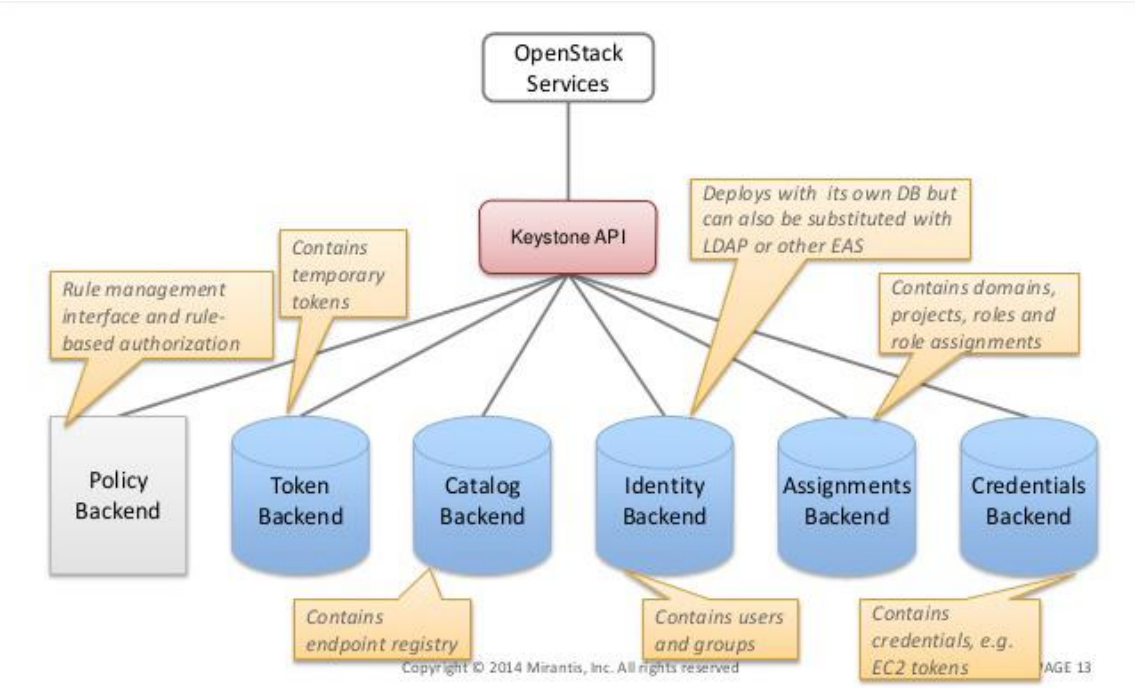

FIGURE 3.1 COMPOSANT DE KEYSTONE OPENSTACK

3.1.1.1 Service D'identité

Le service d'identité fournit une validation des informations d'authentification et des données sur les utilisateurs et les groupes. Dans le cas de base, ces données sont gérées par le service Identité, ce qui lui permet de gérer également toutes les opérations CRUD associées à ces données. Dans des cas plus complexes, les données sont plutôt gérées par un service en arrière plan faisant autorité. Un exemple de cela serait lorsque le service d'identité agit comme une interface pour LDAP **(LDAP est** un protocole réseau offrant une autre forme d'accès aux serveurs d'annuaire existants). Dans ce cas, le

serveur LDAP est la source de la vérité et le rôle du service d'identité est de relayer ces informations avec précision [9].

Les utilisateurs: représentent un consommateur d'API individuel. Un utilisateur lui même doit appartenir à un domaine spécifique. Par conséquent, tous les noms d'utilisateur ne sont pas globalement uniques, mais uniques à leur domaine.

Les groupes: sont un conteneur représentant une collection d'utilisateurs. Un groupe lui même doit appartenir à un domaine spécifique. Par conséquent, tous les noms de groupe ne sont pas globalement uniques, mais uniques à leur domaine.

3.1.1.2 Service de Resource

Le service de ressources fournit des données sur les projets et les domaines [9].

Les projets: représentent l'unité de base de la propriété dans Openstack, en ce sens que toutes les ressources dans Openstack doivent appartenir à un projet spécifique. Un projet lui-même doit appartenir à un domaine spécifique. Par conséquent, tous les noms de projets ne sont pas globalement uniques, mais uniques à leur domaine. Si le domaine d'un projet n'est pas spécifié, il est ajouté au domaine par défaut.

Les domaines: sont un conteneur de haut niveau pour les projets, les utilisateurs et les groupes. Chacun appartient à exactement un domaine. Chaque domaine définit un espace de nom dans lequel un attribut de nom visible par l'API existe. Keystone fournit un domaine par défaut, nommé judicieusement «Par défaut».

En raison de leur architecture de conteneur, les domaines peuvent être utilisés comme un moyen de déléguer la gestion des ressources Openstack. Un utilisateur d'un domaine peut toujours accéder aux ressources d'un autre domaine si une affectation appropriée est accordée.

3.1.1.3 Service d' Affectation

Le service d'affectation fournit des données sur les rôles et les attributions de rôles [9].

Les rôles: déterminent le niveau d'autorisation que l'utilisateur final peut obtenir. Les rôles peuvent être attribués au niveau du domaine ou du projet. Un rôle peut être attribué au niveau de l'utilisateur individuel ou du groupe. Les noms de rôle sont uniques dans le domaine propriétaire.

Assignations de rôles: Un triplet possédant un rôle, une ressource et une identité.

3.1.1.4 Service de Jeton

Le service de jeton valide et gère les jetons utilisés pour l'authentification des demandes une fois que les informations d'identité de l'utilisateur ont déjà été vérifiées [9].

3.1.1.5 Catalogue

Le service de catalogue fournit un registre de points de terminaison utilisé pour la découverte de points de terminaison [9].

REMARQUE

- Nom de domaine. Globalement unique dans tous les domaines.
- Nom de rôle. Unique dans le domaine propriétaire.
- Nom d'utilisateur. Unique dans le domaine propriétaire.
- Nom du projet. Unique dans le domaine propriétaire.
- Nom de groupe. Unique dans le domaine propriétaire.

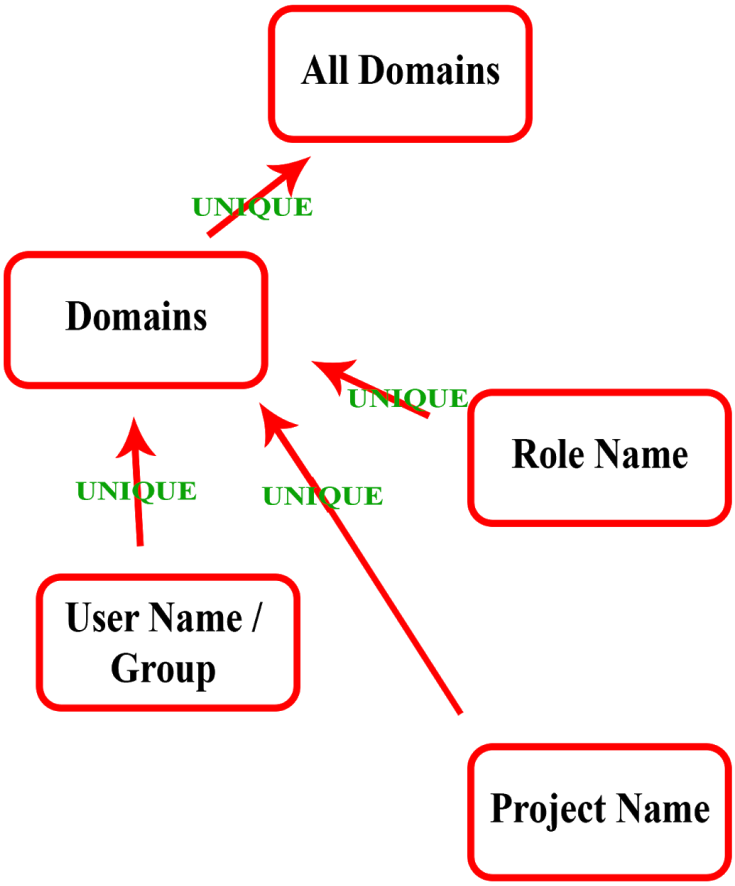

Figure 3. 2 : Unicité De La Donnée Dans Keystone

3.2 LDAP

3.2.1 Définition

Le terme LDAP signifie Lightweight Directory Access Protocol. Comme son nom l'indique, LDAP a été conçu à l'origine pour être un protocole réseau offrant une autre forme d'accès aux serveurs d'annuaire existants, mais à mesure que l'idée de LDAP et des technologies qui l'entouraient a mûri, le terme LDAP est devenu synonyme d'un type spécifique de l'architecture d'annuaire. Nous utilisons le terme LDAP lorsque nous faisons référence à des services d'annuaire conformes à cette architecture, tels que définis dans les spécifications LDAP [14].

La version actuelle de LDAP est LDAP v3 (version 3), une norme développée en 1997 sous le numéro RFC 2251 et largement mise en œuvre dans l'ensemble du secteur. La spécification d'origine a récemment été mise à jour (juin 2006) et les RFC 4510 à 4519 fournissent une spécification clarifiée et beaucoup plus cohérente pour LDAP [15].

Bien que les annuaires en général, et les annuaires LDAP en particulier, ne soient ni nouveaux ni rares dans le monde des technologies de l'information, les technologies motrices ne sont certainement pas aussi bien comprises que les proches parents comme la base de données relationnelle. L'un des objectifs de cette section est d'introduire et de clarifier le fonctionnement et l'utilisation d'un annuaire LDAP.

Dans cette section, nous présenterons certains des concepts importants pour comprendre le protocole LDAP. Le meilleur endroit pour commencer est l'idée du répertoire.

3.2.2 Qu'est-ce qu'un annuaire?

Lorsque nous pensons à un répertoire, nous évoquons des images d'annuaires téléphoniques ou de carnets d'adresses. Nous utilisons ces annuaires pour trouver des informations sur des individus ou des organisations. Un serveur de répertoire est également utilisé de cette manière. Il conserve des informations sur un ensemble d'entités (entités telles que des personnes ou des organisations) et fournit des services pour accéder à ces informations [14].

Bien entendu, un serveur d'annuaire doit également disposer de moyens lui permettant d'ajouter, de modifier et de supprimer des informations. Mais, même si un annuaire téléphonique est supposé être avant tout une ressource pour la lecture, les informations d'un serveur d'annuaire sont supposées être lues plus souvent qu'écrites. Cette hypothèse sur l'utilisation d'un serveur d'annuaire est codifiée ou résumée dans la phrase "lecture élevée, écriture faible". Par conséquent, de nombreuses applications de la technologie LDAP sont orientées vers la lecture et la recherche d'informations.

Certains types de serveurs d'annuaire (imaginer une implémentation simple d'un carnet d'adresses sur un serveur) fournissent simplement un service étroit et spécifique. Un serveur de répertoire à usage unique, tel qu'un carnet d'adresses en ligne, peut ne stocker qu'un type de données très spécifique, tel que les numéros de téléphone, les adresses et les informations de messagerie d'un groupe de personnes. De tels répertoires

ne sont pas extensibles. Au lieu de cela, ils sont à but unique. Mais LDAP a été conçu pour être un serveur de répertoire à usage général. Il n'a pas été conçu pour capturer un type de données spécifique. Au lieu de cela, il a été conçu pour donner aux développeurs la possibilité de définir, de manière claire et précise les données que le répertoire doit stocker. Un tel serveur d'annuaire générique devrait pouvoir stocker de nombreux types d'informations. D'ailleurs, il devrait pouvoir stocker différents types d'informations sur différents types d'entités.

3.2.3 La structure d'une entrée de répertoire

3.2.3.1 Un nom unique: le DN

La stratégie adoptée par LDAP consiste à créer un nom unique pour chaque enregistrement du répertoire, il doit avoir un nom distinctif. Le nom distinctif est un terme LDAP important. Généralement, il est abrégé en DN.

Dans un annuaire LDAP, le concepteur d'annuaire est celui qui décide quels composants constitueront un DN, mais généralement, le DN indique où se trouve l'enregistrement dans l'annuaire, ainsi que certaines informations qui distinguent cet enregistrement des autres enregistrements proches. Cela distingue cet enregistrement des autres enregistrements proches [14].

Un DN, est alors composé d'une combinaison d'informations d'annuaire et ressemble à ceci:

```
dn: o=université, l=faculté, st=département, c=niveau
```

Comme il ressort de cet exemple, lors de la définition des champs qui composeront un DN, il est nécessaire de s'assurer que ces champs seront suffisamment affinés pour distinguer deux entrées différentes. Le DN est un élément important dans une entrée LDAP. Ensuite, nous examinerons de plus près l'idée d'une entrée LDAP et des composants qui la composent.

Un exemple d'entrée LDAP

Une entrée LDAP, ou enregistrement, est l'unité qui stocke des informations sur un élément individuel de l'annuaire. Ainsi un enregistrement dans un annuaire LDAP contient des informations sur une unité spécifique, bien que (puisque LDAP soit générique), la cible exacte de cette unité ne soit pas spécifiée. Ça peut être une personne, une entreprise, ou une entité virtuelle telle qu'un service Openstack.

Une entrée est composée d'un DN et d'un ou plusieurs attributs. Le DN sert d'identifiant unique dans une arborescence d'informations d'annuaire LDAP. Les attributs fournissent des informations sur cette entrée. Ça peut ressembler à ceci:

```
dn:         nom=Mohamed         ilyes,         f=technologie,
dep=informatique,  user=mohamedilyes,u=Abou Bekr Belkaid
nom: Mohamed ilyes
u: Abou Bekr Belkaid
ville: Tlemcen
f: technologie
dep: informatique
cp: 13000
n: M2
n: M1
user: mohamedilyes
objectclass: utilisateur
```

La première ligne est le DN. Toutes les autres lignes de cet enregistrement représentent des attributs. Un attribut décrit un type d'information spécifique. Il y a huit attributs ici dans notre exemple, représentant ce qui suit:

- ✓ Nom de l'organisation (o)
- ✓ Ville (ville)
- ✓ Faculté (f).
- ✓ Département (dep)
- ✓ Code postal (cp)
- ✓ Niveau (n)
- ✓ Nom d'utilisateur (user)
- ✓ Classe d'objet (objectclass), qui spécifie le type (ou les types) d'enregistrement de cette entrée

Dans tout enregistrement donné, un attribut peut avoir une ou plusieurs valeurs (en supposant que la définition de l'attribut autorise plus d'une valeur). L'enregistrement ci-dessus n'a qu'un seul attribut contenant plus d'une valeur. L'attribut n(niveau) a deux valeurs, chacune représentant un niveau différent.

3.2.3.2 L’attribut Objectclass

Le dernier attribut de l'enregistrement donné est l'attribut objectclass. C'est un attribut spécial qui fournit des informations sur le type d'enregistrement (ou d'entrée) dont il s'agit [14][15].

Une classe d'objets détermine quels attributs peuvent être attribués à un enregistrement. La classe d'objet, utilisateur, indique que cet enregistrement décrit une un utilisateur. L'un des champs, le nom de l’utilisateur (nom), est obligatoire pour toute entrée avec une classe d'objet d’utilisateur

La classe d'objet autorise également plusieurs autres attributs non présents dans notre enregistrement, tels que adresse et numéro de téléphone…

Attributs opérationnels

En plus des attributs habituels, le serveur d’annuaire peut également associer des attributs opérationnels spéciaux à une entrée. Les attributs opérationnels sont utilisés par le serveur d’annuaire lui-même pour stocker des informations sur les entrées. De tels attributs ne sont pas conçus pour être utilisés par les utilisateurs finaux (même s’ils peuvent parfois être utiles) et ne sont généralement pas renvoyés lors des recherches LDAP [15].

3.2.3.3 L'arbre d'information d'annuaire

Jusqu'à présent, nous avons vu la structure des entrées LDAP en ce qui concerne la forme des enregistrements. Mais maintenant, nous allons présenter l’une des principales caractéristiques d’un serveur d’annuaire LDAP.

Dans un annuaire LDAP, la structure organisationnelle est sophistiquée, les informations sont organisées en une ou plusieurs hiérarchies où, en haut de la hiérarchie, se trouve une entrée de base et d'autres entrées sont organisées en structures arborescentes sous l'entrée de base. Chaque nœud de la hiérarchie est une entrée, avec un DN et plusieurs attributs. Cette collection d'entrées organisée hiérarchiquement est appelée une arborescence d'informations de répertoire, parfois simplement appelée une arborescence de répertoires ou DIT (directory information tree) [15].

Pour comprendre cette méthode d'organisation de l'information, considérons l'organigramme de l’université. Le sommet de la hiérarchie est l’université elle-même. En dessous de cela, il y a un certain nombre de facultés et de départements, ainsi que

des employés, des enseignants, des étudiants, et d'autres personnes ayant une affiliation officielle avec l'université. Nous pouvons dessiner ceci comme une hiérarchie:

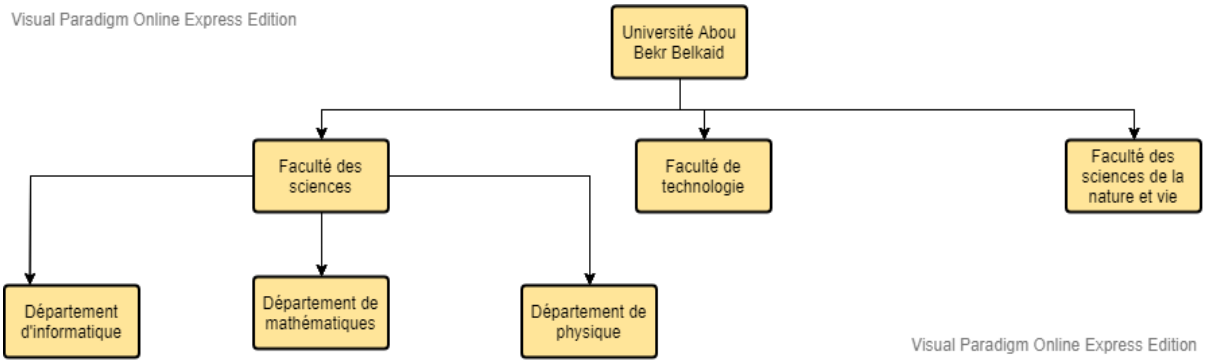

FIGURE 3. 3 DIAGRAMME D'UNE ARBORESCENCE LDAP

Les annuaires LDAP stockent également des données dans des relations hiérarchiques. L'entrée racine se trouve en haut de l'arborescence d'informations sur l'annuaire. En dessous se trouve une entrée subordonnée qui, à son tour, peut avoir ses propres entrées subordonnées. Chacun de ces enregistrements a son propre DN et ses propres attributs.

3.2.4 Que faire avec un serveur LDAP

Nous avons décrit ce qu'est un annuaire LDAP, mais il est également utile de regarder à quoi sert un annuaire LDAP. Quelle est la fonction d'un serveur LDAP? Quel problème est-il censé résoudre?

La première réponse, et la plus évidente, est que LDAP est conçu pour fournir un répertoire numérique. Bien sûr, il y a une part de vérité à cela et les serveurs LDAP peuvent effectivement être utilisés de cette manière. Mais il en va de même pour les bases de données relationnelles et des structures de données encore plus fondamentales. Nous pourrions développer cette réponse en soulignant que LDAP fournit une couche robuste de services recherche avec des filtres complexes, représentant des entités complexes avec des attributs, permettant un accès plus fin aux données, fournissant des services d'annuaire sophistiqués.

L’utilisation la plus courante d’un LDAP, basée sur une conception de LDAP en tant que type étroit d’outil de gestion d’entreprise, est celle d’autorité centrale sur les utilisateurs, les groupes et les comptes dans le réseau.

Un annuaire LDAP stocke des informations sur chaque compte utilisateur, telles que le nom d'utilisateur, mot de passe, nom complet et adresse électronique. D'autres services sur le réseau, de postes de travail aux serveurs de messagerie en passant par les applications Web, peuvent utiliser LDAP en tant que source d'informations d'utilisateur faisant autorité. Les applications peuvent authentifier les utilisateurs par rapport à l'annuaire. Un seul compte d'utilisateur peut être partagé entre plusieurs applications d'entreprise (peut-être toutes).

Comparé à une base de données relationnelle, LDAP peut également être considéré comme un système de stockage. Plutôt que de présenter les données dans des structures tabulaires, LDAP stocke les entrées dans une hiérarchie (comme un système de fichiers). Les relations de base dans un LDAP sont constituées de la relation supérieur à subordonné (un à plusieurs) et de la relation subordonné à supérieur (un à un), bien que d'autres relations puissent être utilisées. La lecture et l'écriture dans les bases de données, sont prises en charge avec des filtres sophistiqués et des structures de données telles que LDIF (LDAP Data Inter-change Format) dans LDAP. Et les annuaires LDAP, écoutent souvent sur des sockets réseau pour fournir des services à d'autres applications.

3.4 Pourquoi intégrer LDAP avec OPENSTACK

Sachant que notre université (UABT) utilise un système d’authentification basé sur LDAP, et qui est le cas de plusieurs d’autres organisations. Cela nous a poussé à l’étudier et l’intégrer avec notre Cloud basé sur openstack afin de préserver ce système d’identification existant et garder le bon fonctionnement des autres applications qui interagissent avec ce système

La figure 3.4 explique l’architecture déployée dans notre cas

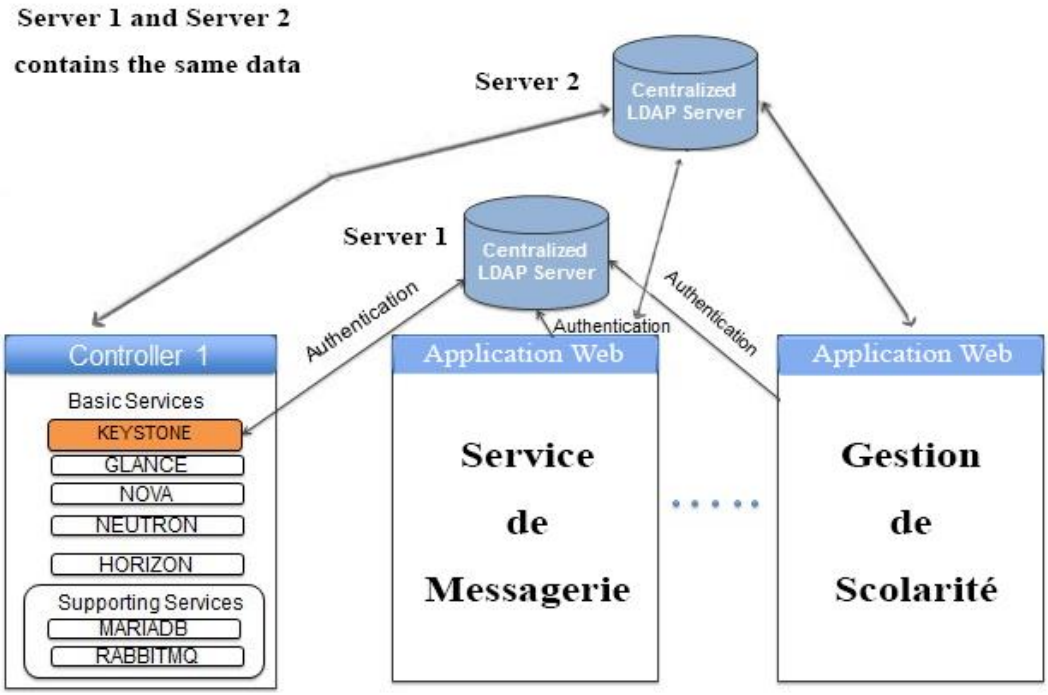

FIGURE 3.4 DEUX SERVEUR LDAP ET 3 APPLICATIONS

Nous allons découvrir comment redéfinir le service d'identité dans OpenStack et permettre un accès de gestion des utilisateurs plus rentable et sécurisé aux ressources situées sur différents points d'extrémité d'entreprise. Sachant que lors d'un déploiement d'un nouveau cloud basé sur openstack dans une entreprise, nous avons intérêt à garder l'authentification centralisée de confiance déjà existante car les utilisateurs s'authentifient, normalement, avec un seul compte sur toutes les applications dans l'entreprise. De plus, la plupart des entreprises demande une stratégie pour diviser les utilisateurs pour chaque service, de sorte qu'il peut y avoir plus d'un serveur d'authentification en place. En exploitant le concept de domaine dans OpenStack, cette division logique peut être conservée et reflétée pour la même entreprise, par plusieurs serveurs LDAP.

Le concept de fédération offre un moyen de réunir différentes parties sous un même toit centralisé. La plupart des entreprises préfèrent exposer une plate-forme unifiée aux utilisateurs internes de différents fournisseurs de services. L'un des cas d'utilisation de fédération les plus courants concerne les systèmes de fédération d'identité et d'authentification. Une organisation peut avoir différents services autour de son infrastructure informatique qui nécessitent une authentification et une autorisation pour chaque utilisateur privilégié.

L'implémentation de nombreux serveurs de base de données pour chaque service augmenterait potentiellement le risque de sécurité lié au mappage de plusieurs comptes pour chaque service utilisateur. Cela peut facilement entraîner la perte de trace de chaque compte individuel pour chaque service lors du départ d'un utilisateur. De plus, gérer l'identité séparément pour chaque service par un système différent peut être très déroutant pour les utilisateurs et représente une tâche de cauchemar administratif.

La fédération d'identités est en train de devenir un moyen courant pour de nombreuses organisations d'étendre leurs services cloud à une extension hybride. Cela constitue un moyen simple de fournir un accès à travers une configuration et des environnements multi-cloud [9, 11].

Le mécanisme de fédération d'identité permet aux utilisateurs de continuer à utiliser leurs informations d'authentification par rapport à la configuration d'identité existante. Les administrateurs ne s'inquièteront pas de l'ajout d'une nouvelle logique d'authentification distincte ni de l'augmentation du niveau de complexité du système. Cela peut être réalisé simplement en tirant parti d'un IdP existant qui contient les comptes d'utilisateurs et mappe leur niveau d'accès à différents services dans les organisations appelées SP (service client qui utilise les informations d'identité pour gérer uniquement l'accès à des services spécifiques.). De cette manière, les utilisateurs pourront accéder à différents services configurés par l'administrateur, ce qui leur permettra de mieux maîtriser la gestion des informations d'identification.

3.5 Installation et Déploiement de Openstack

La distribution RedHat d'openstack ou Rdo est la distribution RPM d'openstack, Il s'agit d'une openstack supportée par la communauté et disponible gratuitement qui fonctionne sur Red Hat Enterprise Linux (RHEL) et ses dérivés tels que Centos et Fedora.[9][11][16]

Les composants openstack fonctionnent parfaitement sous RedHat et ses dérivés. Ils fournissent des outils d'isolation pour faciliter le déploiement d'Openstack.

PackStack est l'utilitaire que nous utiliserons dans ce projet pour installer openstack. Il utilise des modules de marionnettes (puppet) pour déployer automatiquement diverses parties de RDO sur un ou plusieurs serveurs préinstallés via

SSH, nous avons choisi Packstack car il est facile à installer en addition de la possibilité d'ajouter facilement d'autres nœuds au cluster à l'avenir.

Pour l'installation de notre environnements cloud openstack, nous avons utilisé un système d'exploitation Centos , Toutes les étapes sont fournies étape par étape pour la mise en œuvre d'une plate-forme cloud openstack de base à l'aide de l'outil de déploiement de packstack, et pour simplifier la mise en réseau et simplifier la configuration matérielle requise, tous les composants openstack seront déployés sur une seule machine. Le déploiement Openstack résultant consiste en un réseau public et un réseau privé sur une seule machine hébergeant une ou plusieurs instances avec un volume de stockage attaché.

Les services openstack installés incluent le stockage de blocs, le calcul, le tableau de bord (Horizon), l'identité (Keystone), l'image (Glance), la mise en réseau openstack (Neutron) et le stockage d'objets (Swift).

Pour tous les déploiements, vous aurez besoin d'une machine hôte virtuelle ou physique exécutant Centos, RedHat Enterprise Linux ou Fedora .Selon RedHat, 16 Go de RAM et 20 Go d'espace disque est le minimum pour commencer.

Pour L'installation de Centos sur machine virtuelle nous avons utilisé le logiciel Gratuit VirtualBox Sous Windows, Avec 9 Go de Ram, 1 Processeur Virtuel, 50 go de Disque et une configuration Bridge pour le Réseau.

Après installation, certaines configuration sur le nouveau système sont nécessaires pour le bon fonctionnement d'openstack, puis lancer l'installation par les commande suivant:

- Installer la dernière version du pack openstack

```
sudo yum install -y centos-release-openstack-stein
```

- Installer Packstack Installer

```
sudo yum install -y openstack-packstack
```

- Lancer le Program d'installation Packstack

```
sudo packstack --allinone
```

L’installation peut durer 35 Minutes ou Plus selon la capacité, la puissance de la machine utilisée et la bande passante d’internet.

Après une installation complète et réussie, le système est accessible par l'adresse IP de la machine virtuelle via un navigateur web ou par le terminal via des lignes de commande, voici quelques exemples:

Via Terminal
Openstack User List

```
+----------------------------------------------+-------------+
| ID                                           | Name        |
+----------------------------------------------+-------------+
| b32eb4d5bf294b38823e0f5c1bceaf1d             | admin       |
| fb778aef5d7d41f28179981b97032e6a             | nova        |
| 30dd94e85a1542a89e84afb600b39a65             | cinder      |
| ab879f68df92442e8cca52c0405b6aa5             | glance      |
| 8ce15a7a52834bf78ee0084f2bfe875d             | placement   |
| a182534f7fac4f62a2b04e8a9d8aec7c             | neutron     |
| 7e4e7dde2be5456f8f85c6519fb62b41             | ceilometer  |
| b85b3417908b428ab66711bc5211cb3b             | swift       |
| e73f41a7ad61496d8a2c4863621d69d7             | gnocchi     |
| 6e77004a38314a5984701ef4daaaae25             | aodh        |
+----------------------------------------------+-------------+
```

Via Navigateur

Login page :

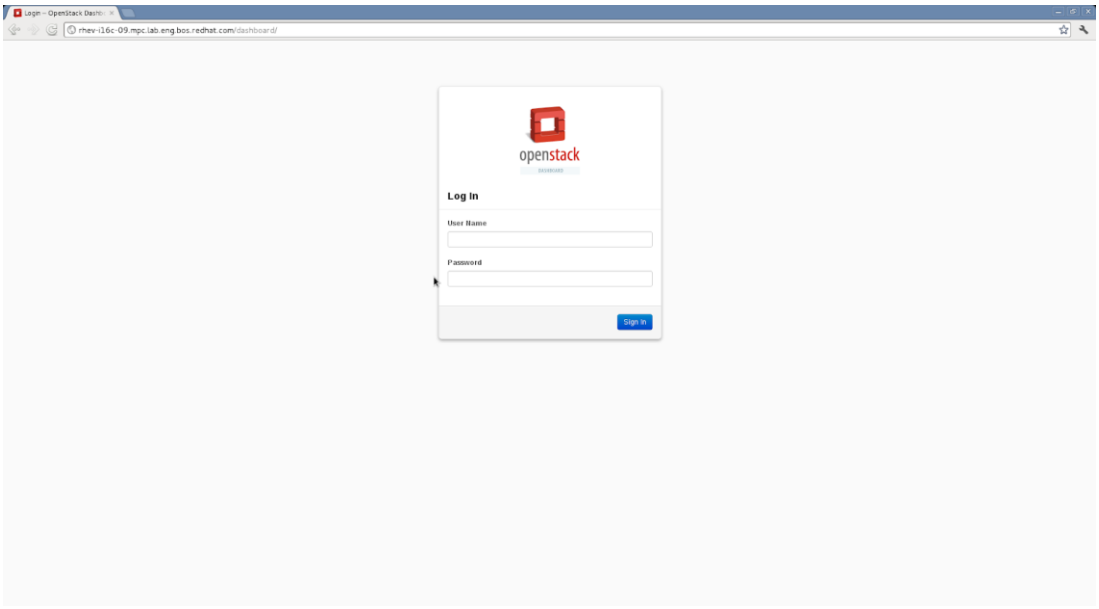

FIGURE 3.5 OPENSTACK LOGIN PAGE

Dashboard (Table de bord):

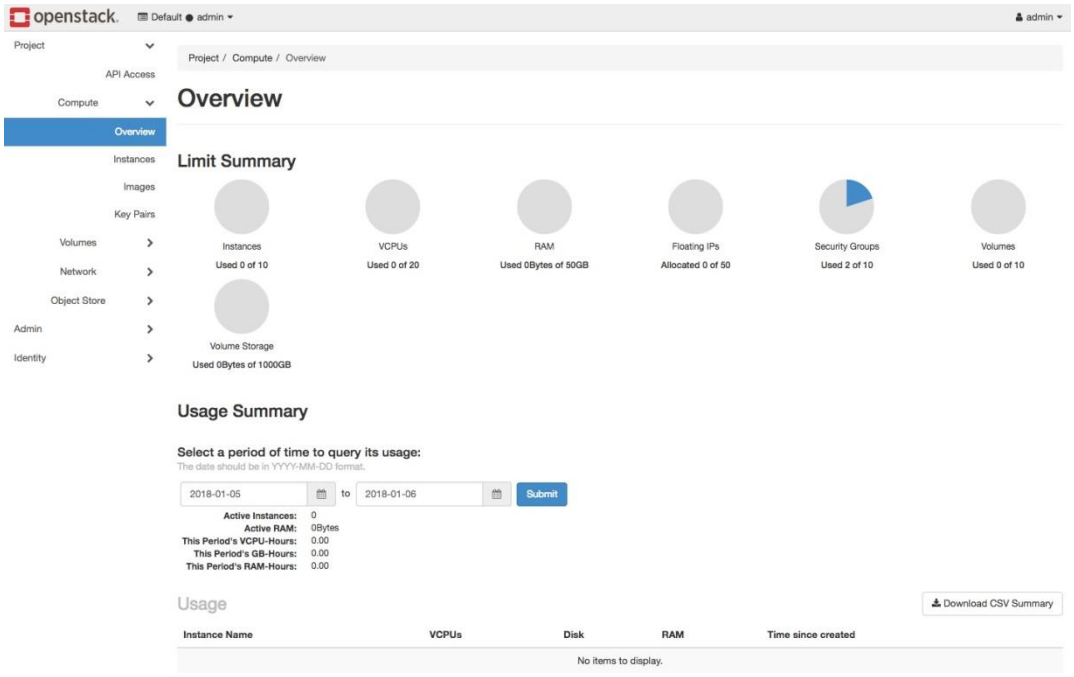

FIGURE 3.6 OPENSTACK DASHBOARD

3.6 Installation et Configuration de LDAP

Dans cette partie, nous expliquerons le processus d'installation et de configuration de la suite d'outils OpenLDAP. Ici, nous ne couvrirons que la configuration de base du serveur SLAPD. Cela servira de base pour les parties suivante de la liaison entre Keystone et LDAP.

OpenLDAP est géré par la fondation OpenLDAP. La fondation gère une suite d'outils que nous appellerons suite OpenLDAP, la suite OpenLDAP comprend les classes d'outils suivantes:

- Daemons (**slapd** and **slurpd**).
- Libraries (notably **libldap**)
- Client applications (**ldapsearch**, **ldapadd**, **ldapmodify**, and others)
- Supporting utilities (**slapcat**, **slapauth**, and others).

Dans notre cas nous avons installé OpenLDAP sur deux machines différentes pour assurer la disponibilité de notre serveur d'authentification. La première sur la même machine openstack, et la deuxième sur une machine distante déployée sur le

Cloud public d'amazon. Cela nous a poussé à introduire la notion du cloud hybride dans notre travail.

Remarque : Les deux machines contiennent le même système d'exploitation centos.

- Pour Installer LDAP lancer la commande suivant:

```
yum install openldap* ldap* nss* db* -y
```

openldap* db*: La suite logicielle openldap (implémentation open source du protocole LDAP) est composée de, openldap-server (Directory Server) openldap-clients (Fournit des outils pour communiquer avec le serveur comme ldapsearch, ldapadd, etc.)

nss*: Les applications nss (name service switch) utilisent le service NSS pour s'authentifier à l'aide de LDAP

slapd*: Slapd est le démon LDAP autonome. Il écoute les connexions LDAP sur un nombre quelconque de ports (389 par défaut), en réponse aux opérations LDAP qu'il reçoit sur ces connexions.

Lorsque l'installation est terminée, nous devons configurer le mot de passe LDAP et nom de domaine.

- Configurer mot de passe d'accès à notre annuaire:

```
cd /etc/openldap/slapd.d/cn=config/

slappasswd -h {CLEARTEXT}

New password:ca3792dd1a8f4465

Re-enter new password:ca3792dd1a8f4465

ca3792dd1a8f4465
```

- Configurer le nom de domaine qui représentera la base de notre annuaire dans notre cas nous avons choisi OPENSTACK.ORG:

```
sed -i -e '/olcRootDN/ s/cn=Manager,dc=my-domain,dc=com/cn=Manager,dc=openstack,dc=org/' olcDatabase={2}hdb.ldif

sed -i -e '/olcSuffix/ s/dc=my-domain,dc=com/dc=openstack,dc=org/' olcDatabase={2}hdb.ldif

sed -i -e '$ a olcRootPW: ca3792dd1a8f4465' olcDatabase={2}hdb.ldif

sed -i -e 's/cn=Manager,dc=my-domain,dc=com/cn=Manager,dc=openstack,dc=org/' olcDatabase\=\{1\}monitor.ldif
```

- Activer et demarrer le service SLAPD:

```
systemctl enable slapd
systemctl start slapd
```

3.7 Intégration LDAP avec KEYSTONE

Chaque composant dans openstack, tel qu'un utilisateur (admin , member …) ou un projet (nova, glance, swift ...), doit être identifié via la Keystone par un nom utilisateur et mot de passe, toutes ces informations sont stockées dans un fichier appelé **answer-file-packstack** qui a été générée avant l'installation de packstack all-in-one.

La figure 3.7 explique de façon simple comment les utilisateurs d'openstack sont identifiés:

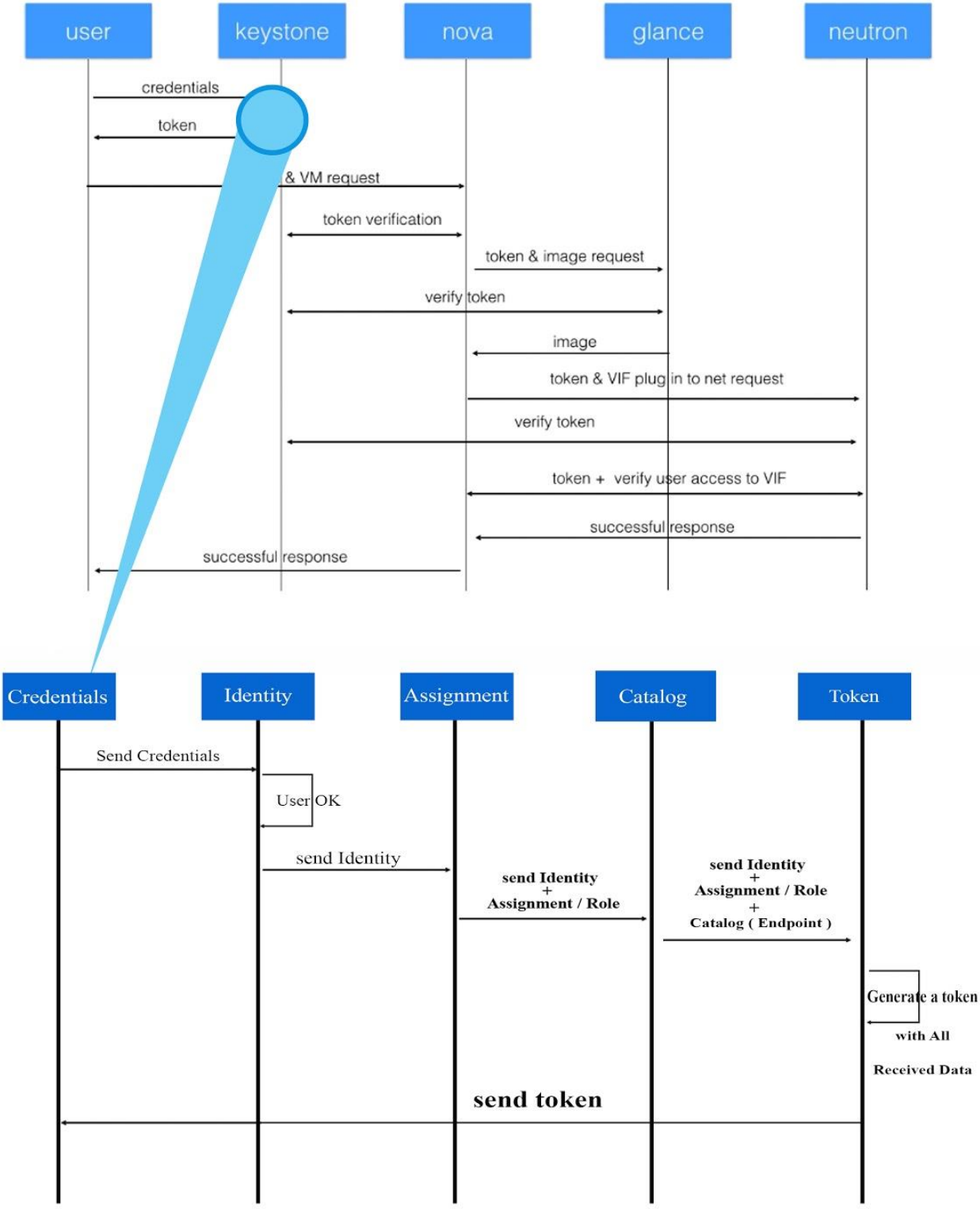

FIGURE 3.7 DIAGRAMME DE SEQUENCE KEYSTONE ET LE SERVICE IDENTITE

Pour effectuer l'intégration de la nouvelle base LDAP, nous devons changer le système d'identification dans keystone, c'est à dire keystone ne va plus utiliser la base SQL dans son sous service identité (SQL) et il va être rediriger vers l'annuaire LDAP créé (OPENSTACK.ORG).

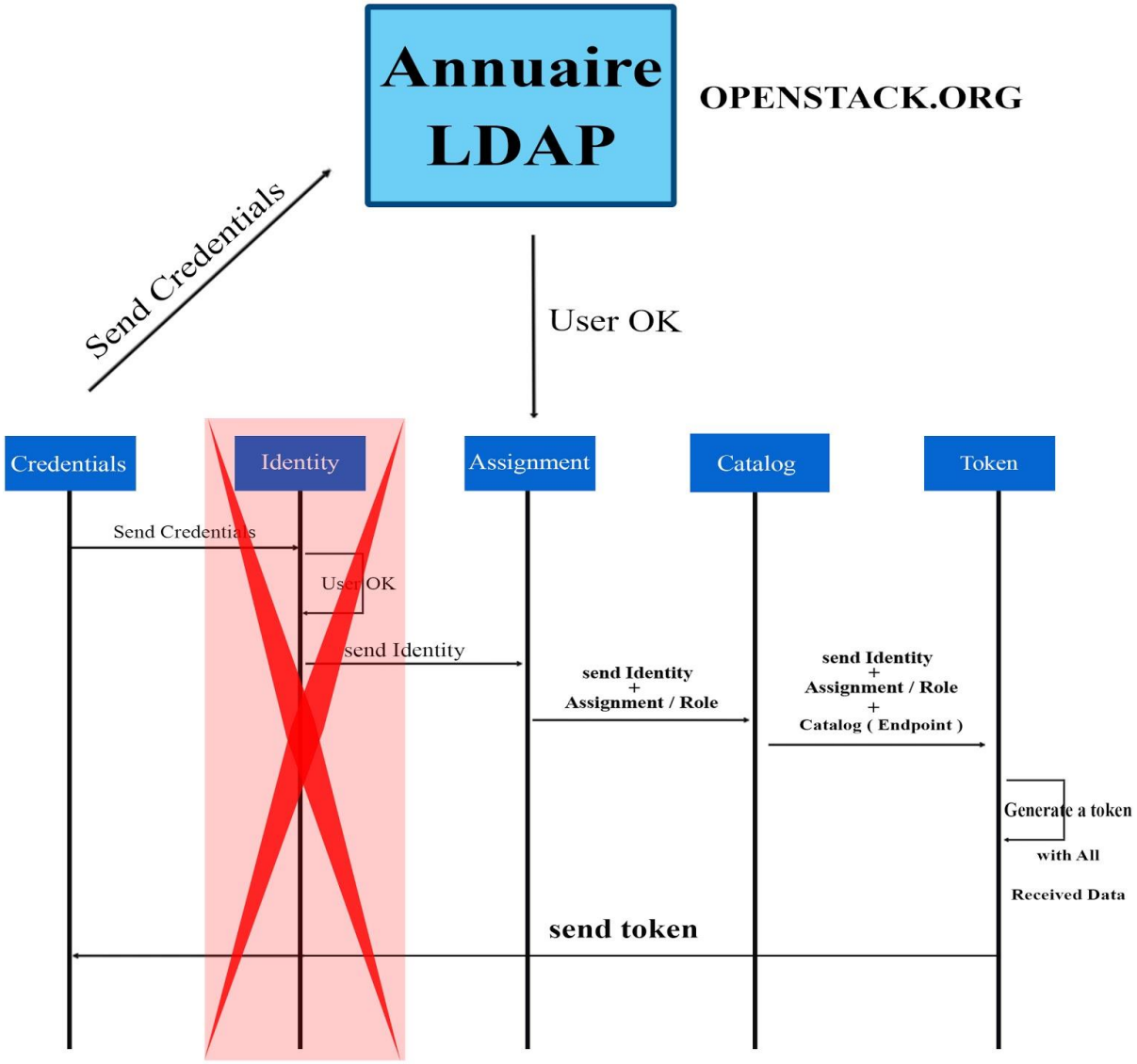

FIGURE 3.8 CHANGEMENT DU SERVICE IDENTITE PAR LDAP

La configuration requise pour redirectionner l'identification vers l'annuaire LDAP est de changer dans le fichier **keystone.conf** la section **[identity].**

Par défaut la section driver contient l'Attribut **driver = sql ,** il suffit de modifier la valeur à ldap => **driver = ldap** puis naviguer vers la section **[ldap]** et spécifier tous les

attributs et les règles nécessaire pour communiquer avec l'annuaire tel que url , cn , dn ,dc …

Exemple:

[ldap]
url = ldap://localhost // le chemin vers notre annuaire
user = cn=Manager,dc=openstack,dc=org // l'utilisateur manager ldap
password = ca3792dd1a8f4465 //mot de passe de l'utilisateur manager
suffix = cn=openstack,cn=org //le nom de domaine dans l'annuaire
user_attribute_ignore = enabled,email,tenants,default_project_id les champs qui peuvent être ignoré

Définition de l'arbre ldap

tree_dn = dc=openstack,dc=org
user_tree_dn = ou=Users,dc=openstack,dc=org
user_objectclass = inetOrgPerson
user_id_attribute = cn //définir l'id de notre user dans ldap
user_name_attribute = sn //définir le nom (username) dans ldap
user_pass_attribute = userPassword //définir le mot de passe de chaque user dans ldap

//Définir quelque règle à suivre
user_allow_create = True
user_allow_update = True

A la fin , il faut redémarrer le service **httpd** pour prendre en considération tous les changement sur le fichier **keystone.conf.**
A cette étape là, la liaison est correcte mais rien ne fonctionne, car comme on a expliqué avant, l'installation packstack inclut tous les projects liée tel que NOVA, GLANCE, SWIFT …, et qui représentent eux même des utilisateur dans le system openstack.

Ils doivent être identifier via keystone , alors que notre annuaire ldap est vide, il faut ajouter tous les utilisateurs existants dans l'ancienne base SQL.

Pour cela nous nous baserons sur le fichier cité avant **answer-file-packstack** et collecter toutes les données d'utilisateurs existants dans la table user de l'ancienne base SQL, puis les insérer dans notre annuaire ldap. Pour ajouter un enregistrement (utilisateur) dans ldap, nous créons un fichier ldif (LDAP Data Interchange Format), puis exécuter la commande ldapadd sur ce dernier.

Exemple : **nova.ldif**

 # nova, Users, openstack.org
dn: cn=fb778aef5d7d41f28179981b97032e6a,ou=Users,dc=openstack,dc=org
objectClass: person
objectClass: inetOrgPerson
sn: nova
cn: fb778aef5d7d41f28179981b97032e6a
userPassword: 60c20c1dcce14c2a
Puis exécuter avec la commande ldapadd.

```
ldapadd -x -D "cn=Manager, dc=openstack, dc=org" -w ca3792dd1a8f4465 <~/nova.ldif
```

Pour gagner plus de temps et faciliter la tâche on ajoute tous les utilisateur dans un seul fichier opns.ldif .

Après avoir ajouté tous les utilisateurs , il est maintenant possible de s'identifier à openstack via l'annuaire LDAP, et effectuer toute opération tel que ajouter, supprimer ou modifier.

3.8 Développement d'une interface pour manipuler LDAP users

Afin de faciliter l'utilisation avec LDAP et éviter les lignes de commande, nous avons créé une interface graphique web développé avec php, et qui va permettre aux administrateurs d'ajouter, supprimer et assigner les rôles/projets à leurs utilisateurs.

Dashboard

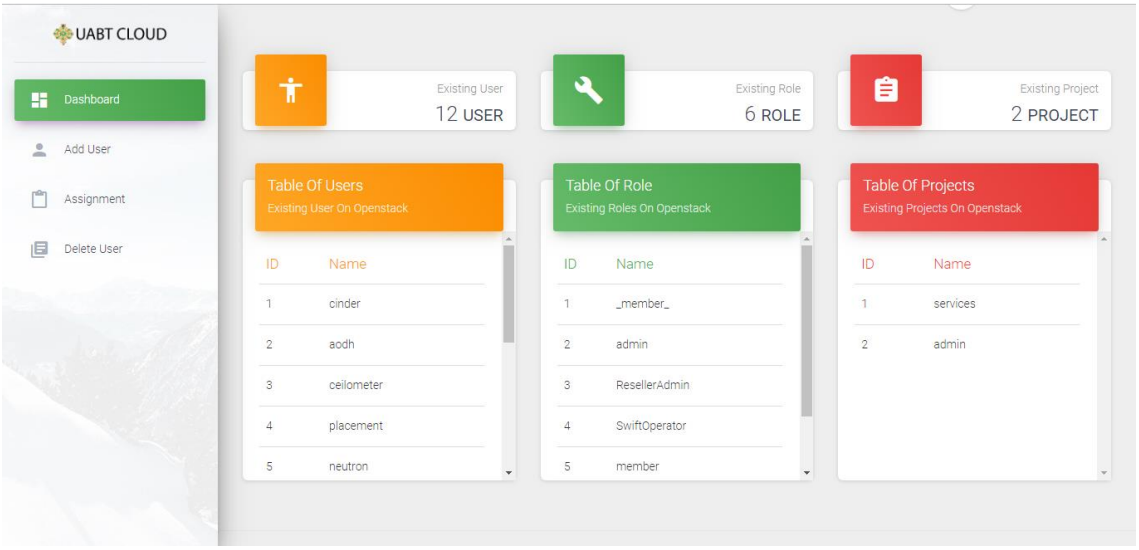

FIGURE 3.9 LDAP DASHBOARD

Ajouter

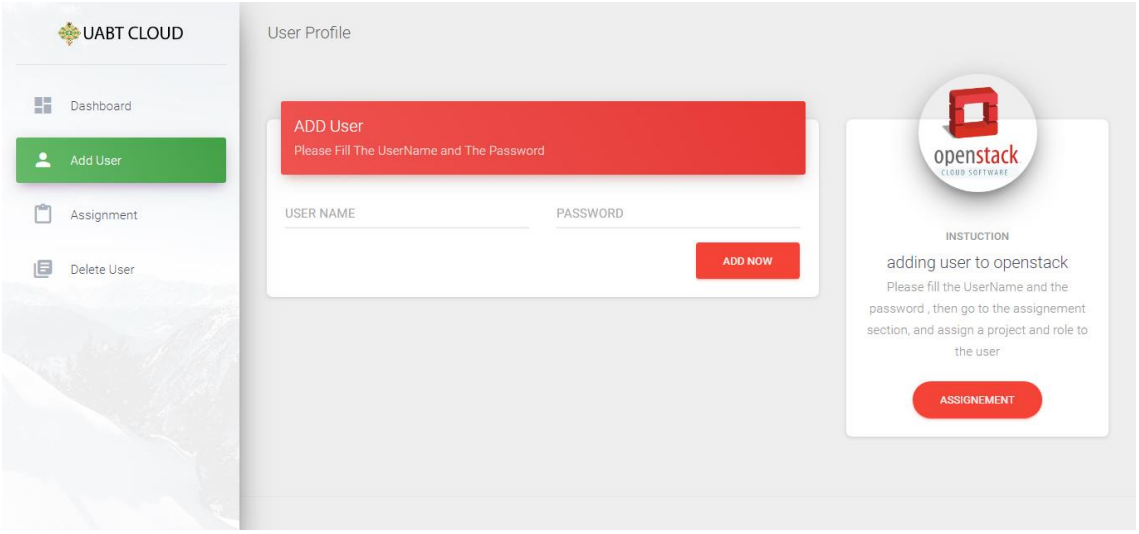

FIGURE 3.10 LDAP ADD USER

Assigner Role et Projet

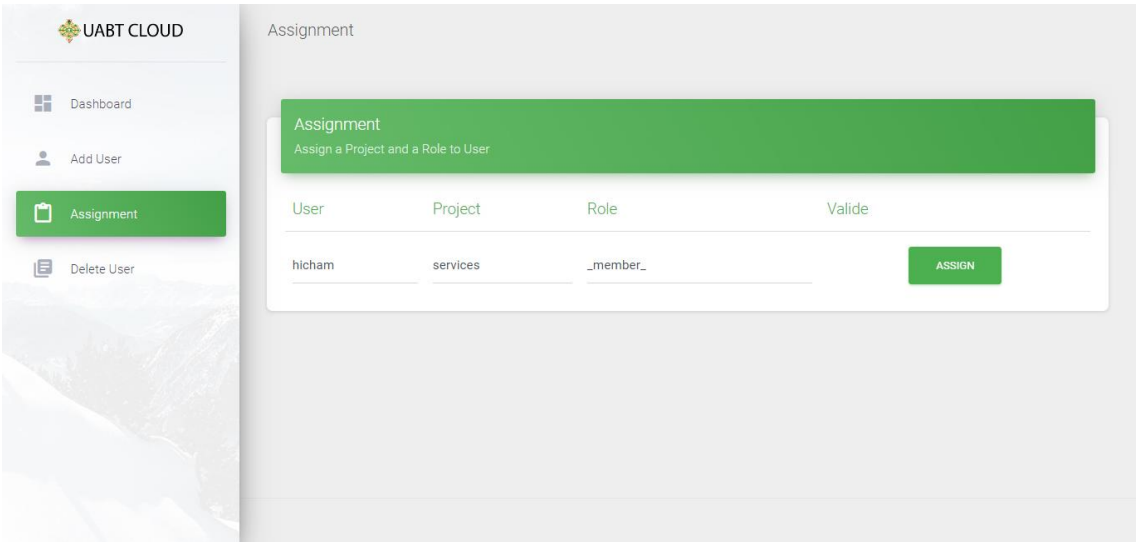

FIGURE 3.11 LDAP ASSIGNER ROLE ET PROJET

Supprimer

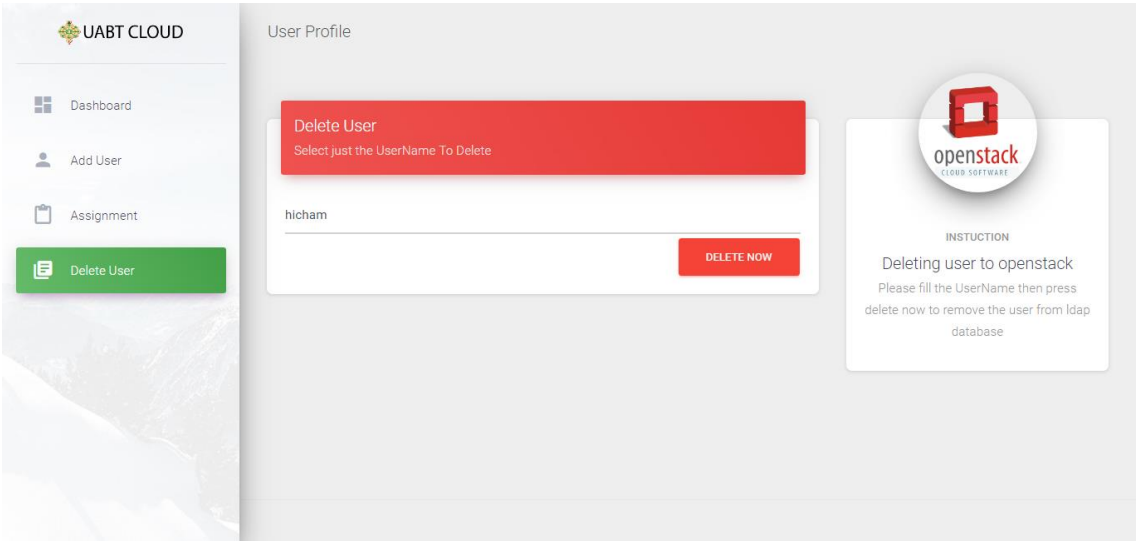

FIGURE 3.12 LDAP DELETE USER

Conclusion

Enfin, La principale contribution de cette intégration est d'étudier comment une entreprise ou un établissement peut introduire le Cloud Computing dans son système tout en garantissant la disponibilité du systeme. Nous nous sommes attaqués à une des limitations actuelles en remplaçant la base de donnée SQL existante dans openstack vers une nouvelle base LDAP fournit par l'université qui vise le changement. Afin d'éviter de modifier de manière importante cette nouvelle base, il était important de rendre les autres mécanismes d'Openstack compatibles avec cette base. Nous avons étudié comment fonctionne LDAP (Arbre Ldap) et comment Openstack peut communiquer et connaître toutes les branches. De cette manière, nous avons réussi à faire fonctionner le service Keystone (Identity) et l'obliger à extraire les données de LDAP, toute en respectant les règles dans l'authentification.

Conclusion général et perspective

Nous vivons une révolution des données, avec une dématérialisation d'un nombre croissant de processus et l'apparition de produit et service numérique totalement nouveau. La croissance continue des besoins en puissance des ressources, l'augmentation de manière exponentielle des quantités de donnée et la nécessité de l'automatisation, pour accélérer les processus métier et concentrer ses efforts sur des opérations à forte valeur ajoutée, a poussé différentes entreprises et établissements étatiques à réfléchir à introduire de nouvelles technologies dans leur système afin de garder leur croissance à l'échelle économique.

Le Cloud Computing apporte une réponse pragmatique aux entreprises pour leur permettre de profiter de ces évolutions technologiques et de satisfaire à une bonne partie de leur besoin tout en garantissant la stabilité, l'agilité et la flexibilité de leurs systèmes.

Nous Avons proposé dans ce projet de fin d'études une solution Cloud basée sur openstack, qui est sur le marché depuis plus de 9 ans. Il fournit l'une des plateformes opens-source les plus réussies pour déployer un Cloud. Le logiciel lui même a été progressivement adapté aux efforts croissants de développement de la communauté pour être plus stable et avoir plus de fonctionnalités afin de satisfaire la croissance des besoins des utilisateurs du Cloud.

Puisque l'objectif est de garantir le passage vers le Cloud, nous avons focalisé l'effort sur le service d'identité dans openstack nommé keystone, en réussissant à l'améliorer et l'enrichir avec une base d'authentification centralisée basée sur LDAP, rappelons que LDAP est un outil de gestion d'annuaire qui gère l authentification.

Le travaille effectué consiste à changer l'identification par défaut dans keystone, en replaçant le sql backend par un annuaire centralisé LDAP, et garantir a l'utilisateur l'accès à toutes les applications et les services dans le réseaux par le même compte enregistré dans l'annuaire LDAP.

Néanmoins les résultats de ce modeste travail constituent les bases d'un travail à poursuivre et à améliorer pour faire d'autres intégrations au futur sur d'autres services openstack tel que le stockage et le réseaux, et fournir un bon Cloud pour l'université.

Bibliographie

[7]Rob Zanella & Sumner Blount, Cloud Security and Governance: Who is on your cloud?, CA Technologies,2010

[8] Ronald L Krutz, Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing,Kindle,2010

[10] Scott Goessling, Kevin L. Jackson, Architecting Cloud Computing Solutions, Packt Publishing ,2018

[11] James Denton,Egle Sigler, CodyBunch, KevinJackson, Openstack Cloud Computing Cookbook - Fourth Edition, Packt Publishing,2018

[12] Michael J. Kavis,Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS), wiley,2014

[13] Pethuru Raj, Harihara Subramanian, Hands-On RESTful API Design Patterns and Best Practices, Packt Publishing,2019

[14] Permana Widhiasta, Harri Stranden, Michel Melot, Heinz Johner, LDAP Implementation Cookbook**,** RedBooks**,**1999

[5] Nutanix Training team, HOW NUTANIX WORKS, The Definitive Guide to Hyperconverged Infrastructure.

[15] Chunhui Yang, Michael Storrs, Sunil Ranahandola, Nathan Owen, Richard Macbeth, Jay Leiserson, Ramakrishna Gorthi, Ami Ehlenberger, Steven Tuttle, Understanding LDAP - Design and Implementation, IBM Redbooks, 2004

[16] Michael Solberg, Ben Silverman, Openstack for Architects - Second Edition, Packt Publishing,2018

[17] Alain Cardon and Mhamed Itmi, New Autonomous Systems,ISTE, 2016.

[18] Gustavo A. A. Santana, Data Center Virtualization Fundamentals, ciscopress, 2013

[19] John Wiley & Sons, Information Storage and Management, EMC Education Services, 2012

[1] Peter Mell (NIST), Tim Grance (NIST) La définition NIST du Cloud Computing. URL: https://csrc.nist.gov/publications/detail/sp/800-145/final, Recommendations of the National Institute of Standards and Technology, September 2011

[2]Arash Mahjani, Security Issues of Virtualization in Cloud Computing Environments Master of Arts (60 credits), Master of Science in Information Security, Luleå University of Technology, Department of Computer science, Electrical and Space engineering, 2015

[3]https://www.futura-sciences.com/tech/definitions/informatique-data-center-15675/.

[4]Pierre Sens, professeur à l'université Pierre et Marie Curie (Paris-VI), directeur-adjoint du LIP6, Universalis.

[6] Jean VAN DEN BROEK D'OBRENAN, ingénieur conseil,Universalis.

[9] https://docs.openstack.org/rocky/

[20] https://aws.amazon.com/fr/types-of-cloud-computing/

[21] https://www.slideshare.net/mirantis/openstack-architecture-43160012

ANNEXE 1 : ETAPE ET COMMANDE D'INSTALATION OPENSTACK (PACKSTACK)

#check your CentOS release

cat /etc/redhat-release

#populate your /etc/environment file with below locale settings

vi /etc/environment

LANG=en_US.utf-8

LC_ALL=en_US.utf-8

#if you are not familiar with vi editor, you can press "i" to start editing a file and press "esc" and

then ":wq" to save file and quit vi editor.

#check the status of firewalld service. Stop and disable it if enabled

systemctl status firewalld

systemctl stop firewalld

systemctl disable firewalld

#check the status of NetworkManager service. Stop and disable it if enabled

systemctl status NetworkManager

systemctl stop NetworkManager

systemctl disable NetworkManager

#enable and start network service

systemctl enable network

systemctl start network

#replace "enp0s3" with your interface name and check it's current settings

cat /etc/sysconfig/network-scripts/ifcfg-enp0s3

#disable selinux from it's config file /etc/selinux/config

vi /etc/selinux/config

SELINUX=disabled

#reboot your system

reboot

#after the reboot check the status of selinux, it should be disabled

getenforce

#On RHEL, download and install the RDO repository RPM(not required for centos)

sudo yum install -y https://rdoproject.org/repos/rdo-release.rpm

#On CentOS install the latest release of openstack package

```
sudo yum install -y centos-release-openstack-ocata
```

#this updates your current packages

```
sudo yum update -y
```

#install packstack installer

```
sudo yum install -y openstack-packstack
```

#to check the IP addresses on your machine

```
ip address show
```

#run the packstack installer with below parameters

```
packstack --allinone
```

#make sure your ethernet interface settings look like this, you should remove the IP address from

the interface

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=OVSPort
NAME=enp0s3
DEVICE=enp0s3
DEVICETYPE=ovs
OVS_BRIDGE=br-ex
ONBOOT=yes
```

#make sure your external bridge settings look like below

```
vi /etc/sysconfig/network-scripts/ifcfg-br-ex
DEVICE=br-ex
DEVICETYPE=ovs
TYPE=OVSBridge
BOOTPROTO=static
IPADDR=<your_IP>
NETMASK=<your_mask>
GATEWAY=<your_gateway_IP>
IPV4_FAILURE_FATAL=no
IPV6INIT=no
DNS1=<DNS_Server_IP>
ONBOOT=yes
```

#restart the network service

```
service network restart
```

#this command provides you the openstack admin privileges

```
source keystonerc_admin
```

#run this command to create your provider network for your instances so they can communicate

#with the outside world

```
neutron net-create external_network --provider:network_type flat --
provider:physical_network extnet --router:external
```

#this command creates the subnet attached to your provider network. You should be doing the

#configuration according to the LAN that your linux machine is connected to

```
neutron subnet-create --name public_subnet --enable_dhcp=False --allocation-pool
start=<IP_pool_first_address>,end=<IP_pool_last_address> --
gateway=<linux_gateway_IP> external_network <your_network_in_CIDR>
```

#example:

```
neutron subnet-create --name public_subnet --enable_dhcp=False --
allocation-pool start=192.168.1.100,end=192.168.1.120 --
gateway=192.168.1.1 external_network 192.168.1.0/24
```

ANNEXE 2 : INTEGRATION LDAP BACKEND AVEC KEYSTONE (IDENITY)

Install OpenLdap and Others Tools

Copy

```
yum install openldap* ldap* nss* db* -y
```

LDAP Configuration Steps:

1. We need to configure our basedn and add a password:

Copy

```
cd /etc/openldap/slapd.d/cn=config/
```

Copy

```
ls
```

Copy

```
slappasswd -h {CLEARTEXT}
```

Type the password as **MALTIBALI** to have consistency during the lab.

Example output:

```
New password: MALTIBALI

Re-enter new password: MALTIBALI
```

Copy

```
sed -i -e '/olcRootDN/ s/cn=Manager,dc=my-domain,dc=com/cn=Manager,dc=openstac
k,dc=org/' olcDatabase={2}hdb.ldif
```

b. Make sure olcSuffix is updated according to openstack dc.

Copy

```
sed -i -e '/olcSuffix/ s/dc=my-domain,dc=com/dc=openstack,dc=org/' olcDatabase={2}hdb.ldif
```

c. Now we need to add the password **“MALTIBALI”**, you may have to add the **olcRootPW** line at end of the file by running the below command,

Copy

```
sed -i -e '$ a olcRootPW: MALTIBALI ' olcDatabase={2}hdb.ldif
```

3. Specifying your dc, you might want to use in **olcAccess** parameter under: olcDatabase\=\{1\}monitor.ldif Modify **dc** that you want to use running the below command,

Copy

```
sed -i -e 's/cn=Manager,dc=my-domain,dc=com/cn=Manager,dc=openstack,dc=org/' olcDatabase\=\{1\}monitor.ldif
```

4. Enable and Start slapd service using the following command:

Copy

```
systemctl enable slapd

systemctl start slapd
```

5. The command to verify the configuration file as shown below. This should display “testing succeeded” message as shown below.

Copy

```
cd
```

Create an ldif file with all existing openstack user : exemple

Copy

```
cat << EOF >> ~/openstack.ldif

# nova, Users, openstack.org

dn: cn=nova,ou=Users,dc=openstack,dc=org

objectClass: person

objectClass: inetOrgPerson

sn: nova

cn: nova

userPassword: nova //find it on the answer file

EOF
```

Make changes to **[identity]** section in the same configuration file for changing the driver from **sql** to **ldap**

Copy

```
openstack-config --set /etc/keystone/keystone.conf identity driver ldap
```

c. Add entry to **[assignment]** section in the same configuration file to include driver for **sql**.

Copy

```
openstack-config --set /etc/keystone/keystone.conf assignment driver sql
```

Copy

```
cat <<EOF>> /etc/keystone/keystone.conf

[ldap]

url = ldap://localhost

user = cn=Manager,dc=openstack,dc=org

password = MALTIBALI

suffix = cn=openstack,cn=org

use_dumb_member = True

user_attribute_ignore = enabled,email,tenants,default_project_id

tree_dn = dc=openstack,dc=org

user_tree_dn = ou=Users,dc=openstack,dc=org

user_objectclass = inetOrgPerson

user_id_attribute = cn

user_name_attribute = sn

user_pass_attribute = userPassword

user_allow_create = True
```

```
user_allow_update = True


EOF
```

Copy

```
systemctl restart httpd.service
```

Copy

```
ldapsearch -x -b 'dc=openstack,dc=org' '(objectclass=*)'
```

Import schemas

Copy

```
ldapadd -Y EXTERNAL -H ldapi:/// -f/etc/openldap/schema/cosine.ldif

ldapadd -Y EXTERNAL -H ldapi:/// -f/etc/openldap/schema/inetorgperson.ldif

ldapadd -Y EXTERNAL -H ldapi:/// -f/etc/openldap/schema/nis.ldif
```

Import Base Structure:

a. Now we can import the base structure in to the LDAP directory using the ldapadd command as shown below.

Copy

```
ldapadd -x -D "cn=Manager, dc=openstack, dc=org" -W <~/nova.ldif
```

b. Restart keystone service

Copy

```
systemctl restart httpd.service
```

Copy

```
ldapsearch -x -b 'dc=openstack,dc=org' '(objectclass=*)'
```

Copy

```
source ~/keystonerc_admin
```

If the LDAP mappings are correct in keystone.conf, the “user list” command should show the list of users in the LDAP database.

Copy

```
openstack user list
```

Example Output:

```
+----------------------------------+---------+

| ID                               | Name    |

+----------------------------------+---------+

| c87f5bf84abe4bca8daa6e8283d179c6 | admin   |

| 4011e55571e14498b304d709b3d2f72a | nova    |

| 1906f67d3fb74997aaac3c88ea86a5cb | cinder  |

| 8648129611604feead716fc7eafdd3ec | glance  |

| 4fe832edba8b42f6a41b1bbe8a5c002b | heat    |

| 8d8b3c5ecf3148cf928049560454299c | neutron |

+----------------------------------+---------+
```

Résumé

Le changement rapide de la technologie et l'augmentation immense des données, obligent les moyennes et grandes entreprises à se mettre à jour avec la nouveauté informatique car un petit retard causera une grande perte de temps et par la suite des pertes financières difficiles à gérer. Une de ses nouveautés est le domaine du Cloud Computing qui vient de franchir les hauts niveaux de la gestion, automatisation et rapidité à résoudre différentes tâches, et qui représente une tendance consommée par les entreprises.

Notre travail consiste à faire une migration d'un simple système d'entreprise à un Cloud privé qui contiendra une bonne partie de l'ancien système mais avec un nouveau mécanisme. L'importante tâche dans ce travail est de centralisé l'identification tout en gardant tous les utilisateurs existants et avec un seul accès à toutes les applications et systèmes intégrés dans l'entreprise.

Abstract

The fast change of technologies and the immense increase of data, oblige the medium and big company to be upgraded with the novelty of computer science. Small delay may cause a great loss of time and then, a financial losses that are difficult to manage. One of those innovations is cloud computing that has just crossed the high level of management, automation and speed to solve different tasks, and which represents a trend consumed by the companies.

Our job is to migrate a simple enterprise system to a private cloud that will contain the main part of the old system but with a new mechanism. The important task in this work is to centralize the identification while keeping all existing users and with only one access to all applications and systems integrated into the company.

ملخص

التغيير السريع للتكنولوجيات والزيادة الهائلة في البيانات ، ألزم الشركات المتوسطة والكبيرة بالترقية مع حداثة التكنولوجيا ، لأن أي تأخير بسيط قد يؤدي إلى ضياع كبير للوقت ثم إلى خسائر مالية في وقت لاحق ومن الصعب تقبل ذلك ، أحد هذه الابتكارات هو الحوسبة السحابية التي تجاوزت مستوى عالٍ من الإدارة والسرعة و الحل الأوتوماتيكي للمشاكل المختلفة ، والتي تمثل اتجاهًا تستهلكه الشركات حاليا.

تتمثل مهمتنا في الانتقال من نظام مؤسسة بسيط إلى سحابة خاصة تحتوي على جزء جيد من النظام القديم ولكن مع وجود آلية جديدة ، تتمثل المهمة في هذا العمل في جعل عملية تحديد الهوية مركزية مع الاحتفاظ بجميع المستخدمين الحاليين مع تطبيق واحد فقط الوصول إلى جميع التطبيقات ونظام متكامل في الشركة.