

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de : Génie électrique et d'informatique
Département : Informatique



MÉMOIRE DE FIN D'ÉTUDES

En vue de l'obtention du diplôme de

Master en Informatique

Spécialité : Conduite de projets informatiques

Présenté par

YAKOUBI Abdelhak et **ZEKRINI Massinissa**

Thème : Etude et mise en place d'une solution
cloud privé avec Openstack

Proposé et dirigé par : Mme. AOUDJIT

Soutenu le 21/12/2020 devant le jury composé de :

Mme. BELKADI Présidente du Jury

Mme. BELATTAF Examinatrice

Mme. AOUDJIT Directrice du mémoire

Année universitaire : 2019/2020

Remerciements

A l'issue de ce travail, Nous tenons d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et le courage de le mener à terme.

Nous tenons à exprimer nos plus grands remerciements à nos très chers parents pour leur soutien moral et leurs encouragements. Que dieux nous les garde et les protège.

*À notre Encadreur, **Madame AOUDJIT** pour son encadrement, ses remarques pertinentes, son soutiens et sa motivation, nous la remercions infiniment de nous avoir fait confiance et d'avoir autant cru en nous et pour chaque petit détail qu'il nous a appris tout au long de notre cursus.*

Nous tenons à exprimer nos sincères remerciements à tous les enseignants qui nous ont transmis leur savoir et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Nous tenons à remercier les membres de jury qui ont accepté d'évaluer notre travail.

Enfin, Nous remercions toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Abdelhak.Y && Massinissa.Z

Dédicaces

Je dédie ce travail,

Á ma très chère mère pour son sacrifice et son soutien moral.

Á mes chères sœurs et chers frères.

*Á mes chères amies Nesrine.H et Yasmine.B pour leur soutien et
encouragement tout au long de mon parcours.*

*Á mon binôme et cher ami **Massinissa.Z** qui a été complémentaire avec
moi durant tout notre cursus universitaire.*

*Á la mémoire de notre cher défunt enseignant **Mr. HEMDANI**
Chabane ; paix à son âme.*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit
de votre soutien infaillible. Merci d'être toujours là pour moi.*

Abdelhak.Y

Dédicaces

Je dédie ce travail,

A mes très chers Parents,

A mon cher frère et mes chères sœurs,

A tout ma famille,

*A mon binôme et cher ami **Abdelhak. Y**,*

A mes chers amis je ne saurais tous les citer.

Massinissa.Z

Table des matières

Introduction	1
I Généralités du cloud computing	3
I.1 Introduction	4
I.2 Définitions et généralités	4
I.2.1 Définitions	4
I.2.2 Les caractéristiques essentielles du cloud computing . .	6
I.2.3 Le cloud computing en chiffres[1]	7
I.2.4 Le cloud computing en Algérie	7
I.3 Historique du cloud computing	8
I.4 Avantages et limites du cloud computing	10
I.4.1 Les avantages du cloud computing	10
I.4.2 Les limites du cloud computing	11
I.5 Les différents services du cloud computing	12
I.5.1 Infrastructure as a service « IAAS »	13
I.5.2 Platform as a service « PAAS »	13
I.5.3 Software as a service « SAAS »	14
I.6 Modèles de déploiement du cloud computing	15
I.6.1 Le nuage privé	15
I.6.2 Le nuage communautaire	16
I.6.3 Le nuage public	16
I.6.4 Le nuage Hybride	17
I.7 Classification du cloud computing	17
I.7.1 Cloud d’Entreprise	18
I.7.2 Cloud gouvernemental et recherche scientifique	18

TABLE DES MATIÈRES

I.7.3	Cloud pour Réseaux Sociaux et Jeux	18
I.7.4	Cloud pour Fournisseurs de Services	19
I.8	La virtualisation[2] [3]	19
I.8.1	Les types de virtualisation	20
I.8.2	Solutions de virtualisation	22
I.9	Les Hyperviseurs	23
I.9.1	Les hyperviseurs de type 1	23
I.9.2	Les hyperviseurs de type 2	23
I.10	Les Acteurs de cloud computing	24
I.11	Les vulnérabilités dans la sécurité dans du cloud	30
I.11.1	La perte de maîtrise de gouvernance	30
I.11.2	Isolement des environnements et des données	30
I.11.3	Risques de conformité	31
I.11.4	La publication des interfaces de gestion	31
I.11.5	La protection des données	31
I.11.6	La suppression dangereuse ou incomplète des données	31
I.11.7	Les utilisateurs malveillants	32
I.12	Conclusion	32
II	Plateformes de cloud computing	33
II.1	Introduction	34
II.2	Plateformes de cloud computing	34
II.2.1	Les solutions propriétaires	34
II.2.2	Les solutions open source	37
II.3	Identification des besoins	40
II.3.1	Besoin Fonctionnels	40
II.3.2	Besoins non fonctionnels	41
II.3.3	Solution proposée	42
II.4	Présentation d'openstack	42
II.4.1	Définition	42
II.4.2	Historique	43
II.5	Les composants d'Openstack	44
II.5.1	Services de base[4]	44

TABLE DES MATIÈRES

II.5.2	Services partagés	50
II.5.3	Autres services (les services additionnels)	54
II.6	Architecture d’Openstack	55
II.6.1	Architecture logique	55
II.6.2	Architecture conceptuelle	56
II.6.3	Architecture technique	57
II.7	Avantages et Inconvénients	57
II.7.1	Avantages d’openstack	57
II.7.2	Inconvénients d’openstack	58
II.8	Conclusion	59
III	La mise en place de plateforme Openstack	60
III.1	Introduction	61
III.2	La mise en place de la plateforme Openstack	61
III.2.1	Types d’installation d’Openstack	61
III.2.2	Les prérequis	63
III.2.3	Outils	63
III.3	Installation des paquets de base et préparation du système	65
III.3.1	Configuration de la carte réseau	65
III.3.2	Mise à jour du système	66
III.3.3	Installation de l’outil de base de données MySQL « mariadb »	67
III.3.4	Gestion de fil d’attente, installation du service RabbitMQ	68
III.3.5	Installer le service d’identité memcached	69
III.3.6	Installation du service ETCD	70
III.3.7	Installation du protocole NTP « chrony »	71
III.3.8	Installer openstack avec la méthode single node (microstack)	72
III.4	Les étapes de configuration du serveur	79
III.4.1	Exemple concret	79
III.4.2	Déroulement de travail sur la plateforme	80
III.5	Migration de machines virtuelles	123
III.5.1	Migration à froid	123

TABLE DES MATIÈRES

III.5.2 Migration à chaud	125
III.6 Conclusion	127
Conclusion générale	128
Bibliographie	131

Table des figures

I.1	L'évolution d'internet au fil du temps.	9
I.2	Les différents avantages du cloud computing.	10
I.3	Les services de cloud computing.	13
I.4	Répartition des charges et comparaison avec le modèle classique.	14
I.5	Les deux types d'hyperviseurs.	24
I.6	Logo du salesForce.	25
I.7	Logo du amazon.	26
I.8	Logo Google App-Engine.	27
I.9	Logo Microsoft - Windows Azure.	29
II.1	Logo du Amazon EC2.	35
II.2	Logo du Alibaba Cloud.	36
II.3	Logo du IBM Cloud.	37
II.4	Logo du Eucalyptus.	38
II.5	Logo du OpenNebula.	38
II.6	Logo du Openstack.	39
II.7	Architecture du composant Neutron.	47
II.8	Architecture du composant Cinder.	50
II.9	Architecture du composant Keystone.	51
II.10	Architecture du composant Glance.	53
II.11	Architecture logique d'openstack.	56
II.12	Architecture conceptuelle d'openstack.	57
III.1	Les différentes architectures possibles.	62
III.2	Logo du virtualBox.	63
III.3	Logo du Ubuntu.	64
III.4	Logo du OpenStack Choisi.	64

TABLE DES FIGURES

III.5	Tableau de bord OpenStack.	77
III.6	Première vue du Dashboard.	78
III.7	Création du rôle "membre".	80
III.8	Création du rôle "lecteur".	81
III.9	Création de l'utilisateur "secrétaire".	81
III.10	Création de l'utilisateur "agent".	82
III.11	Création de l'utilisateur "analyste1".	82
III.12	Création de l'utilisateur "analyste2".	83
III.13	Création du rôle : membre.	83
III.14	Création du rôle : lecteur.	84
III.15	Création de l'utilisateur : secrétaire.	84
III.16	Création de l'utilisateur : agent d'administration.	85
III.17	Création de l'utilisateur : analyste1.	85
III.18	Création de l'utilisateur : analyste2.	85
III.19	Attribution des rôles aux utilisateurs et au projet "admin".	86
III.20	Création du réseau public.	87
III.21	Création du sous/réseau.	87
III.22	Création du réseau privé.	88
III.23	Création du sous/réseau.	88
III.24	Création du routeur.	89
III.25	Ajout d'interface dans le routeur.	89
III.26	Vue globale de l'architecture réseau de l'entreprise.	90
III.27	Allocation d'une @IP flottante à un pool externe.	91
III.28	Confirmation de l'ajout.	91
III.29	Création de clé SSH.	92
III.30	Téléchargement de la clé.	92
III.31	Création de groupe de sécurité "techteam".	93
III.32	Confirmation de création de groupe.	93
III.33	Espace de modification et ajout de règles.	94
III.34	Ajout d'une règle au groupe.	94
III.35	Création de groupe de sécurité "adminteam".	95
III.36	Confirmation de création de groupe.	95
III.37	Espace de modification et ajout de règles.	96

TABLE DES FIGURES

III.38	Ajout d'une règle au groupe.	96
III.39	Création de flavor.	97
III.40	Attribution du flavor au projet "admin".	97
III.41	Confirmation de la création.	98
III.42	Création d'image système.	99
III.43	Confirmation de la création d'image.	99
III.44	Création d'instance.	100
III.45	Attribution d'une image système.	101
III.46	Attribution d'un gabarit.	101
III.47	Attribution des réseaux à l'instance.	102
III.48	Attribution de groupes de sécurité.	102
III.49	Attribution de la clé RSA.	103
III.50	construction de l'instance.	103
III.51	Création de l'instance avec succès.	104
III.52	Configuration de l'instance.	104
III.53	Association de l'@ flottante à l'instance.	105
III.54	Association de l'adresse flottante avec succès.	105
III.55	Architecture globale de l'instance reliée aux deux réseaux.	106
III.56	Passage au mode console.	106
III.57	L'instance en mode console.	107
III.58	Vérification de l'adresse IP de l'instance.	108
III.59	Création du réseau public.	109
III.60	Création du sous/réseau.	110
III.61	Création du réseau privé.	110
III.62	Création du sous/réseau.	111
III.63	Création du routeur : "myrouter".	112
III.64	Génération de clé SSH "mykey".	113
III.65	Importation de la clé "mykey".	115
III.66	Création du groupe de sécurité "techteam".	115
III.67	Création du groupe de sécurité "adminteam".	116
III.68	Création d'une règle dans le groupe de sécurité "techteam".	116
III.69	Création d'une règle dans le groupe de sécurité "adminteam".	117
III.70	Création d'un gabarit (flavor).	118

TABLE DES FIGURES

III.71	Création de l'image "myimage" avec succès.	119
III.72	Aperçu des réseaux.	120
III.73	Commande de création de l'instance.	120
III.74	Création de l'instance avec succès.	121
III.75	Aperçu des instances.	121
III.76	Récupération de l'adresse IP flottante : @192.168.0.164.. . . .	122
III.77	Vérification d'adressage IP de l'instance.	122
III.78	Liste des instances.	123
III.79	Aperçu d'informations de l'instance à migrer.	124
III.80	Commande de la migration à froid.	124
III.81	Confirmation de déplacement de l'instance.	125
III.82	Liste des instances et options possibles.	126
III.83	Choix de l'hôte courant et l'hôte destinataire.	127



Liste des tableaux

III.1 Tableau des prérequis.	63
III.2 Tableau des privilèges.	80

Introduction générale

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

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

Depuis l'apparition de cette technologie, et les bénéfices qu'elle apporte les entreprises migrent à grand pas vers le cloud donne la naissance d'un nombre important de plateformes open source comme Eucalyptus, OpenNebula et Openstack, et propriétaires comme Alibaba Cloud, amazon EC2 et IBM Cloud pour la mise en place de cette technologie.

Le but de ce travail est d'approfondir et d'expérimenter nos connaissances sur ce thème de Cloud Computing, puis de faire son état de l'art, en vue de choisir la meilleure solution disponible, de la déployer et l'évaluer. Pour se faire nous avons déployé un Cloud privé de type infrastructure en tant que service (IaaS).

Ce mémoire est organisé en 3 chapitres :

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

Le chapitre -2- est consacré à l'étude de quelques plateformes de cloud computing mises sur le marché, identifier nos besoins et définir la solution choisie.

Le chapitre -3-, nous allons nous pencher sur l'installation de la solution de Cloud Computing privé choisie, nous allons décrire les différentes étapes à suivre pour la bonne installation et configuration de cette dernière.

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

Généralités du cloud computing

I.1 Introduction

Le cloud computing est une nouvelle technologie qui permet aux entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosses quantités d'informations.

Actuellement, le cloud computing représente une évolution dans le monde informatique. En effet, le cloud computing est émergé dans les dernières années comme une solution universelle utilisée par différents types d'utilisateurs.

Face à cette tendance, nous allons présenter dans ce chapitre quelques généralités sur le cloud computing, à savoir sa définition, ses caractéristiques, les services qui l'offre, ses éléments constitutifs, ses modèles de déploiement ainsi que certaines notions pour la bonne compréhension de ce concept et enfin on termine avec ses avantages et inconvénients.

I.2 Définitions et généralités

I.2.1 Définitions

L'informatique dans le nuage est plus connu sous sa forme anglo-saxonne : « Cloud Computing », mais il existe bien d'autres synonymes francophones tels que : « info_nuagique » ou encore « informatique dématérialisé ».

Même si aujourd'hui les experts ne peuvent pas attribuer une définition exacte à ce concept, mais la majorité s'accordent à dire qu'elle inclue la notion de services disponibles à la demande extensibles à volonté. Contrairement aux systèmes actuels, les services sont virtuels et illimités et les détails des infrastructures physiques sur lesquels les applications reposent ne sont pas connus de l'utilisateur c'est pourquoi on parle d'informatique dématérialisé.

Nous prenons quelques définitions qui ont circulés :

- **NIST (National Institute of Standards and Technology) :**

«Le cloud computing est un modèle qui permet un accès omniprésent, pratique et à la demande à un réseau partagé et à un ensemble de ressources informatiques configurables (telles que : réseaux, serveurs, outils de stockage, applications et services) qui peuvent être rapidement mises à la disposition du client avec un minimum de gestion et sans interaction direct avec le fournisseur de services».[5]

- **CISCO :**

«Le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité de ressources»[6].

Les points essentiels à retenir de ces définitions sont les notions de « scalability¹ », d'extensibilité à la demande, d'élasticité, c'est à dire qu'on ne paie que ce qu'on utilise «Pay-as-you-go». C'est un avantage considérable par rapport à une infrastructure propre à l'entreprise où les serveurs sont très souvent sous-utilisés, cela permet aux start-up de lancer un service sans aucun investissement capitalistique en hardware.

Donc pour simplifier le Cloud Computing n'est pas un ensemble de technologies mais un modèle de fourniture de gestion et de consommation de services informatiques notamment des outils de stockage, de base de données, de serveurs, gestion réseaux, logiciels, applications d'analyse et intelligence artificielle via un nuage (internet), ces services à la demande localisés dans des Datacenter sont accessibles depuis n'importe où et à chaque instant, dans l'objectif d'offrir une innovation plus rapide, flexible et économique.

1. « Scalability » est la capacité à s'adapter à un changement d'ordre de grandeur de la demande.

I.2.2 Les caractéristiques essentielles du cloud computing

- **Self-service à la demande**

La mise en œuvre des systèmes est entièrement automatisée et c'est à l'utilisateur de mettre en place et gérer la configuration à distance en fonction de ses besoins, au moyen d'une console de commande.

- **Accès réseau large bande**

Les centres de traitement des fournisseurs du cloud sont généralement raccordés directement sur le backbone Internet pour bénéficier d'une excellente connectivité. Cela permet l'accès aux ressources via le réseau par des systèmes hétérogènes en moins de 50 ms de n'importe quel endroit sur la planète.

- **Réservoir de ressources (non localisées)**

Les prestataires de service de cloud offrent des dizaines de milliers de serveurs et de moyens de stockage via des Datacenter équipés de hautes technologies pour permettre la disponibilité de ressources et des montées en charge rapide. Il est souvent possible de choisir des zones géographiques près des utilisateurs.

- **Elasticité (redimensionnement rapide)**

La mise en ligne d'une nouvelle instance d'un serveur est réalisée en quelques minutes, l'arrêt et le redémarrage en quelques secondes. Toutes ces opérations peuvent s'effectuer automatiquement par des scripts. Ces mécanismes de gestion permettent de bénéficier pleinement de la facturation à l'usage en adaptant la puissance de calcul au trafic instantané, Ainsi, un client peut louer un serveur sur une courte période et à la taille voulue.

- **Service mesurable et facturable (pay-as-you-use)**

La facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Il n'y a généralement pas de coût de mise en service (c'est l'utilisateur lui même qui réalise les opérations).

I.2.3 Le cloud computing en chiffres[1]

- 58% des entreprises mondiales utilisent le stockage cloud.
- 75% des entreprises favorisent le cloud hybride pour des raisons économiques.
- 95% des entreprises ont migré certaines de leurs applications dans le cloud.
- Plus de 383 Md\$ de chiffre d'affaires en 2020 contre 3Md\$ en 2008.
- Seulement 27% des entreprises estiment que leurs données sont sécurisées.
- Seulement 2% des entreprises américaines n'ont aucune application dans le cloud.

I.2.4 Le cloud computing en Algérie

L'arrivée du cloud en Algérie s'est avérée commencer en 2013 par le déploiement du premier prestataire national l'ISSAL² basé à Oran mais il fallait attendre jusqu'à janvier 2018 que l'Algérie régularise le cloud computing au même titre que la géo-localisation en publiant un cahier des charges élaboré par l'ARPCE³ comportant des conditions et des modalités d'établissement et d'exploitation de services d'hébergement difficile à réaliser selon les experts comme l'exigence d'implantation des Datacenter sur le territoire national ainsi que des mesures de sécurité compliquées à suivre selon les fournisseurs qui ont été prêts à s'investir dans ce nouveau défi en Algérie, qui jugent que certaines clauses du cahier des charges sont inatteignables.

Il est également important de citer que l'Algérie possède actuellement trois data centers, dont celui de l'ISSAL, CERIST⁴ rattaché au ministère de l'Enseignement supérieur et de la recherche scientifique ainsi que le fameux centre de traitement de données d'EBS⁵ de Sidi_Abdellah à Alger qui abrite plus de 35 jeunes entreprises.

2. ISSAL : Partenaire entreprise de Google en Algérie.

3. ARPCE : Autorité de Régulation de la Poste et des Communications Electroniques.

4. CERIST : Centre de recherche sur l'information scientifique et technique.

5. EBS : Electronic Business Services.

Bien que des dizaines d'entreprises commencent à franchir le pas et d'autres qui pensent sérieusement intégrer dans un avenir proche mais le chemin du cloud computing dans notre pays semble très long.

I.3 Historique du cloud computing

Dans les années 1950; longtemps avant la naissance de l'expression « cloud computing », ses fondements existent sous forme de transfert de fichiers et de bases de données sur des ordinateurs centraux dotés de puissance de calcul et d'espace de stockage considérables appelés mainframes (ancêtres des serveurs cloud) conservés localement dans les structures des entreprises ou des universités, à cette époque les utilisateurs étudiant soit il ou employé de l'entreprise est autorisé à partager les capacités des systèmes centraux via leurs propres machines pour bénéficier de ses performances dans leurs calculs. Les ingénieurs réseaux schématisaient donc son réseautage sous forme d'un nuage (en anglais : cloud)[7].

L'idée du cloud computing a fait surface aux débuts des années 1990, surtout après la naissance de nombreuses technologies telles internet en 1991, les navigateurs web MOSAIC en 1993 et Netscape en 1994, ainsi que la fondation d'EBay et d'Amazon pour la vente en ligne en 1995 qui a fait accélérer le processus de développement du cloud d'une manière indirecte et spontanée avec leur gourmandise et nécessité de stockage d'informations vu que ces derniers vise un grand nombre de clients qui s'accroît du jour en jour.

Au début des années 2000, le cloud computing est né concrètement grâce aux investigations menées par Amazon web services, une librairie en ligne de vente de toutes sortes de fournitures. La société leader de e-commerce a investi dans un immense parc d'infrastructures pour satisfaire des millions de ses clients durant les périodes de pointe telle que Noël, puis de louer ses ressources inexploitées à d'autres utilisateurs (particuliers, entreprises et administrations) durant le reste de l'année. L'affluence croissante des clients a poussé l'entreprise à effectuer des extensions de ses parcs pour satisfaire la demande ce qui a donné l'idée à d'autres sociétés qui ont fait de même en fournissant des services identiques.

Plusieurs facteurs propices ont conduit à l'essor du cloud computing et surtout à sa rapidité de développement :

- L'externalisation⁶ à la fin des années 90s (comme le stockage de données hors des murs de l'entreprise).
- L'augmentation des débits des réseaux (grâce à L'ADSL puis à la fibre optique).
- La diminution des coûts des capacités de traitement et de stockage (progrès technique en matière de fabrication de disques durs).
- La virtualisation (concept plus ancien constitue le socle du cloud computing).
- Le développement massif de l'informatique mobile et ubiquitaire⁷.

C'est à partir de 2013 que les entreprises se dirigent massivement vers le cloud computing afin de bénéficier de toute sorte de services et de technologies offerts par les prestataires du marché et réaliser également des gains considérables de temps, de ressources et de chiffre d'affaires.

L'évolution d'Internet

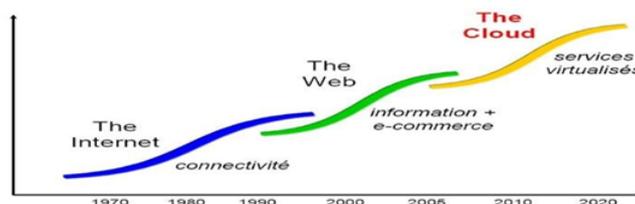


FIGURE I.1 – L'évolution d'internet au fil du temps.

Aujourd'hui la pénétration du cloud computing dans le monde est énorme dans presque tous les domaines allant des jeux vidéo, cinéma, musique et réseaux sociaux passant par les PME et grandes entreprises de numériques

6. L'externalisation est le transfert de tout ou partie d'une fonction d'une organisation vers un partenaire externe.

7. Ubiquitaire désigne le fait que l'informatique est omniprésente.

et de téléphonie et arrivant aux petites utilisations de particuliers à travers la toile d'araignée et cela ne fait que commencer.

I.4 Avantages et limites du cloud computing

Le cloud computing est un modèle forfaitaire à la fois rentable pour l'entreprise, l'utilisateur ainsi que le fournisseur, ce dernier tire profit du fait de la réutilisation de ses mêmes ressources facturées à chaque usage tandis que l'entreprise lui permet de se concentrer sur ces principales fonctions de production hors maintenance de son système d'information ou stockage de ses bases de données. L'utilisateur lui, fait appel aux services cloud SaaS (applications tant que service) sans se rendre compte à travers ses utilisations de messagerie électronique et de réseaux sociaux via internet.

Malgré tous ces points forts, le cloud possède également une petite marge de risque et de limites à admettre ou à développer au futur.

I.4.1 Les avantages du cloud computing



FIGURE I.2 – Les différents avantages du cloud computing.

1. Flexibilité idéale et accès optimisé aux ressources

Selon l'évolution de ses besoins, l'entreprise peut augmenter ou réduire ses ressources dans un délai réduit sans se soucier de l'obtention des

équipements grâce à l'allocation dynamique, ainsi de pouvoir accéder aux multiples logiciels et applications proposées par le fournisseur de services dont elle ne pourrait s'offrir en interne.

2. Optimisation des coûts

Les services de cloud permettent à une entreprise de démarrer une activité professionnelle sans avoir à investir dans l'infrastructure IT, qui nécessite également une gestion interne plus coûteuse entre achat de matériels, affectation du personnel, locaux, électricité, maintenance et renouvellement des équipements. Autre point intéressant est la facturation à l'usage, ce que signifie que l'utilisateur paie uniquement ce qu'il consomme.

Et pour avoir une idée claire du gain réalisé, on reprend l'observation du Michael Crandell à propos du cloud d'Amazon qui dit qu'un serveur et ses charges chez ce fournisseur coûtent entre 70\$ à 150\$ contre 400\$ s'il était hébergé par l'entreprise en interne.

3. Une diminution des pertes de données accidentelles

Grace à la réplication de données qui se fait précieusement et en temps réel et qui permet donc aux utilisateurs de récupérer les informations cruciales de leur établissement.

4. Facilité d'utilisation, disponibilité, évolutivité

Ces qualités sont réalisées grâce aux hautes technologies de pointes utilisées par les fournisseurs de service telles que les Data centers, les supercalculateurs, les connexions internet haut débit et une puissance de calcul que les entreprises peuvent pas s'offrir en interne.

I.4.2 Les limites du cloud computing

1. Confidentialité et sécurité des données

C'est la limite majeure de la solution cloud, les données sont hébergées en dehors de l'entreprise et la possibilité de voir ses données volées ou mal utilisées reste inquiétante même si elle est plus tôt rare mais l'absence d'un service pendant quelques minutes ou l'accès aux données confidentielles par les hackers peut engendrer des pertes considérables.

De ce fait le seul moyen qui rassure le client est le contrat de garantie signé avec le fournisseur.

2. Mauvaise dépendance

Difficile de convaincre le fournisseur si l'entreprise souhaite se lancer dans une activité secrète et personnalisée ou encore de développer des fonctionnalités très spécifiques, cela dépend donc du meilleur choix de prestataire cloud qui se base sur la confiance.

I.5 Les différents services du cloud computing

Le cloud computing est composé de trois couches :

- Application (SaaS, Software as a Service).
- Plateforme (PaaS, Platform as a Service).
- Infrastructure (IaaS, Infrastructure as a Service).

La figure suivante représente les différentes couches de cloud allant de la moins visible à l'utilisateur final à la plus visible schématisée sous forme de pyramide. L'infrastructure en tant que service «IAAS» est géré par les architectes réseaux, La plateforme en tant que service «PAAS» destinée aux développeurs d'applications et finalement le logiciel ou application en tant que service «SAAS» est le résultat final vu et exploité par l'utilisateur.

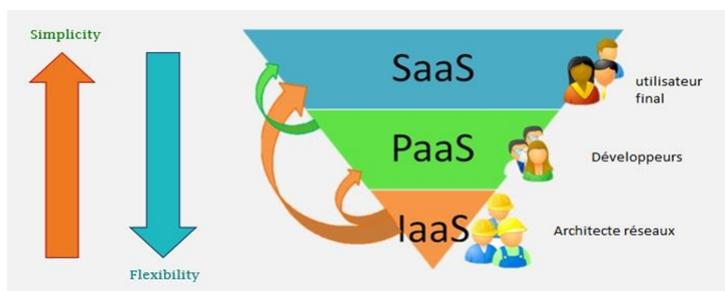


FIGURE I.3 – Les services de cloud computing.

I.5.1 Infrastructure as a service « IAAS »

C'est le service de plus bas niveau, où seul le serveur est dématérialisé. L'utilisateur peut avoir accès à plusieurs technologies proposées par le fournisseur situées dans les data centers hautement sécurisés (serveurs, machines virtuelles, bande passante et même des systèmes d'exploitation). Malgré que la gestion reste flexible et indépendante vis-à-vis des données hébergées mais cette solution oblige l'entreprise à affecter un administrateur système en interne tout comme dans les systèmes classiques. Parmi les meilleurs prestataires de cette catégorie de services on trouve : Amazon avec EC2 ou bien Orange business services avec Flexible Computing.

I.5.2 Platform as a service « PAAS »

Destinée aux équipes de développement. Le fournisseur offre une plateforme qui permet de créer, développer, tester, exécuter et gérer plus rapidement des applications web ou mobiles sans se soucier de la configuration d'hébergement ou de stockage et cela grâce aux composants offerts tels que la prise en charge des langages de programmation, kits de développement logiciels (SDK) ainsi que des systèmes de gestion de base de données.[8]

Les principaux fournisseurs de la catégorie PaaS sont : Microsoft avec AZURE, Google avec Google App Engine.

I.5.3 Software as a service « SAAS »

Les solutions SaaS constituent la forme la plus répandue de Cloud Computing, c'est une Méthode de diffusion d'applications mise à la disposition des consommateurs sur internet généralement sous forme d'abonnement et avec une facturation à l'usage réel. L'utilisateur se connecte via internet par un navigateur web ou installe directement l'application de façon locative sur son PC, téléphone ou tablette sans l'obligation d'effectuer les mises à jour ni les migrations de données qui sont faites par le fournisseur qui héberge et gère la maintenance et la sécurité de ses applications logicielles. Les prestataires de solutions SaaS les plus connus sont Microsoft qui offre Office365 et Google avec Google Apps.[8]

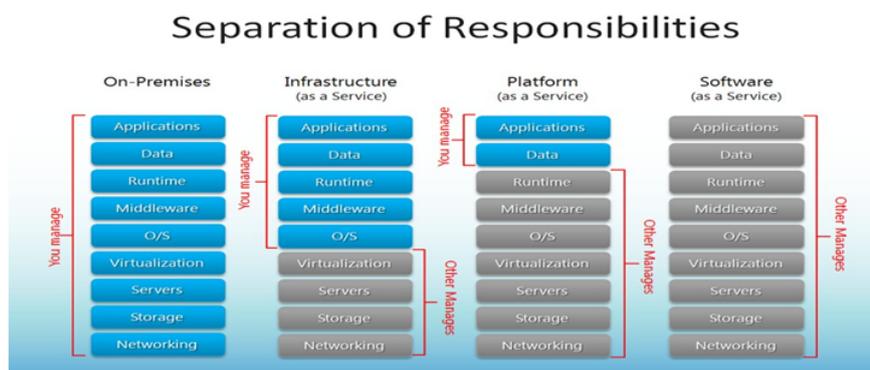


FIGURE I.4 – Répartition des charges et comparaison avec le modèle classique.

En dehors de ces trois principaux services, d'autres sont également disponibles mais moins utilisés à l'image de :

- **Data as a service** : C'est la mise à disposition de données délocalisées.
- **Business process as a service (BPaaS)** : il s'agit de l'externalisation d'un processus d'entreprise.

- **Desktop as a service (DaaS)** : bureau virtuel hébergé.
- **Network as a service (NaaS)** : fournit les services de réseautage.
- **Storage as a service (STaaS)** : stockage de fichiers chez des prestataires externes.
- **Communication as a service (CaaS)** : offre principalement des solutions de communications à distance et des ressources partagées.

I.6 Modèles de déploiement du cloud computing

Aujourd'hui, En vue d'améliorer leurs capacités de traitement et d'accélérer les processus de production et de commercialisation, les entreprises ont adopté le cloud grâce aux ressources informatiques souples et évolutives fournies selon les besoins professionnels de l'utilisateur mais à chaque organisation ses exigences en matière de cout, de disponibilité et surtout de confidentialité; c'est pour cela qu'il existe une multitude de modèles de déploiement dont on peut distinguer quatre types principaux présentés comme suit :

I.6.1 Le nuage privé

Ce type de cloud offre une architecture propriétaire et davantage de sécurité et de contrôle maximum de données puisque ces dernières sont hébergées soit :

- (a) **En interne** : c'est ce qu'on appelle le cloud privé internalisé, l'entreprise possède et gère entièrement les infrastructures de calcul et de stockage ainsi de pouvoir personnaliser ses applications sans faire appel aux tiers.
 - (b) **En externe** : C'est le cloud privé externalisé, qui offre les mêmes privilèges qu'en interne mais avec le bénéfice de la gestion du matériel par des équipes expertes en la matière travaillant chez un
-

fournisseur ce qui permet de diminuer la charge sur le personnel de l'entreprise qui peuvent également accéder à leurs données via des réseaux sécurisés de type VPN.

Mais dans les deux cas, l'entreprise s'assure que les ressources allouées ne seront pas partagées avec d'autres clients ce qui renforce un élément primordial dans l'économie actuelle et surtout pour les grandes entreprises mondiales ou chez les institutions gouvernementales qui est : la sécurité des données.

I.6.2 Le nuage communautaire

Plus récent mais rarement utilisé qui réserve un avenir meilleur, il s'agit donc d'un cloud utilisé par plusieurs entreprises ou organismes publics qui ont les mêmes besoins et exigences, son intérêt est de créer une structure partagée pour une collaboration commune généralement dans le même domaine de travail à titre d'exemple, le projet open cirrus qui regroupe plusieurs universités françaises au profit de la recherche scientifique ou encore Amadeus fournisseur de solutions informatiques à l'industrie du tourisme et du voyage partagé par les compagnies aériennes Air France, Lufthansa, Iberia et SAS dans le but de proposer à leurs clients des services via un seul et unique logiciel.

I.6.3 Le nuage public

Ce type de cloud très utilisé offre des services au grand public via une interface à libre service gratuits ou payants à l'usage, ces services proposés peuvent être des capacités de stockage pour les utilisateurs individuels en grande partie ainsi que des applications, outils réseaux et tests machines virtuelles et systèmes d'exploitations virtualisés à partir du matériel détenu par le fournisseur, regroupés dans des « data_lakes⁸ », orchestrés au moyen de logiciels de gestion et d'automatisation puis enfin soumises aux client via internet ou des connexions réseaux dédiés.

8. Data_lakes : méthode récente de stockage de données massives et volumineuses.

Cette méthode peut permet d'évoluer les charges de travail en cas de demande imprévue et les données des clients sont hébergées sur une multitude de serveurs accessibles par un grand nombre d'utilisateurs mais la sécurité des données est gérée par des équipes spécialisées de manière permanente contre tout accès illégal par d'autres locataires du cloud.

Les principaux leaders de cloud public sont : Amazon Web Services, Google Cloud(Compute Engine), Microsoft Azure ainsi qu'Alibaba Cloud.

I.6.4 Le nuage Hybride

Est un environnement combinant deux clouds ou plus généralement privé et public, tout en gardant l'indépendance de chacun à part entière et garantissant l'orchestration entre les deux plateformes grâce à de logiciels de gestion propriétaires, ce type de cloud permet de répartir les charges de travail ainsi de remédier à de nombreux obstacles que les entreprises trouvent troublant pour la migration de leurs données vers un unique cloud.

De nombreuses entreprises travaillent avec des workloads⁹ sensibles et critiques ce qui ne leur permet pas de délocaliser leur système d'information ainsi que d'autres institutions se débrouillent parfaitement sans cloud depuis des décennies et possèdent donc un centre de calcul professionnel et entretenu, de ce fait la solution cloud hybride est la plus adéquate et qui permet de réunir les avantages des deux approches.

I.7 Classification du cloud computing

L'utilisation du cloud computing ne se limite pas uniquement aux entreprises à caractère commercial. En fonction des raisons de sa mise en place, nous distinguons quatre catégories de plateformes de Cloud Computing à savoir :

9. Workload : est la quantité de traitement que l'ordinateur doit effectuer à un moment donné.

I.7.1 Cloud d'Entreprise

•

Pour les PME¹⁰ disposant généralement de peu de ressources et de moyens de maintenance de leurs matériels se dirigent également vers le cloud à défaut d'espace de stockage insuffisant ou de vouloir concurrencer leurs rivaux sur la marché, se regroupent autour d'un projet de nuage partagé dans le but de mutualiser leurs capacités et de spécialiser leurs administrations. Et pour les grandes entreprises qui occupent une place commerciale importante sur le marché de l'économie mondiale tous secteurs confondus ne peuvent pas rester dans l'utilisation des méthodes classiques pour l'hébergement et le traitement de leurs données, elles font donc recours aux services de cloud computing pour satisfaire leurs clients qui ne cessent d'accroître de jour en jour.

I.7.2 Cloud gouvernemental et recherche scientifique

Pour des raisons de développement et de recherche scientifique, les gouvernements déboursent des budgets importants pour financer des centres de traitement de données dans le but d'encourager l'enseignement et d'offrir un environnement cloud spécialisé aux services de la communauté scientifique sous forme de banques de données et de capacités de calcul de haute technologie réservés aux chercheurs dont l'accès est unique et la sécurité représente un enjeu politique sensible, ces infrastructures étatiques sont également au profit des institutions militaires et des organismes de renseignements et de sécurité nationale.

I.7.3 Cloud pour Réseaux Sociaux et Jeux

Le développement des réseaux sociaux et l'évolution boostée de la qualité des jeux en ligne nécessitent aujourd'hui de grande quantité de ressources que les entreprises ne peuvent pas se permettre en interne, cette nécessité est

10. PME^s : petites et moyennes entreprises.

due à la croissance exponentielle des utilisateurs sur internet également ces dernières années via leurs PCs, Smartphones ou tablettes.

Aujourd'hui, jouer sur internet en streaming ou rester connecté à un réseau social en temps réel ce sont des choses que fait partie du quotidien de l'utilisateur, chose qu'il réalise grâce aux fournisseurs de services de cloud computing qui offrent de leur côté aux entreprises du domaine la disponibilité et la flexibilité de leurs applications et surtout d'encourager l'innovation et de faciliter le déploiement de ces technologies en accélérant le processus de mise en ligne pour n'importe quelle nouvelle application.

I.7.4 Cloud pour Fournisseurs de Services

Le fournisseur de service est une société tierce à caractère commercial qui offre des infrastructures, plateformes et logiciels informatiques à la disposition d'autres sociétés appelées « clientes ». Bien que peu de fournisseurs sont leaders et contrôlent la majeure partie du cloud à l'échelle internationale à l'image de Google, IBM ou Amazon mais il existe des centaines à travers le monde, leur plus grand défi est la sécurité et l'administration de données.

I.8 La virtualisation[2] [3]

La virtualisation est une technique qui simule le comportement d'une ressource physique à l'aide d'un logiciel. Les machines virtuelles qui sont apparues dans les années 1960, sont l'utilisation de techniques de virtualisation sur les ordinateurs. Une machine virtuelle est un programme qui simule un ordinateur complet qui prendra en charge un système d'exploitation conventionnel et utilisera les mêmes fonctionnalités (processeur, mémoire, réseau, etc.) qu'un ordinateur physique. La machine virtuelle, qui n'est qu'un programme informatique, est hébergée sur une machine physique (sur certains matériels, vous pouvez exécuter la machine virtuelle sur une autre machine virtuelle).

La virtualisation est une abstraction de matériel physique, utilisée pour générer des ressources virtuelles pour faire fonctionner plusieurs machines

virtuelles au sein de la même machine physique. Par conséquent, la virtualisation implique trois composants principaux :

- Un système d'exploitation principal installé sur la machine physique, appelé système hôte, car il joue le rôle d'hôte à d'autres systèmes d'exploitation.
- Un hyperviseur un outil de virtualisation installé sur le système hôte qui fournit l'environnement dans lequel différentes machines virtuelles s'exécutent.
- Un système d'exploitation installé dans une machine virtuelle, appelé système invité, qui fonctionne indépendamment des autres systèmes invités dans d'autres machines virtuelles.

I.8.1 Les types de virtualisation

- La virtualisation de stockage.
- La virtualisation de réseau.
- La virtualisation des postes de travail.
- La virtualisation de serveurs.

Virtualisation de stockage

Elle forme une couche de virtualisation active et transparente entre les périphériques de stockage sur disque pour optimiser la disponibilité, les performances et l'utilisation des grands et petits centres de données. L'intégration globale des fonctions de protection, d'approvisionnement, de mise en cache, de réplication et de migration gérées de manière centralisée vise à masquer les caractéristiques physiques des unités de stockage. Les avantages de ce type de virtualisation sont :

- Prolongement du cycle de vie de vos investissements en matière de stockage et ce de manière rentable.
 - Bénéficie d'un premier niveau de Plan de Reprise d'Activité.
-

Virtualisation de réseau

La virtualisation des fonctions réseau (NFV) sépare les fonctions réseau clés (telles que les services d'annuaire, le partage de fichiers et la configuration des adresses IP) pour la distribution dans divers environnements. Lorsque les fonctions logicielles sont indépendantes des machines physiques sur lesquelles elles étaient hébergées, il est possible de regrouper des fonctions spécifiques dans le nouveau réseau et de les affecter à l'environnement. La virtualisation de réseau réduit le nombre de composants physiques requis pour créer de nombreux réseaux indépendants, tels que des commutateurs, des routeurs, des serveurs, des câbles et des concentrateurs. Ceci est particulièrement répandu dans le secteur des télécommunications.

Virtualisation des postes de travail

La virtualisation des postes de travail permet aux administrateurs systèmes et réseaux de gérer beaucoup plus facilement les postes de travail de l'entreprise et de répondre avec flexibilité aux demandes des utilisateurs. Un poste de travail virtualisé ou bureau virtuel peut être hébergé soit directement sur l'ordinateur du client soit sur un serveur dans le centre de données.

Virtualisation des serveurs

Les serveurs sont des ordinateurs conçus pour prendre en charge un grand nombre de tâches spécifiques pour permettre à d'autres ordinateurs (par exemple, des ordinateurs portables et des ordinateurs de bureau) d'exécuter diverses autres tâches. La virtualisation de serveurs est le fait de créer plusieurs serveurs virtuels sur un serveur physique. Ces serveurs créés tournent alors sur la même machine physique, tout en ayant les mêmes propriétés que s'ils avaient chacun une machine physique.

Les avantages de ce type sont :

- Consolidation des serveurs et optimisation de l'infrastructure.
 - Augmentation de la flexibilité et de l'efficacité opérationnelle.
-

- Disponibilité accrue des applications et amélioration de la continuité d'activité.

I.8.2 Solutions de virtualisation

Dans cette section, nous présentons les outils de virtualisation les plus utilisés qui sont Xen, KVM, VMware et HyperV :

- **Xen**

Est une solution libre de virtualisation permettant de faire tourner plusieurs systèmes d'exploitation sur une même machine physique. Il est de type hyperviseur, c'est à dire qu'il vient s'insérer entre le matériel et le noyau. Xen est considéré comme une solution à base de para-virtualisation, car les systèmes invités doivent être modifiés pour cohabiter[9].

- **KVM**

Est un projet de virtualisation complète qui est actuellement en développement pour un module de para-virtualisation. Il est intégré depuis le noyau Linux 2.6.20 et permettant une virtualisation matérielle des processeurs. Ainsi, il ne fonctionne que sur un processeur de type Intel VT ou AMD-V.

- **VMware Server**

Est une société qui offre des produits propriétaires liés à la virtualisation d'architectures x86. Elle est leader dans le marché de la virtualisation pour PC. Son produit de virtualisation VMware Server est de type virtualisation complète pour serveur sous GNU/Linux et/ou Microsoft Windows[10].

- **HyperV**

Est une solution de virtualisation basée sur la virtualisation 64 bits pour Microsoft. Il est considéré comme une solution de paravirtualisation[11].

I.9 Les Hyperviseurs

Un hyperviseur est un logiciel qui permet de créer et d'exécuter des machines virtuelles. Un hyperviseur isole son système d'exploitation et ses ressources des machines virtuelles, et permet de créer et de gérer ces machines virtuelles. L'hyperviseur attribue à chaque machine virtuelle les ressources qui ont été allouées, et gère la planification de leurs ressources en fonction des ressources physiques.

Le matériel physique utilisé en tant qu'hyperviseur est appelé « hôte », tandis que toutes les machines virtuelles qui utilisent ses ressources sont appelées « invités ».

Il existe deux types d'hyperviseurs qui peuvent être utilisés pour virtualiser des ressources : les hyperviseurs de type 1 et les hyperviseurs de type 2.

I.9.1 Les hyperviseurs de type 1

Un hyperviseur de type 1, également appelé hyperviseur de système nu ou natif, s'exécute directement sur le matériel de l'hôte pour gérer les systèmes d'exploitation invités. Il prend la place du système d'exploitation de l'hôte et planifie directement les ressources des machines virtuelles sur le matériel.

Ce type d'hyperviseur est fréquemment utilisé dans les datacenters d'entreprise et dans d'autres environnements basés sur des serveurs. Nous citons comme exemples : Hyper-V, XenServer, VMware ESXi.

I.9.2 Les hyperviseurs de type 2

Un hyperviseur de type 2, également appelé hyperviseur hébergé, s'exécute sur un système d'exploitation traditionnel en tant que couche logicielle ou application.

Il fonctionne en dissociant les systèmes d'exploitation invités du système d'exploitation hôte. Les ressources des machines virtuelles sont planifiées au niveau d'un système d'exploitation hôte, lui-même exécuté sur le matériel. Nous citons comme exemples : KVM, VMware Workstation, VirtualBoX.

Il faut noter que chaque hyperviseur possède son propre logiciel de gestion.

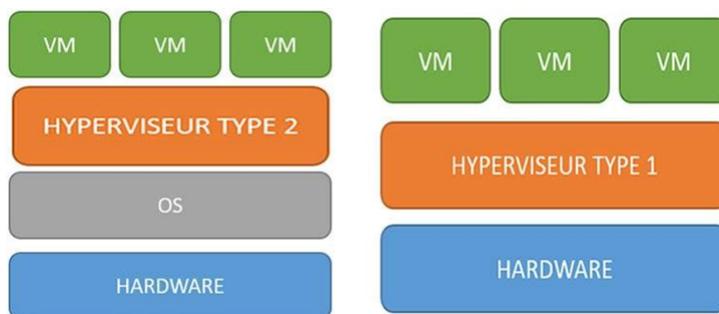


FIGURE I.5 – Les deux types d’hyperviseurs.

I.10 Les Acteurs de cloud computing

Le marché du cloud computing est partagé entre acteurs : les éditeurs, les fournisseurs.

- **Les éditeurs** : Les éditeurs sont des entreprises qui fournissent des solutions cloud. L’éditeur n’est pas forcément un fournisseur de services, en d’autres termes, sa portée n’est pas de fournir un service cloud, mais pour fournir un moyen d’héberger une solution cloud.
- **Les fournisseurs** : les fournisseurs de cloud font référence aux entreprises qui fournissent des environnements informatiques (tels que des clouds publics ou des clouds privés hébergés) qui peuvent extraire, collecter et partager des ressources évolutives via le réseau. Certains fournisseurs de cloud proposent également des services en ligne tels que l’IaaS (Infrastructure-as-a-Service), le PaaS (Platforms-as-a-Service) et le SaaS (Software-as-a-Service).

Voici quelques fournisseurs majeurs de cloud computing :

- **SalesForce**[12]
 - Le plus ancien, propose des offres depuis 2003 (API SForce).
 - Cloud public uniquement.
 - Eco-système d'applications AppExchange (800).
 - Offre gratuite assez limitée.
 - Datacenters à l'étranger.
 - Vise essentiellement les Grands Comptes.



FIGURE I.6 – Logo du salesForce.

- ▷ Service de type PaaS :
 - Développement rapide à la souris pour assembler différentes briques.
 - 2 langages sont proposés : Apex (java-like) et Visualforce (xml-like).
- ▷ Service de type SaaS :
 - Logiciel pour gérer la relation client (GRC¹¹).
 - SLA¹² disponible mais variable selon les clients.
- ▷ Base de données :
 - Intégrée à l'environnement de développement.
 - Gestion des tables et de leurs relations WYSIWYG¹³.

11. GRC : ensemble des outils et technologies destinés à la bonne Gestion de la Relation Client.

12. SLA (Service Level Agreement) : Contrat de fourniture des services aux clients.

13. WYSIWYG (What You See Is What You Get) : Ce que vous voyez est ce que vous obtenez

- **Amazon - Web Services (AWS)[13]**
 - Lancé en version complète en 2006 et mise à jour régulière depuis.
 - Cloud public, aussi disponible en cloud privé.
 - SLA disponible
 - Basé sur Xen.



FIGURE I.7 – Logo du amazon.

- ▷ Service de type IaaS :
 - Elastic Compute Cloud (EC2) : intégration de machines virtuelles, existantes ou nouvellement créées via AWS
- ▷ Service de type PaaS :
 - Elastic MapReduce : même calcul en parallèle sur de grosses quantités de données.
 - Simple Queue Service (SQS) : système de messagerie entre applications.
- ▷ Service de type SaaS :
 - Simple Storage Service (S3) : écriture et lecture de flux de données.
 - Elastic Block Storage (EBS) : disque dur privé pour la partie EC2.
 - CloudFront : offre CDN¹⁴ qui repose sur du contenu stocké sur S3.
- ▷ Base de données :
 - SimpleDB : services de base d'indexation et de recherche par clef.

14. CDN (Content Delivery Network) :réseau de diffusion de contenu.

- Relational Database Service : compatible avec MySQL.
- **Google - App Engine**[14]
 - Lancé en avril 2008, mis à jour en avril 2009.
 - Cloud public uniquement.
 - Offre gratuite conséquente (plusieurs millions de requêtes par mois).
 - Orienté services web.
 - Pas de SLA, support via un forum public.
 - Beta (comme beaucoup de produits Google).
 - Plus pour les startups que les Grands Comptes.



FIGURE I.8 – Logo Google App-Engine.

- ▷ Service de type Paas :
 - 2 langages supportés : java et python.
 - Beaucoup de limitations (durée des requêtes, restriction sur les API).
 - NB** : API (Application Programming Interface) : un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

 - ▷ Service de type SaaS :
 - GMail
 - Google Documents
 - Base de données
 - Datastore (technologie propriétaire BigTable)
-

— Langage de requête SQL-lite (GQL¹⁵)

15. GQL (Graph Query Language) : nouveau langage d'interrogation de requêtes graphiques.

- **Microsoft - Windows Azure**[15]
 - Lancé le premier février 2010
 - Cloud public uniquement
 - Orienté services web ou traitement long
 - Datacenter en Europe (Dublin et Amsterdam)
 - SLA disponible, variable selon les services.
 - Eco-système d'applications avec une nouvelle version de Pinpoint¹⁶.



FIGURE I.9 – Logo Microsoft - Windows Azure.

- ▷ Service de type PaaS :
 - Langages supportés (CLR, Java SDK, Ruby SDK)
 - AppFabric : connectivité et gestion d'identités
 - Système de messagerie inter applications
- ▷ Service de type SaaS :
 - Office Web Apps
 - Service de type StaaS
 - Blob Service : stockage de texte et de données binaire
- ▷ Base de données :
 - SQL Azure : SGBDR, compatible SQL Server

16. Pinpoint : répertoire d'applications logicielles et services professionnels consultable en ligne.

I.11 Les vulnérabilités dans la sécurité dans du cloud

Cependant, le cloud computing est un défi majeur pour les fournisseurs de services, leurs clients et bien sûr pour les attaquants externes. Des milliers d'informations provenant de différents clients sont stockées au même endroit et exposées à Internet. C'est la cible idéale pour des centaines d'utilisateurs malveillants.

La sécurité est un aspect qui doit être pris en compte lors du déploiement d'un Cloud. L'adoption des technologies basées sur le Cloud Computing est inévitable, mais des précautions doivent être prises en compte pour éviter que les gains potentiels pour avoir adoptés ne soient pas compromis par les menaces de sécurité.

En fait, selon une étude récente, la sécurité est le plus grand défi auquel sont confrontés les responsables informatiques qui souhaitent adopter des solutions et des services d'hébergement cloud. Selon un rapport rédigé par l'ENISA fin 2009, les menaces pesant sur le cloud computing peuvent être réparties dans les catégories suivantes :

I.11.1 La perte de maîtrise de gouvernance

Le client a cédé une partie de la puissance de gestion de l'infrastructure informatique au fournisseur. Dans ce cas, le contrat de niveau de service (SLA) doit jouer un rôle clé dans la défense des intérêts des clients.

I.11.2 Isolement des environnements et des données

Le partage des ressources est l'une des caractéristiques les plus importantes du cloud computing. Par exemple, plusieurs clients peuvent partager le même serveur physique. Si la séparation de l'environnement n'est pas suffisamment efficace, une «intrusion» peut se produire entre les clients. Dans le cas d'un environnement partagé entre plusieurs «clients locataires», deux types d'attaques peuvent se produire, l'un est de type «guest-hopping» et

l'autre est une attaque directe sur l'hyperviseur.

I.11.3 Risques de conformité

En externalisant certains services et processus de base, le respect des lois sur la protection des données et des normes réglementaires (telles que PCI DSS et ISO 27001) peut devenir très compliqué. Les prestataires de services peuvent imposer des restrictions à l'audit de leur infrastructure.

I.11.4 La publication des interfaces de gestion

L'interface de gestion du service contractuel (par exemple, dans le modèle SaaS) est publiée directement sur le réseau. Par rapport au système traditionnel qui ne peut accéder à l'interface de gestion qu'à partir du réseau interne, cela augmente considérablement le risque.

I.11.5 La protection des données

Pour les clients des services de cloud computing, la protection des données est très difficile. Il est très difficile de protéger les données distribuées à plusieurs endroits. S'assurer que les données sont traitées correctement est également compliqué parce que le contrôle sur les transferts de données est hors de la portée de son propriétaire.

I.11.6 La suppression dangereuse ou incomplète des données

La suppression sûre des données est une question très complexe qui implique le développement d'une série de processus pour s'assurer qu'il n'y a pas de copie des données à aucun endroit. La réutilisation des ressources matérielles est très courante dans le cloud computing. Par exemple, être affecté d'une section de stockage dans lequel, jusqu'à récemment, les données d'un autre client ont eu lieu. Cela peut entraîner à un risque de perte de confidentialité, si les données précédentes n'ont pas été supprimées complètement et

en toute sécurité.

I.11.7 Les utilisateurs malveillants

Le cloud computing nécessite des profils utilisateur de haut niveau pour la gestion. L'administrateur système dispose de tous les privilèges sur différentes ressources de différents clients. Un utilisateur malveillant qui a réussi à enfreindre la sécurité du système et à accéder à la session administrateur aura accès à de nombreuses informations des clients.

I.12 Conclusion

Le cloud computing aide à réduire les problèmes qu'ils rencontrent Les entreprises, par exemple : espace de stockage, mise en œuvre de diverses applications Et la disponibilité des données. Donc le Cloud est un modèle de distribution et de partage de ressources à grande échelle, qu'il offre aux utilisateurs un environnement plus simple et plus efficace pour travailler.

Plateformes de cloud computing

II.1 Introduction

Dans ce chapitre, nous allons nous intéresser aux plateformes existantes sur le marché mondial du cloud computing puis nous choisissons une solution adéquate à nos besoins et présenter ses services de base, ses composants, et son architecture et nous finissons par identifier ses avantages ainsi que ses limites.

II.2 Plateformes de cloud computing

Une plateforme de cloud computing est un ensemble d'outils logiciels intégrés qui permet de surveiller et contrôler les ressources de cloud computing. Ce type de plateforme peut être utilisé exclusivement pour un déploiement de cloud privé ou public, mais cible en général les modèles hybrides et multi-cloud, avec pour objectif un contrôle centralisé de diverses infrastructures en cloud.

Les fonctionnalités exactes d'une plateforme du cloud varient selon les fournisseurs. Certains offrent de nombreux outils, tandis que d'autres se contentent à des secteurs de niche ou à des marchés verticaux.

Les solutions du cloud sont classées en deux grandes catégories : les solutions propriétaires (payantes) et les solutions open source (gratuites pour services de base).

II.2.1 Les solutions propriétaires

Amazon EC2

Elastic Compute Cloud (Amazon EC2) est un service Web proposé par la plateforme Amazon depuis le 25 août 2006, fournit une capacité de calcul sécurisée et redimensionnable dans le cloud. Destiné aux développeurs, EC2 permet un déploiement extensible des applications en fournissant une interface web par laquelle un client peut créer des machines virtuelles, c'est-à-dire des instances du serveur, sur lesquelles le client peut charger n'importe quel

logiciel de son choix[16].



FIGURE II.1 – Logo du Amazon EC2.

Pour répondre aux besoins de chacun, Amazon propose une gamme de types d'instances aux configurations différentes en termes d'UC, de mémoire, et de ressources de stockage et réseau. En outre, chaque type est disponible en plusieurs tailles pour satisfaire les différentes charges de travail.

L'utilisateur peut donc créer une instance du serveur à partir d'images machines appelées AMI (Amazon Machine Images). Ces images sont comparables à des modèles configurés avec un système d'exploitation et d'autres logiciels, le tout formant l'environnement d'exploitation de l'utilisateur.

Alibaba Cloud (ALIYUN)

Alibaba Cloud, est une multinationale de cloud computing lancée en 2009 par le géant chinois du e-commerce Alibaba Group. Il est également le 4ème principal fournisseur de services cloud dans le monde. Ce service multiplateforme fournit des services informatique à travers une gamme complète de produits pour les bases de données, la mise en réseau, la sécurité, l'analyse et le Big Data, la gestion de domaines et de sites Web, les services d'application, les services multimédias, les intergiciels, etc.[17]



FIGURE II.2 – Logo du Alibaba Cloud.

ECS (Elastic Compute Service) d’Alibaba Cloud est proposé dans 17 régions internationales avec plusieurs zones de disponibilité toute gérée par un seul compte mondial basé à Singapore. Cela grâce à une infrastructure de pointe (serveurs, processeurs, supercalculateurs) et une protection anti-DDos intégrée. Ce service comprend « Auto Scaling » et « Server Load Balancer », deux moteurs facturés uniquement en fonction de l’utilisation du runtime, comme la création d’instance ECS, les durées réelles de location de l’équilibre de charge public ou privé et le réseau circulation.

IBM Cloud

IBM Cloud est un ensemble de services de cloud computing pour les entreprises proposé par la société de technologie de l’information IBM (International Business Machines) à partir de 2007, connue comme la société mère de virtualisation avec le développement de premières machines virtuelles et de calcul mainframe dans les années 1960s. Cette plateforme combine les modèles de services PAAS et IAAS. Elle propose des outils pour la collaboration, le développement et les tests basés sur le cloud, l’analyse, l’intégration business-to-business et aide à moderniser les applications existantes, à renforcer la sécurité et à développer de nouvelles applications qui boostent les initiatives numériques. Il offre des solutions de développement natif cloud qui peuvent rapidement générer de la valeur, ainsi que des licences flexibles adaptables aux besoins spécifiques[18].



FIGURE II.3 – Logo du IBM Cloud.

A l'aide d'une infrastructure d'application appelée WebSphere, IBM prend en charge les modèles de programmation et les normes ouvertes pour une meilleure virtualisation. Ajoutant à cela une plateforme multicloud hybride de nouvelle génération, des fonctionnalités avancées pour la gestion de base de données et d'Intelligence Artificielle, big data et une expertise approfondie des entreprises dans 20 secteurs d'activité, ce qui fait de lui le fournisseur de cloud public le plus ouvert et le plus sécurisé du marché.

II.2.2 Les solutions open source

Eucalyptus

C'est la solution OpenSource la plus répandue de cloud computing. Issue en 2007 du projet VGrADS à l'université de Californie. Eucalyptus est un acronyme de « Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems », qui se traduit par « Utilitaire d'Architecture informatique élastique pour relier vos programmes à des systèmes fonctionnels »[19].

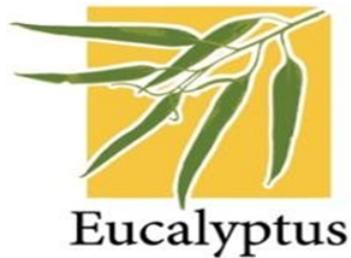


FIGURE II.4 – Logo du Eucalyptus.

Elle permet de créer des clouds Iaas (Infrastructure as a service) de type privé ou hybride. Elle est compatible à Amazon Web Services, qui permet d'utiliser son cloud computing privé sur un cloud computing public peut s'installer facilement sur la majorité des distributions GNU/Linux : Debian, CentOS.

OpenNebula

Le projet OpenNebula a commencé comme une entreprise de recherche en 2005 par Ignacio M. Llorente et Ruben S. Montero. La première version publique du logiciel a eu lieu en 2008[20].



FIGURE II.5 – Logo du OpenNebula.

OpenNebula est une plateforme de Cloud Computing Open source pour la gestion des Datacenter hétérogènes et distribués. Cette plateforme gère l'infrastructure virtuelle du Datacenter de construction privée, publique et hybride implémentant le service IaaS. OpenNebula supporte les hyperviseurs XEN, KVM et optionnellement VMware.

Openstack

Le projet « Openstack » lancé en 2010 par Rackspace Hosting et la NASA est une multi- plateforme cloud disponible soit gratuitement soit personnalisé avec les services demandés selon les besoins de l'utilisateur[4].



FIGURE II.6 – Logo du Openstack.

OpenStack permet de créer et gérer des clouds privés ou publics, implémentant le service IaaS à partir de pools de ressources virtuelles. Les outils qui constituent la plateforme OpenStack assurent les principaux services de cloud computing, à savoir, le calcul, la mise en réseau, le stockage, la gestion des identités et la gestion des images.

II.3 Identification des besoins

Cette première partie, sera pour énoncer et analyser les différents besoins fonctionnels et non fonctionnels du nuage.

II.3.1 Besoin Fonctionnels

Gestion d'images

Image système stockée par le service de gestion d'images de la plateforme, permet aux utilisateurs de découvrir, enregistrer et récupérer les images de machines virtuelles, lancer des nouvelles instances de cette image, créer une nouvelle ou supprimer une existante.

Gestion d'instances

Une instance est une machine virtuelle qui s'exécute dans le cloud. L'utilisateur pourrait consulter la liste d'instances des machines virtuelles actuelles, il aura aussi les possibilités d'éditer, mettre fin, en pause, redémarrer ou supprimer une instance. Egalement il pourra se connecter à la console VNC¹ de l'instance ou créer une nouvelle.

Gestion des volumes

Un volume est un périphérique de stockage de masse que vous attachez à une instance pour activer le stockage persistant. Les utilisateurs peuvent attacher un volume à une instance en cours d'exécution ou détacher un volume et le joindre à une autre instance à tout moment.

Gestion des flavors

Un flavor (gabarit) définit la capacité de calcul, de mémoire et de stockage d'un serveur virtuel, également connu sous le nom d'instance. En tant

1. VNC (Virtual Network Computing) : système de virtualisation et de contrôle de l'environnement de bureau d'un ordinateur distant.

qu'utilisateur administratif, vous pouvez créer, modifier et supprimer des flavors.

Gestion des projets et les utilisateurs

L'utilisateur peut créer des projets et créer des comptes pour de nouveaux utilisateurs à l'aide du Dashboard OpenStack. Et aussi il peut associer les utilisateurs à des rôles, des projets ou les deux.

Gestion de la sécurité et de l'accès

L'utilisateur pourrait consulter les adresses IP disponibles pour connecter les instances au réseau public avec la possibilité de création de groupes de sécurité et leurs règles, pare-feux et leurs interfaces d'édition et enfin la liste de clés SSH ou certificats.

II.3.2 Besoins non fonctionnels

Simplicité d'un service à la demande

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

Extrême flexibilité

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

Accès léger

Aucun matériel ou logiciel propriétaire n'est requis pour accéder aux ressources. Il se fait au travers d'applications facilement disponibles (parfois libres), généralement depuis un simple navigateur Internet.

Suret 

Les  v nements ind sirables ne doivent pas se produire lors de l'acc s   la machine virtuelle.

Vivacit 

Les op rations requises par la machine virtuelle doivent assurer l'avancement du programme.

II.3.3 Solution propos e

Apr s avoir pr sent  quelques plateformes de cloud computing et l'identification des besoins, il est question maintenant de faire le choix qui nous convient le plus.

Dans notre cas la solution Openstack est la plus r pondue   nos crit res, car son but principal est de construire un Cloud priv  pur, elle poss de une interface souple et simple   utiliser et assure une interaction facile avec le cloud.

II.4 Pr sentation d'openstack

II.4.1 D finition

OpenStack est un ensemble d'outils logiciels pour la construction et la gestion de plateformes de cloud computing pour les nuages publics et priv s ; un environnement open source qui donne un contr le complet sur le calcul du cloud.

Le projet « Openstack » lanc  en 2010 par Rackspace Hosting et la NASA est une multi-plateforme cloud disponible soit gratuitement soit personnalis  avec les services demand s selon les besoins de l'utilisateur. La technologie poss de une architecture modulaire compos e de plusieurs projets corr l s (Nova, Swift, Glance, etc) qui permettent de contr ler les diff rentes ressources des machines virtuelles telles que la puissance de calcul, le stockage

ou encore le réseau inhérents au centre de données sollicité.

La dernière version d'Openstack est la 22ème nommée « Victoria » sortie le 14/10/2020 et qui comprend des améliorations dans les fonctionnalités de base, l'automatisation, la migration à froid intercellulaire, les applications conteneurisées et la prise en charge de nouveaux cas d'utilisation à plusieurs niveaux de la pile. Elle intègre également des fonctionnalités avancées de mise en réseau et Kubernetes native.

Aujourd'hui, OpenStack est soutenu par une communauté open source mondiale et à ce rythme, il continue d'être l'un des trois meilleurs projets open source au monde en termes de contributions actives, avec le noyau Linux et Chromium.

A noter qu'en octobre 2020, l'OpenStack Foundation change de nom et devient l'**Open Infrastructure Foundation (OIF)** pour marquer le franchissement d'une nouvelle étape dans son évolution continue, pour créer des communautés open source qui écrivent des logiciels d'infrastructure en production.

II.4.2 Historique

OpenStack, développé initialement par Rackspace et la NASA en juillet 2010, propose un regroupement de logiciel open source sous licence Apache pour mettre en place une IaaS (infrastructure as a service) publique ou privée.

Cet ensemble de briques permet de configurer et d'utiliser les parties : calcul, stockage, orchestration et réseau d'un Cloud IaaS. Historiquement, Rackspace contribuait à la partie stockage d'OpenStack tandis que la partie calcul était développée par la Nasa. A l'heure actuelle, OpenStack a une fondation, nommée OpenStack Consortium, et intègre plus de 150 entreprises, dont RedHat Canonical, Dell et Citrix. La gouvernance de ce produit est assurée par la communauté et par les entreprises sponsorisant la fondation.

L'objectif du projet OpenStack est de permettre à toute organisation de créer et d'offrir des services de *cloud computing* en utilisant du matériel standard. La première version livrée au public par la communauté, dont le surnom est « Austin », fut disponible dès octobre 2010 et la dernière version

est « Victoria » réalisée le 14 octobre 2020 comme cité auparavant. Selon la politique de l'entreprise, il est prévu de livrer régulièrement des mises à jour logicielles à quelques mois d'intervalle (environ 6 mois)[21].

II.5 Les composants d'Openstack

II.5.1 Services de base[4]

Interface utilisateur : Horizon

C'est le tableau de bord du système OpenStack. Il offre toutes les possibilités pour les administrateurs système d'accéder et de gérer le cloud. C'est le premier composant que tout le monde "voit" en commençant à utiliser l'OpenStack; les développeurs seront également en mesure d'accéder à tous les composants via l'interface de programmation d'applications (API), tandis qu'Horizon est le seul endroit où les administrateurs du système interagiront avec l'architecture OpenStack, il utilise comme framework Django et permet ainsi de visualiser les différents composants d'OpenStack et d'agir dessus.

Service de calcul : Nova

C'est le moteur informatique fondamental d'OpenStack; le calcul est l'un des composants les plus importants et obligatoires dans le cloud computing. Il s'agit essentiellement d'un hyperviseur de virtualisation, il agit comme un contrôleur qui gère toutes les ressources dans un environnement virtuel. Il est également utilisé pour gérer les configurations du métal nu de haute performance. Codé en Python et a utilisé de nombreuses bibliothèques prédéfinies pour offrir un fonctionnement robuste. Les technologies d'hyperviseur qui pourraient être utilisées sont Xen, KVM, et VMware selon la version d'OpenStack utilisée. Pour l'accès à la base de données, Openstack fait appel aux services du système de gestion SQL.

- **Fonctionnalités de nova**

- **Le nova-api** : gère les demandes et les réponses de/vers l'utilisateur final.
- **Le nova-compute** : créé et détruit les instances au fur et à mesure qu'une demande est faite.
- **Nova-scheduler** : planifie les tâches à la nova-compute.
- **Le registre des glaces** : stocke les détails de l'image ainsi que ses métadonnées.
- **Le magasin d'images** : stocke les images prédéfinies par l'administrateur/utilisateur.
- **Le nova-network** : assure la connectivité et le routage du réseau.
- Une «base de données API» utilisée principalement par nova-api et nova scheduler pour suivre les informations d'emplacement relatives aux instances, ainsi qu'un emplacement temporaire pour les instances en cours de création mais non encore planifiées.
- Le service nova-conductor qui décharge les tâches de longue durée pour le service de niveau et joue le rôle de proxy de base de données ou gère les conversions d'objet.

Service de gestion de réseautage : Neutron

Ceci est responsable de l'établissement d'une structure neuronale entre les composants pour une meilleure connectivité. Il gère tous les aspects réseau pour l'infrastructure de réseau virtuel et les aspects de la couche d'accès à l'infrastructure de réseau physique dans votre environnement OpenStack ; le réseau OpenStack permet aux tenants de créer des topologies de réseau virtuel avancées qui peuvent inclure des services comme un firewall, un load balancer, et un réseau privé virtuel (VPN). Il fournit des réseaux, sous-réseaux et routeurs comme des abstractions objet ; chaque abstraction a des fonctionnalités qui imite son homologue physique : les réseaux contiennent des sous-réseaux, et les routeurs dirigent le trafic entre les différents sous-réseaux et réseaux. Chaque routeur a une unique passerelle qui se connecte à un

réseau et de multiples interfaces connectées à des sous-réseaux.

Les sous-réseaux peuvent accéder à des machines sur d'autres sous-réseaux connectés au même routeur. Ainsi un utilisateur peut créer son propre réseau de machines virtuelles.

Neutron donne le choix à l'utilisateur d'OpenStack de créer au choix deux options de réseau :

1. **Option de mise en réseau 1 : réseaux de fournisseur** : L'option de réseau de fournisseur déploie le service de réseau OpenStack de la manière simple. Il relie essentiellement les réseaux virtuels aux réseaux physiques et s'appuie sur l'infrastructure de réseau physique pour les services de couche 3 (routage). En outre, un service DHCP fournit des informations d'adresse IP aux instances.
2. **Option de mise en réseau 2 : réseaux en libre-service** : L'option de réseaux en libre-service complète l'option de réseaux de fournisseurs avec les services de couche 3 (routage) qui activent les réseaux en libre-service à l'aide de méthodes de segmentation en superposition telles que VXLAN². En plus clair, on peut créer des réseaux personnalisés ensuite des routeurs virtuels qui les relient vers les réseaux physiques en utilisant NAT³.

2. VXLAN : Virtual Extensible LAN.

3. NAT : Network Address Translation.

L'architecture de neutron est composée de ces éléments de base :

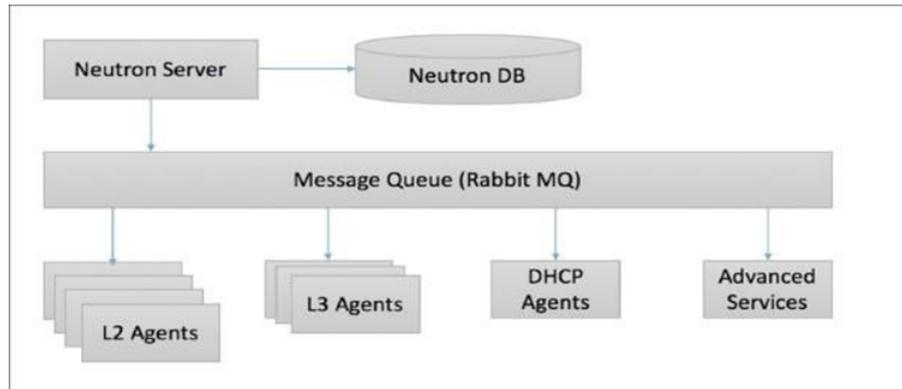


FIGURE II.7 – Architecture du composant Neutron.

Explication

Le serveur Neutron : La fonction de ce composant est d'être le visage de l'environnement Neutron dans son ensemble. Il est essentiellement composé de trois modules :

- **Service REST :** le service REST accepte les demandes d'API des autres composants et expose tous les détails de fonctionnement internes en termes de réseaux, sous-réseaux, ports, etc. C'est une application WSGI⁴ écrite en Python qui utilise le port 9696 pour la communication.
- **Service RPC :** le service RPC⁵ communique avec le bus de messagerie et sa fonction est d'activer une communication d'agent bidirectionnel.
- **Plugin :** un plugin se décrit mieux comme une collection de modules Python qui implémente une interface standard, qui accepte et reçoit

4. WSGI : Web Server Gateway Interface.

5. RPC : Remote Procedure Call.

certains appels d'API standard, puis se connecte aux périphériques en aval. Ils peuvent être de simples plugins ou implémenter des pilotes pour plusieurs classes de périphériques.

Agent L2 : L'agent L2 s'exécute sur l'hyperviseur (noeuds de calcul) et sa fonction consiste simplement à connecter de nouveaux périphériques, ce qui signifie qu'il fournit des connexions à de nouveaux serveurs dans des segments de réseau appropriés et qu'il envoie également des notifications lorsqu'un périphérique est connecté ou supprimé. Dans notre installation, nous utiliserons l'agent OVS⁶.

Agent L3 : Les agents L3 s'exécutent sur le noeud de réseau et sont responsables du routage statique, du transfert IP et d'autres fonctionnalités de L3, telles que DHCP.

Processus de base de Neutron : Les étapes de lancement de nouvelle instance avec neutron sont la suivante :

- (a) Démarrage de la machine virtuelle.
- (b) Créez un port et informez le DHCP du nouveau port.
- (c) Créez un nouveau périphérique (bibliothèque de virtualisation - libvirt).
- (d) Port fil (connectez la machine virtuelle au nouveau port).
- (e) Démarrage complet.

Service de stockage en bloc : Cinder

Cinder est connu comme le composant de stockage en bloc d'OpenStack. Cela fonctionne d'une manière analogue aux méthodes traditionnelles de localisation et d'accès à des emplacements spécifiques sur un disque ou un disque dur ; il fournit des installations de stockage de niveau persistantes

6. OVS (Open VSwitch) : Implémentation open source d'un commutateur multicouche virtuel distribué.

pour les appareils informatiques orientés vers le cloud. Il aide à la création, l'ajout et la suppression des nouveaux périphériques de bloc dans le serveur. Il fournit une intégration complète avec l'OpenStack réduisant l'utilisation de composants matériels extraterrestres. Il peut être consulté et indiqué à l'aide de la CLI⁷ ou de services Web. Les systèmes basés sur Linux peuvent aller avec Cloudbyte, ECS (Elastic Cloud Storage), Coraid, SAN Storage, etc.

Architecture de Cinder

Architecture de Cinder est composée des éléments suivants :

- **Cinder-api** : c'est la passerelle principale vers le composant Cinder.
- **Cinder-scheduler**
 - Prend la demande du service api.
 - Fonctionne avec le service de volume pour satisfaire la demande.
 - Fonctionne généralement sur le noeud du contrôleur.
- **Cinder-backup** : est une interface pour les volumes de sauvegarde sur le stockage comme TSM⁸, Google Cloud Storage, il est capable de passer à plusieurs nœuds pour une opération de simulation.
- **Cinder-volume driver**
 - Interagir avec les systèmes de stockage des fournisseurs.
 - Peut fonctionner sur un nœud de contrôleur.
 - Parfois déchargé sur un hôte différent.

7. CLI : Command Line Interface.

8. TSM (Trivoli Storage Manager) : Logiciel de sauvegarde informatique.

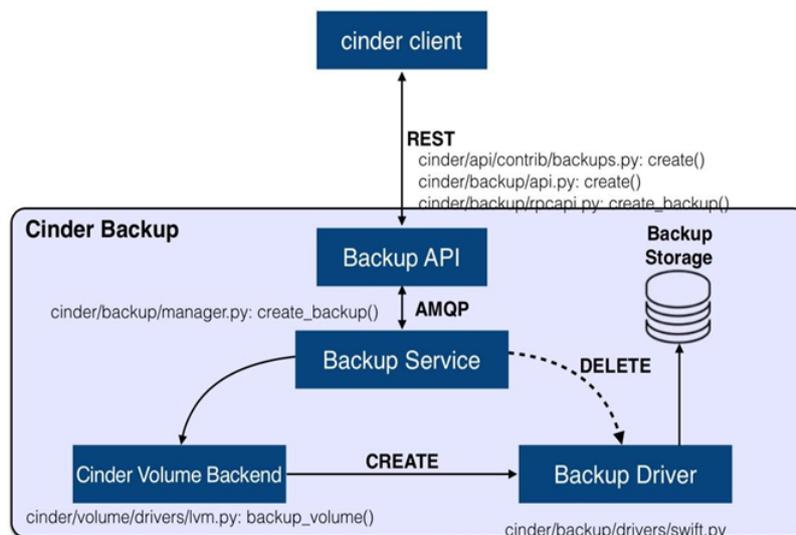


FIGURE II.8 – Architecture du composant Cinder.

Service de stockage objet : Swift

Est un système de stockage pour les objets et les fichiers ; plutôt que l'idée traditionnelle d'une référence aux fichiers par leur emplacement sur un disque dur, cela rend la mise à l'échelle facile. Et garantit également que les données sont sauvegardées, en cas de défaillance de la machine ou de perte de réseau.

II.5.2 Services partagés

Service d'identité : Keystone

Représente un service d'identité centralisé et de services d'autorisation sur la pile ouverte ; il fournit un annuaire central contenant la liste des services et la liste des utilisateurs d'Openstack ainsi que leurs rôles et autorisations. Au sein d'Openstack, tous les services et tous les utilisateurs utilisent Keystone afin de s'authentifier les uns avec les autres. Ce service supporte plusieurs formats d'authentification comme les mots de passe et autres.

Architecture de keystone

Keystone est organisé en groupe de services exposés aux ordinateurs d'extrémité. Il y a 6 services :

- **Identity** : valide les informations d'identification des utilisateurs.
- **Ressource** : fournit des données sur les projets.
- **Assignment** : fournit des données sur les rôles et les attributions de rôles entre les objets de Services d'identité et de ressources.
- **Token** : valide et gère les Tokens.
- **Catalog** : fournit un registre de points de terminaison.
- **Credentials Backend** : est une table de liaison entre un Credential et un Backend, utilisée simplement parce qu'un utilisateur peut avoir une information d'identification (par exemple une clé SSH) qui est valide sur plusieurs backends.

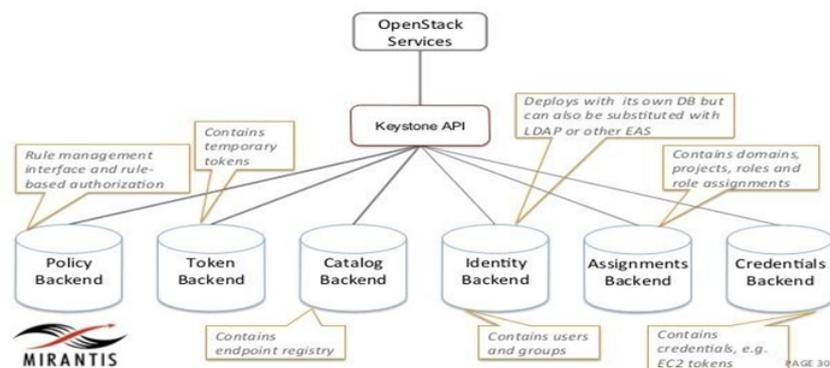


FIGURE II.9 – Architecture du composant Keystone.

Service d'images : Glance

Glance est un service qui sert de registre ou de conteneur d'images de machines virtuelles. Avec la technique qu'on appelle « le coup d'œil », il permet à l'utilisateur de copier les images dans une instance (machine virtuelle)

plus rapidement car les services de coup d'œil font des images de machines virtuelles en tant que modèles stockés dans la mémoire. Glance peut stocker ces images disques de plusieurs façons : dans un dossier sur serveur, mais aussi à travers le service de stockage objet d'OpenStack ou dans des espaces de stockages décentralisés comme Ceph. En plus des images Glance stocke aussi des informations sur celles-ci, les métadonnées par exemple le format du disque (comme QCOW2⁹ ou RAW) ou les conteneurs de celles-ci (OVF¹⁰ par exemple).

Architecture de Glance

Glance a les composants suivants :

- **Glance-api**, qui accepte les appels d'API pour la découverte, la récupération et le stockage d'images.
- **Glance-registry**, qui stocke, traite et récupère les informations de métadonnées pour les images.
- **Base de données**, qui stocke le **référentiel de stockage** des métadonnées de l'image, qui s'intègre à divers composants externes à OpenStack tels que les systèmes de fichiers classiques, Amazon S3 et HTTP pour le stockage des images.

9. QCOW2 : Qemu Copy On Write.

10. OVF : Open Virtual Machine Format.

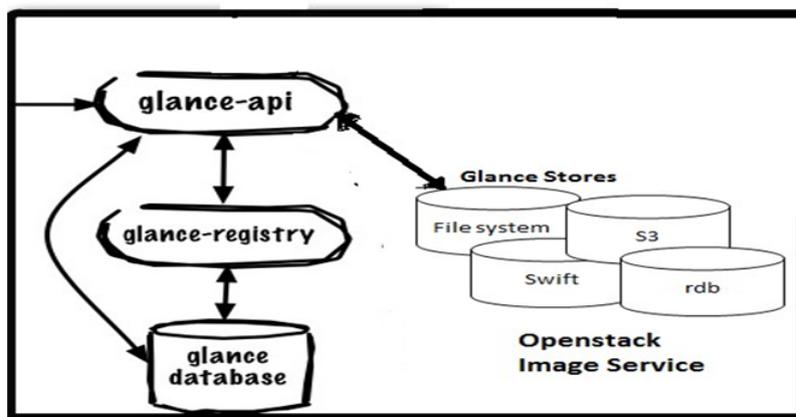


FIGURE II.10 – Architecture du composant Glance.

Service d'orchestration : Heat

Heat est un service utilisé pour compiler / collaborer plusieurs applications cloud composites à l'aide d'AWS en utilisant l'API REST (Representational State Transfer) et de l'API de requête compatible CloudFormation. Ce logiciel intègre d'autres composants d'Openstack dans un modèle à un fichier. À partir de ces modèles, il permet la création de presque tous les types de ressources openstack telles que les instances, les flottantes, les adresses IP, les volumes, les groupes de sécurité, les utilisateurs, ainsi que des fonctionnalités avancées telles que la haute disponibilité, les instances de mise à l'échelle automatique et les piles imbriquées.

Service de métrologie : Telemetry

Il effectue une réglementation étroite de l'utilisation des composants cloud de chaque utilisateur et fournit une facture pour les services utilisés.

Anciennement appelé Ceilometer, ce service permet de collecter différentes métriques sur l'utilisation du cloud. Par exemple il permet de récolter le nombre d'instances lancées dans un projet et depuis combien de temps. Ces métriques peuvent être utilisées pour fournir des informations nécessaires à un système de facturation.

Par exemple : elles sont aussi utilisées dans les applications ou par d'autres composants d'Openstack pour définir des actions en fonction de certains seuils comme avec le composant d'orchestration (Heat).

Service de base de données : Trove

Il fournit une fonctionnalité Database-as-a-Service, évolutive et fiable qui permet d'installer et de gérer facilement des instances de base de données relationnelle et NoSQL au sein d'OpenStack. À ce jour les services de base de données supportés sont les suivants : MySQL, Redis, PostgreSQL, MongoDB, Cassandra, Couchbase et Percona.

Service de traitement de données : Sahara

Il a pour but de fournir aux utilisateurs les moyens simples de provisionner des clusters de Hadoop en spécifiant plusieurs paramètres comme la version, la topologie du cluster ou d'autres. Après avoir rempli ces paramètres, Sahara déploie le cluster en quelques minutes et fournit aussi les moyens d'évolution du cluster en rajoutant des nœuds à la demande.

II.5.3 Autres services (les services additionnels)

- **Ironic** : service de Bare Metal : ensemble de composants qui permettent la gestion et le déploiement de machines physiques.
 - **Manila** : service de gestion de systèmes de fichiers partagés, fournit un accès coordonné à des systèmes de fichiers partagés ou distribués.
 - **Zaqar** : service de middleware à la demande, est un service de messagerie qui permet aux développeurs de partager des données entre des composants d'application distribuée exécutant différentes tâches, sans perte de messages ni nécessité que chaque composant soit hautement disponible.
 - **Designate** : service de gestion des DNS, il fournit des fonctionnalités de provisionnement cloud pour les zones et les enregistrements DNS.
-

- **Barbican** : service de gestionnaire de clés, fournit une API RESTful pour le stockage et la fourniture de données secrètes comme des passphrases, clés de chiffrement, et certificats X.509.
- **Magnum** : service de gestion d'infrastructure de conteneurs, est un service API OpenStack permettant à des moteurs d'orchestration de conteneurs (COE) comme Docker Swarm, Kubernetes and Mesos d'être disponibles comme ressources de premier plan dans OpenStack.
- **AODH** : Service d'alarme de télémétrie, il remonte des alertes quand les compteurs collectés ou les données d'événements enfreignent les règles définies.

II.6 Architecture d'Openstack

OpenStack n'est pas un simple outil de virtualisation, c'est une solution aboutie de Cloud computing. Le projet regroupe un ensemble de logiciels pour gérer les ressources de calcul, stockage distribué, réseau des instances, orchestration des services et enfin la redondance de l'ensemble de ses composants. L'architecture de cette solution est conçue pour être très modulaire.

II.6.1 Architecture logique

D'un point de vue logique, OpenStack peut être représenté au minimum avec trois services : un bloc de Compute, un bloc de Networking et un bloc de Storage. Ces services utilisent un bus de communication pour communiquer entre eux. Ils exposent publiquement leurs fonctionnalités via des API. Un dashboard permet de se connecter aux services avec un portail présentant une IHM (Interface Homme-Machine).

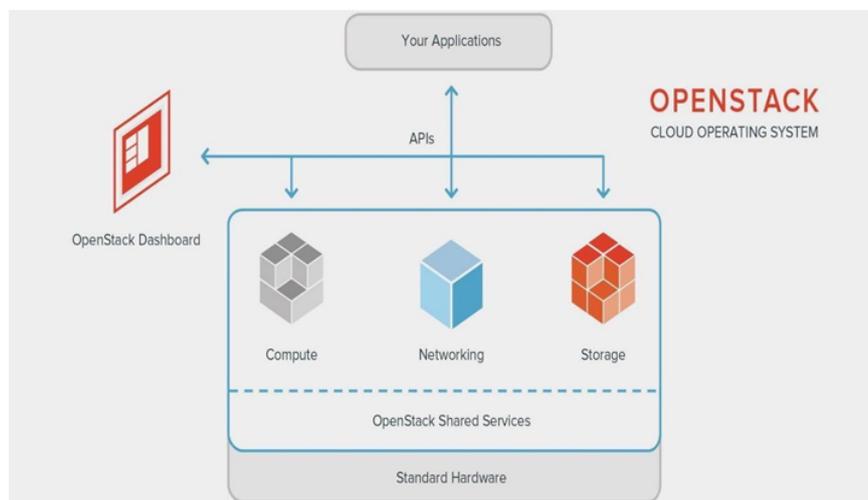


FIGURE II.11 – Architecture logique d'openstack.

II.6.2 Architecture conceptuelle

D'un point de vue conceptuelle, OpenStack se compose de plusieurs parties indépendantes, appelées Services OpenStack. Tous les services peuvent être authentifiés par un service d'identité commun (Keystone). Les services individuels interagissent entre eux par le biais des API publiques, sauf si des commandes d'administrateur privilégiées sont nécessaires.

En interne, les services OpenStack sont composés de plusieurs processus. Tous les services ont au moins un processus d'API qui écoute les requêtes API, les prétraite, et les transmet aux autres parties du service. À l'exception du service d'identité, le travail se fait par des processus distincts.

Pour la communication entre les processus d'un service, un gestionnaire d'allocation de messages AMQP est utilisé. L'état du service est stocké dans une base de données dédiée.

Lors du déploiement et de la configuration du cloud OpenStack, nous pouvons choisir parmi plusieurs gestionnaires d'allocation de messages ou bases de données, comme RabbitMQ, MySQL, MariaDB et SQLite conçues pour une meilleure orchestration entre les modules de la plateforme Openstack.

Le diagramme suivant montre l'architecture de cloud OpenStack la plus courante, bien que ce ne soit pas la seule possible :

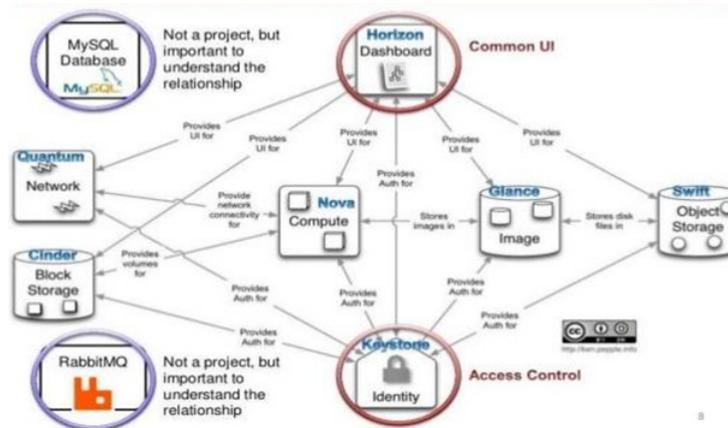


FIGURE II.12 – Architecture conceptuelle d'openstack.

II.6.3 Architecture technique

D'un point de vue technique, OpenStack fonctionne soit en environnement virtuel, soit en environnement Bare-Metal ou les deux en fonction des services installés choisis par l'utilisateur.

II.7 Avantages et Inconvénients

II.7.1 Avantages d'openstack

- Prise en charge de l'hyperviseur : OpenStack prend en charge Xen et KVM, mais une prise en charge limitée de VMware ESX, du serveur Citrix Xen et de Microsoft Hyper-V.
 - Intégration étendue avec les technologies de stockage et de calcul : utilisation de stockage d'objets OpenStack pour fournir un stockage constant afin de gérer les disques locaux sur un cluster de nœuds
-

informatiques. Gestion de divers types d'images de machine via le service d'images OpenStack, tels que OVF, VMDK, VDI, VHD et Raw.

- Fonctions réseau améliorées : OpenStack dispose d'un composant réseau (Neutron), qui peut être directement intégré à OpenFlow, et réalise des logiciels d'extension cloud et multi-locataires de plus haut niveau dans le cloud en adoptant diverses technologies réseau définies ci-dessous. En outre, la structure du réseau OpenStack contient également des services tels que l'équilibrage de charge, les services de détection d'intrusion (IDS) et la technologie de pare-feu. Toutes ces fonctionnalités font d'OpenStack une plate-forme de gestion avec un bon basculement et une bonne flexibilité.
- Grande communauté : OpenStack peut être considéré comme le modèle de contrôle cloud basé sur la pile le plus mature. Il est pris en charge par de grands acteurs du secteur tels que Dell, HP et IBM, et compte de nombreux participants.

II.7.2 Inconvénients d'openstack

- Difficile à configurer et à déployer : les administrateurs ont déclaré que plusieurs composants clés doivent être gérés à partir de différentes consoles de ligne de commande OpenStack dispose de huit composants modulaires cela englobe une architecture légèrement fragmentée.
 - Débogage plus difficile : beaucoup d'interactions entre plusieurs composants donc une architecture très complexe .
 - Les logiciels en open source ne proposent en général pas d'assistance centralisée.
 - Malgré l'aide en ligne et documentation disponible mais la recherche peut être longue et fastidieuse contrairement aux solutions commerciales, où il n'existe pas d'interlocuteur central.
-

II.8 Conclusion

OpenStack est l'un des meilleurs environnements de cloud computing du marché. De nombreuses bases de données de grandes entreprises ont intégré OpenStack dans leurs backends. L'évolutivité linéaire et la simplicité des fonctionnalités open source ont attiré de nombreux clients et passionnés de technologie à se manifester et à contribuer au développement. Cela rend OpenStack plus performant que ces dernières années. Compte tenu de tous les avantages et d'une évolutivité sans fin, OpenStack est considéré comme une solution abordable à long terme.

La mise en place de plateforme Openstack

III.1 Introduction

Nous nous intéressons dans ce chapitre, aux principales étapes d'installation d'OpenStack version Ussuri . Nous allons commencer par la préparation du système, sur lequel sera implémentée cette solution, et ceci par la configuration réseau, ainsi que l'installation des différents paquets requis pour le bon fonctionnement des différents services et composants. Ensuite on passe à l'installation et la configuration de Keystone, Glance, Nova etc. Nous finalisons par la mise en œuvre du Dashboard qui est l'interface graphique et l'outil de gestion d'OpenStack et une présentation d'un exemple de création d'une instance de machine virtuelle et différentes opérations possibles sur cette dernière.

III.2 La mise en place de la plateforme Openstack

III.2.1 Types d'installation d'Openstack

Selon la documentation officielle d'Openstack, il y a plusieurs architectures de déploiement de cette technologie, le choix se fait selon les besoins d'utilisation, les ressources disponibles et les exigences des entreprises et utilisateurs.

Le schéma ci-dessous montre les différentes méthodes existantes :



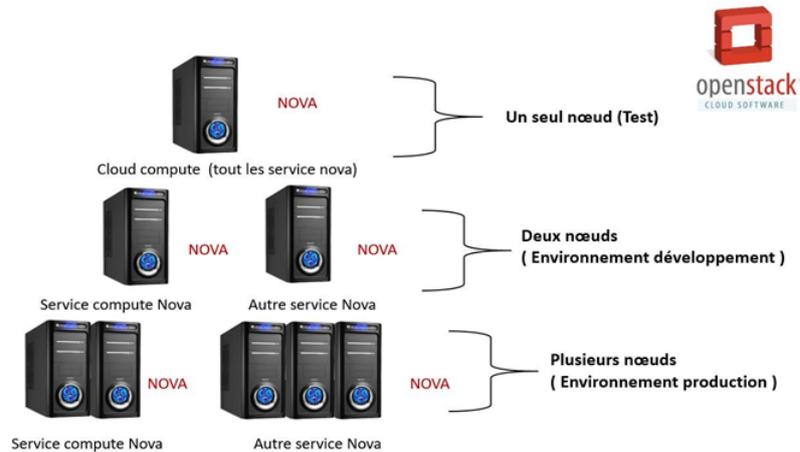


FIGURE III.1 – Les différentes architectures possibles.

Comme on le constate dans la figure, openstack peut s’installer de deux manières différentes :

Single Node (un seul nœud) : dans cette configuration, tous les services sont installés et exécutés sur un système unique. Ce type de déploiement est adapté à des fins d’évaluation. Un tel déploiement n’est cependant pas adapté pour une utilisation dans un environnement de production.

Multi Node (plusieurs nœuds) : les composants d’Openstack seront installés sur des noeuds différents.

NB : Openstack propose aussi plusieurs types d’installation ; via des scripts ou bien via des packages etc.

* Dans notre cas, nous allons installer Openstack sur un seul nœud à travers la méthode MicroStack[22].

III.2.2 Les prérequis

Ressources	Configuration requise
Mémoire	6 GB
CPU	2 CPU (4 cores)
Stockage	100 GB
Réseau	Network card NAT
Hyperviseur	VirtualBox
Machine virtuelle	Ubuntu18.04 LTS (Linux)
Connexion internet	Excellente

TABLE III.1 – Tableau des prérequis.

III.2.3 Outils

Hyperviseur Virtualbox 6.1.14

Au lieu d'installer OpenStack directement sur la machine hôte, le logiciel de virtualisation VirtualBox sera utilisé pour créer une machine virtuelle, sur laquelle seront installés les composants d'OpenStack, et cela pour créer le nombre d'interfaces d' désirées, l'installation d'OpenStack sur une machine virtuelle a aussi pour objectif d'isoler la machine hôte et la machine virtuelle pour ne pas affecter le reste de notre environnement de travail.



FIGURE III.2 – Logo du virtualBox.

Cette machine virtuelle sera configurée avec 5 Go de RAM ,2 CPU et 60 Go de disque dur, sur laquelle sera installée la distribution Ubuntu18.04 LTS.

Système d'exploitation Linux Ubuntu 18.04 LTS

Nous avons choisi d'installer la version 18.04 LTS d'Ubuntu car cette dernière est la plus adéquate à utiliser avec OpenStack et ne souffre pas de problèmes de compatibilité par rapport aux autres versions.



FIGURE III.3 – Logo du Ubuntu.

Version d'Openstack

Nous avons choisis de configurer la version Ussuri d'OpenStack. Notre choix s'est porté sur cette version car c'est la dernière version date de 03/05/2020 jusqu' à la réalisation de ce projet, qui est améliorée de fonctionnalités par rapport aux versions qui la précède.



FIGURE III.4 – Logo du OpenStack Choisi.

III.3 Installation des paquets de base et préparation du système

III.3.1 Configuration de la carte réseau

La version 18.04 (Bionic Beaver) d'Ubuntu utilise netplan pour la gestion des interfaces réseau donc remplace `/etc/network/interfaces` qui est utilisé précédemment.

- Pour pouvoir configurer les cartes réseau nous ouvrons le fichier YAML qui se trouve à l'emplacement `/etc/netplan` comme suit :

```
root@controller:~# cd /etc/netplan
root@controller:/etc/netplan# ls
01-network-manager-all.yaml
root@controller:/etc/netplan# nano 01-network-manager-all.yaml
```

- Ensuite le modifier selon la configuration souhaitée :

```
GNU nano 2.9.3                                01-network-manager-all.yaml
## Let NetworkManager manage all devices on this system
network:
  ethernets:
    enp0s3:
      addresses: [10.0.2.15/24]
      gateway4: 10.0.2.2
      dhcp4: no
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]
  version: 2
```

- Enfin, nous exécutons la commande ci-dessous pour prendre en considération la modification effectuée :

```
root@controller:/etc/netplan# netplan apply
```

Dans notre architecture, nous avons attribué l'adresse IP comme suit : 10.0.2.15/24 pour le nœud appelé "controller". Après la configuration de la carte, il faut configurer la résolution de nom et cela en définissant le nom et l'adresse IP de notre système dans les fichiers `/etc/hosts` et `/etc/hostname`.

- Pour vérifier l'accès à internet, on ping Google.com avec 4 paquets comme suit :

```
root@controller:~# ping -c 4 google.com
PING google.com (172.217.171.206) 56(84) bytes of data.
64 bytes from mrs09s06-in-f14.1e100.net (172.217.171.206): icmp_seq=1 ttl=113 time=53.9 ms
64 bytes from mrs09s06-in-f14.1e100.net (172.217.171.206): icmp_seq=2 ttl=113 time=65.7 ms
64 bytes from mrs09s06-in-f14.1e100.net (172.217.171.206): icmp_seq=3 ttl=113 time=48.7 ms
64 bytes from mrs09s06-in-f14.1e100.net (172.217.171.206): icmp_seq=4 ttl=113 time=46.5 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 46.580/53.756/65.719/7.416 ms
```

III.3.2 Mise à jour du système

- On va mettre à niveau les packages de Ubuntu avec la commande suivante :



```
root@controller:~# apt update && dist-apt upgrade
```

III.3.3 Installation de l’outil de base de données MySQL « mariadb »

- On commence par installer les paquets :

```
root@controller:~# apt install mariadb-server python-pymysql
```

- Ensuite on crée un fichier `/etc/mysql/mariadb.conf.d/99-openstack.cnf` et on ajoute par la suite les configurations suivantes :

```
GNU nano 2.9.3 /etc/mysql/mariadb.conf.d/99-openstack.cnf
[mysqld]
bind-address = 10.0.2.15

default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

- Enfin, on redémarre le service MySQL :
-

```
root@controller:~# service mysql restart
```

III.3.4 Gestion de fil d'attente, installation du service RabbitMQ

Utilisé pour la coordination des opérations et des informations d'état entre les services. Openstack prend en charge plusieurs services de file d'attente de messages, notamment RabbitMQ , Qpid et ZeroMQ . Dans cette installation, on a implémenté RabbitMQ.

- Installation de paquets :

```
root@controller:~# apt install rabbitmq-server
```

- Ajouter l'utilisateur openstack :

```
root@controller:~# rabbitmqctl add_user openstack RABBIT_PASS
```

- Autorisez les utilisateurs à configurer, lire et écrire sur openstack :
-

```
root@controller:~# rabbitmqctl set_permissions openstack ".*" ".*" ".*"  
Setting permissions for user "openstack" in vhost "/"
```

III.3.5 Installer le service d'identité memcached

Memcached est utilisé par les services d'identité, c'est un système d'usage général servant à gérer la mémoire cache distribuée. Il est souvent utilisé pour augmenter la vitesse de réponse des sites web créés à partir de bases de données. Il gère les données et les objets en RAM de façon à réduire le nombre de fois qu'une même donnée stockée dans un périphérique externe.

- Installer les packages du service :

```
root@controller:~# apt install memcached python-memcache
```

- Ouvrir le fichier `/etc/memcached.conf` pour configurer le service à utiliser l'adresse IP de notre système :

```
GNU nano 2.9.3 /etc/memcached.conf  
  
# This parameter is one of the only security measures that memcached has, so make sure  
# it's listening on a firewalled interface.  
-l 10.0.2.15
```

- Redémarrer le service avec la commande :
-

```
root@controller:~# service memcached restart
```

III.3.6 Installation du service ETCD

ETCD (Equipement Terminal de Circuit de Données) est un magasin de données clé-valeur Open Source cohérent et distribué, qui stocke la configuration de systèmes ou clusters de machines distribuées, coordonne leur planification et assure la découverte des services. ETCD facilite et sécurise les mises à jour automatiques, coordonne la planification des tâches affectées aux hôtes et aide à la mise en place d'un réseau pour les conteneurs.

- Installer les paquets :

```
root@controller:~# apt install etcd
```

- Ouvrir le fichier `/etc/default/etcd` et effectuer les modifications suivantes :

```
GNU nano 2.9.3 /etc/default/etcd
ETCD_NAME="controller"
ETCD_DATA_DIR="/var/lib/etcd"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster-01"
ETCD_INITIAL_CLUSTER="controller=http://10.0.2.15:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://10.0.2.15:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://10.0.2.15:2379"
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS="http://10.0.2.15:2379"
```

- Enfin, on redémarre les services :
-

```
root@controller:~# service etcd restart
```

III.3.7 Installation du protocole NTP « chrony »

NTP (Network Time Protocol), utilisé pour synchroniser correctement via un réseau informatique, l'horloge locale d'ordinateurs ainsi que les services entre plusieurs machines si nécessaire, dans ce projet on a installé chrony une implémentation du NTP.

- Installer les paquets :

```
root@controller:~# apt install chrony
```

- Editer le fichier `/etc/chrony/chrony.conf` et ajouter la ligne : `allow 10.0.2.15/24` (la description du sous-réseau) :

```
GNU nano 2.9.3 /etc/chrony/chrony.conf
# Welcome to the chrony configuration file. See chrony.conf(5) for more
# information about usable directives.
allow 10.0.2.15/24
# This will use (up to):
# - 4 sources from ntp.ubuntu.com which some are ipv6 enabled
# - 2 sources from 2.ubuntu.pool.ntp.org which is ipv6 enabled as well
# - 1 source from [01].ubuntu.pool.ntp.org each (ipv4 only atm)
# This means by default, up to 6 dual-stack and up to 2 additional IPv4-only
# sources will be used.
# At the same time it retains some protection against one of the entries being
# down (compare to just using one of the lines). See (LP: #1754358) for the
# discussion.
#
# About using servers from the NTP Pool Project in general see (LP: #104525).
# Approved by Ubuntu Technical Board on 2011-02-08.
# See http://www.pool.ntp.org/join.html for more information.
pool ntp.ubuntu.com iburst maxsources 4
pool 0.ubuntu.pool.ntp.org iburst maxsources 1
pool 1.ubuntu.pool.ntp.org iburst maxsources 1
pool 2.ubuntu.pool.ntp.org iburst maxsources 2
```

- Redémarrer le service NTP :

```
root@controller:~# service chrony restart
```

III.3.8 Installer openstack avec la méthode single node (microstack)

- Télécharger et installer les packages de base d'Openstack avec la commande suivante :

```
root@controller:~# sudo snap install microstack --devmode --beta
2020-11-18T21:45:36+01:00 INFO snap "microstack" has bad plugs or slots: hugepages-control (unknown interface "hugepages-control")
2020-11-18T21:46:00+01:00 INFO snap "microstack" has bad plugs or slots: hugepages-control (unknown interface "hugepages-control")
microstack (beta) ussurì par Canonical ✓ installé
```

- Activer et configurer les packages d'Openstack pour les rendre fonctionnels avec la commande suivante :

```
root@controller:~# sudo microstack init --auto --control
```

NB : cette commande permet également de réinitialiser les composants et services d'Openstack à chaque nouvelle utilisation de la plateforme.

- **Déroulement de l'installation** :
 - Faire de notre système le nœud principal "control" :

```
2020-11-18 21:52:12,556 - microstack_init - INFO - Setting up as a control node.
```

- Ouverture du tableau de bord horizon :

```
2020-11-18 21:52:21,461 - microstack_init - INFO - Opening horizon dashboard up to *
```

- Activation et configuration des services du composant d'identité Keystone : création de projet, préparation du composant et configuration du gestionnaire de clés de python Fernet :

```
2020-11-18 21:52:33,236 - microstack_init - INFO - Configuring Keystone Fernet Keys ...
2020-11-18 21:53:06,052 - microstack_init - INFO - Bootstrapping Keystone ...
2020-11-18 21:53:20,505 - microstack_init - INFO - Creating service project ...
2020-11-18 21:53:27,568 - microstack_init - INFO - Keystone configured!
```

- Configuration du service de modèle de données "placement" et lancement de la fonctionnalité de migrations de données :
-

```
2020-11-18 21:53:27,611 - microstack_init - INFO - Configuring the Placement service...
2020-11-18 21:53:53,018 - microstack_init - INFO - Running Placement DB migrations...
```

- Configuration du composant de calcul Nova ainsi que la fonctionnalité principale de réseautage au moment de migrations d'instance, le cas de plusieurs machines :

```
2020-11-18 21:54:03,467 - microstack_init - INFO - Configuring nova control plane services ...
2020-11-18 21:54:19,623 - microstack_init - INFO - Running Nova API DB migrations (this may take a lot of time)...
2020-11-18 21:54:52,279 - microstack_init - INFO - Running Nova DB migrations (this may take a lot of time)...
```

- Création de volumes (gabarits) par défaut :

```
2020-11-18 21:56:24,721 - microstack_init - INFO - Creating default flavors...
```

- Configuration du composant de réseau Neutron :
-

```
2020-11-18 21:56:57,976 - microstack_init - INFO - Configuring Neutron
```

- Configuration du composant d'images systèmes Glance, et ajout d'image par défaut CirrOS, une distribution Linux minimale conçue pour être utilisée comme image de test sur différentes plateformes de cloud :

```
2020-11-18 21:59:32,256 - microstack_init - INFO - Configuring Glance ...  
Waiting for 10.0.2.15:9292  
2020-11-18 22:00:24,620 - microstack_init - INFO - Adding cirros image ...
```

- Création de règles des groupes de sécurité :

```
2020-11-18 22:00:28,610 - microstack_init - INFO - Creating security group rules ...
```

- Création des services du composant de stockage en bloc Cinder :
-

```
2020-11-18 22:00:41,555 - microstack_init - INFO - Configuring the Cinder services...
2020-11-18 22:01:56,354 - microstack_init - INFO - Running Cinder DB migrations...
```

– Fin d’installation :

```
2020-11-18 22:02:34,471 - microstack_init - INFO - Complete. Marked microstack as initialized!
```

- Maintenant la plateforme Openstack est désormais fonctionnelle sur notre ordinateur et nous pouvons exécuter différentes tâches et opérations selon nos besoins.
 - Récupérer le mot de passe pour accéder au Dashboard d’Openstack :

```
root@controller:~# sudo snap get microstack config.credentials.keystone-password
IEL9004n1EcIeg9WyUcXVDZBlHqLGaw1
```

– Vérifier les composants installés et leur fonctionnement :

```
root@controller-VirtualBox:~# microstack.openstack catalog list
-----
Name      Type      Endpoints
-----
keystone  Identity  microstack
          admin: http://10.0.2.15:5000/v3/
          microstack
          public: http://10.0.2.15:5000/v3/
          microstack
          internal: http://10.0.2.15:5000/v3/
placement placement  microstack
          admin: http://10.0.2.15:8778
          microstack
          public: http://10.0.2.15:8778
          microstack
          internal: http://10.0.2.15:8778
nova      compute   microstack
          public: http://10.0.2.15:8774/v2.1
          microstack
          admin: http://10.0.2.15:8774/v2.1
          microstack
          internal: http://10.0.2.15:8774/v2.1
neutron   network   microstack
          internal: http://10.0.2.15:9696
          microstack
          admin: http://10.0.2.15:9696
          microstack
          public: http://10.0.2.15:9696
cinderv3  volumev3  microstack
          internal: http://10.0.2.15:8776/v3/c1ddcbe180e74568a6d35581191101b2
          microstack
          public: http://10.0.2.15:8776/v3/c1ddcbe180e74568a6d35581191101b2
          microstack
          admin: http://10.0.2.15:8776/v3/c1ddcbe180e74568a6d35581191101b2
glance    image      microstack
          internal: http://10.0.2.15:9292
          microstack
          public: http://10.0.2.15:9292
          microstack
          admin: http://10.0.2.15:9292
cinderv2  volumev2  microstack
          admin: http://10.0.2.15:8776/v2/c1ddcbe180e74568a6d35581191101b2
          microstack
          internal: http://10.0.2.15:8776/v2/c1ddcbe180e74568a6d35581191101b2
          microstack
          public: http://10.0.2.15:8776/v2/c1ddcbe180e74568a6d35581191101b2
```

- A partir du navigateur web, on tape l'adresse IP de notre système ; la page d'authentification suivante doit s'afficher :

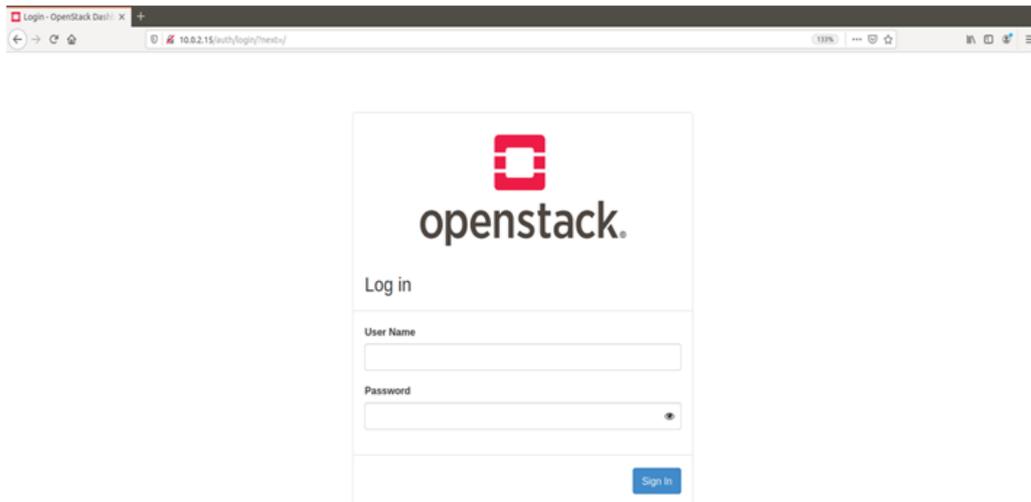


FIGURE III.5 – Tableau de bord OpenStack.

- Ensuite on accède comme étant l'utilisateur par défaut "admin" et on tape également le mot de passe qu'on a récupéré auparavant pour y accéder à la suivante page d'accueil de la plateforme :

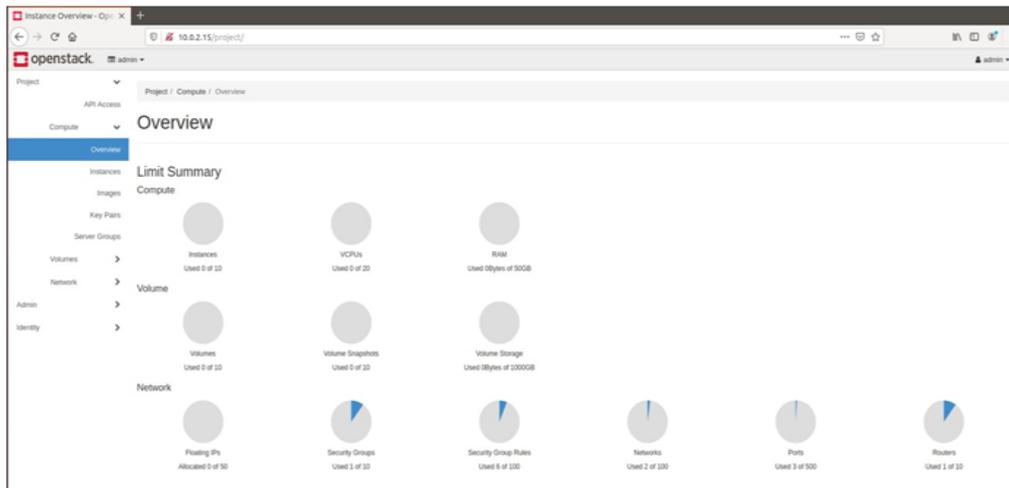


FIGURE III.6 – Première vue du Dashboard.

- A noter que cette vue d'ensemble nous donne l'utilisation de quotas alloués comme le nombre d'instances (10 au max), nombre de cœurs (20 au max), la mémoire vive (50Gb au max), l'espace disque ainsi que les adresses IP publiques. Cependant, cette vue n'est pas représentative des ressources disponibles sur le cloud, il ne s'agit là que des quotas autorisés par l'entreprise de son nouveau nom : Open Infrastructure Foundation.
- Puis, pour pouvoir créer des images et manipuler des volumes on devrait changer la configuration du fichier :
`/var/snap/microstack/common/etc/nginx/snap/nginx.conf`
et fixer la taille limite à **32Go** de mémoire par exemple, en ajoutant la ligne suivante : `client_max_body_size 32768M`; comme suit :

```
user snap_daemon snap_daemon;
worker_processes auto;
pid /var/snap/microstack/common/run/nginx.pid;

events {
    worker_connections 768;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    client_max_body_size 32768M;

    include /snap/microstack/218/usr/conf/mime.types;
    default_type application/octet-stream;
```

- Et on réinitialise les changements avec la commande suivante :

```
root@controller:~# sudo snap restart microstack
```

III.4 Les étapes de configuration du serveur

III.4.1 Exemple concret

- Pour mieux tester le fonctionnement de la nouvelle technologie au sein de l'entreprise X, un administrateur de base de données est désigné responsable de la nouvelle solution cloud, il va donc gérer le projet principal "admin" via la plateforme Openstack.
- En premier lieu il va s'occuper de la gestion de travail en créant les utilisateurs ainsi que leur rôles selon les privilèges qu'il leur donne comme suit :

Utilisateur	Rôle
Secrétaire	Lecteur
Agent d'administration	Lecteur
Analyste1	Membre
Analyste2	Membre

TABLE III.2 – Tableau des privilèges.

- Ensuite, il devrait allouer une adresse IP flottante pour le projet.
- Enfin, configurer et lancer une instance de serveur.

III.4.2 Déroulement de travail sur la plateforme

Gestion de travail

La configuration d'un serveur sur Openstack peut se faire de deux manières différentes :

Via le Dashboard

Création de rôles :

Il va créer deux rôles : Lecteur et membre, en accédant à l'onglet : **identity->Roles – Create Role**, puis le renommer et cliquer sur le bouton "submit" :

FIGURE III.7 – Création du rôle "membre".

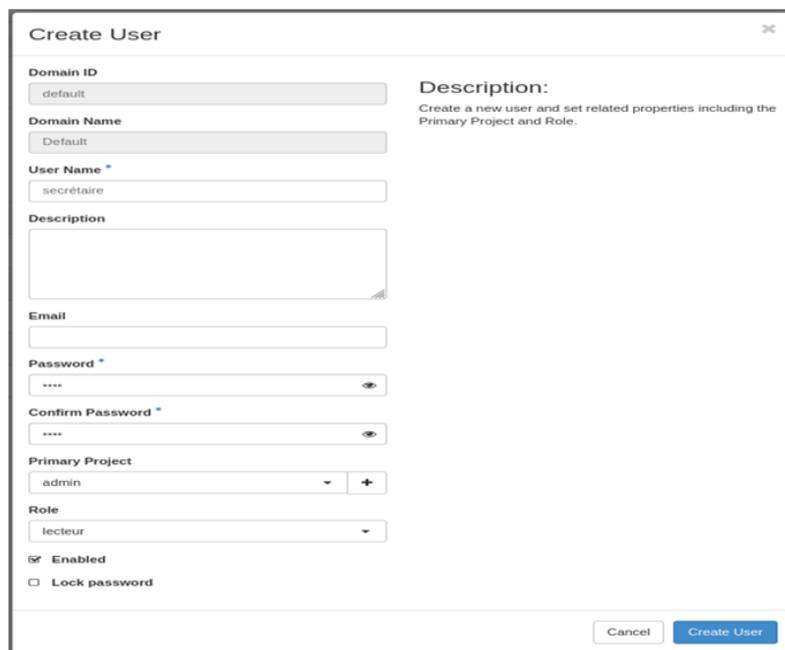


The screenshot shows a 'Create Role' dialog box. At the top, it says 'Create Role' with a close button (X). Below that, there is a 'Name' field with a red asterisk, containing the text 'lecteur' and a green checkmark on the right. At the bottom, there are two buttons: 'Cancel' on the left and 'Submit' on the right.

FIGURE III.8 – Création du rôle "lecteur".

Création d'utilisateurs :

Pour créer les utilisateurs il accède à l'onglet : **identity->Users – Create User**, puis configurer en l'associant à un rôle et un projet et en l'attribuant un mot de passe :



The screenshot shows a 'Create User' dialog box. It has a close button (X) at the top right. On the left side, there are several input fields: 'Domain ID' (default), 'Domain Name' (Default), 'User Name' (secrétaire), 'Description' (empty text area), 'Email' (empty), 'Password' (masked with dots), 'Confirm Password' (masked with dots), 'Primary Project' (admin), and 'Role' (lecteur). On the right side, there is a 'Description:' section with the text 'Create a new user and set related properties including the Primary Project and Role.' At the bottom, there are two checkboxes: 'Enabled' (checked) and 'Lock password' (unchecked). At the very bottom, there are 'Cancel' and 'Create User' buttons.

FIGURE III.9 – Création de l'utilisateur "secrétaire".

The screenshot shows the 'Create User' form in OpenStack. The form is titled 'Create User' and contains several input fields: Domain ID (default), Domain Name (Default), User Name (secretaire), Description (empty), Email (empty), Password (masked with ****), Confirm Password (masked with ****), Primary Project (admin), Role (lecteur), and checkboxes for Enabled (checked) and Lock password (unchecked). A 'Create User' button is at the bottom right.

FIGURE III.10 – Création de l'utilisateur "agent".

The screenshot shows the 'Create User' form in OpenStack. The form is titled 'Create User' and contains several input fields: Domain ID (default), Domain Name (Default), User Name (agent), Description (agent s'administration), Email (empty), Password (masked with ****), Confirm Password (masked with ****), Primary Project (admin), Role (lecteur), and checkboxes for Enabled (checked) and Lock password (unchecked). A 'Create User' button is at the bottom right.

FIGURE III.11 – Création de l'utilisateur "analyste1".

FIGURE III.12 – Création de l'utilisateur "analyste2".

Via le Shell

Création de rôles :

Création des rôles : lecteur et membre :

```

root@controller:~# microstack.openstack role create membre
+-----+-----+
| Field   | Value                                     |
+-----+-----+
| description | None                                     |
| domain_id  | None                                     |
| id        | 09385facbfea4927b6031b9715aa5700      |
| name      | membre                                  |
| options   | {}                                       |
+-----+-----+
    
```

FIGURE III.13 – Création du rôle : membre.

```
root@controller:~# microstack.openstack role create lecteur
+-----+-----+
| Field | Value |
+-----+-----+
| description | None |
| domain_id | None |
| id | fa6c5eb3f73a4f08923ca5d619dabfcd |
| name | lecteur |
| options | {} |
+-----+-----+
```

FIGURE III.14 – Création du rôle : lecteur.

Création d'utilisateurs :

Création des utilisateurs : agent d'administration, secrétaire, analyste1 et analyste2 :

```
root@controller:~# microstack.openstack user create --domain default \
> --password-prompt secrétaire
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | default |
| enabled | True |
| id | 29c86ed53cd44b62b083a55176c359d5 |
| name | secrétaire |
| options | {} |
| password_expires_at | None |
+-----+-----+
```

FIGURE III.15 – Création de l'utilisateur : secrétaire.

```
root@controller:~# microstack.openstack user create --domain default \  
> --password-prompt agent  
User Password:  
Repeat User Password:  
+-----+  
| Field | Value  
+-----+  
| domain_id | default  
| enabled | True  
| id | 4e9197147f79426bad4688d22c665049  
| name | agent  
| options | {}  
| password_expires_at | None  
+-----+
```

FIGURE III.16 – Création de l'utilisateur : agent d'administration.

```
root@controller:~# microstack.openstack user create --domain default \  
> --password-prompt analyste1  
User Password:  
Repeat User Password:  
+-----+  
| Field | Value  
+-----+  
| domain_id | default  
| enabled | True  
| id | 58b8d7f42ba940bc959549b4f05db28d  
| name | analyste1  
| options | {}  
| password_expires_at | None  
+-----+
```

FIGURE III.17 – Création de l'utilisateur : analyste1.

```
root@controller:~# microstack.openstack user create --domain default \  
> --password-prompt analyste2  
User Password:  
Repeat User Password:  
+-----+  
| Field | Value  
+-----+  
| domain_id | default  
| enabled | True  
| id | e21731cd5e2e4f469577cfded3372ad0  
| name | analyste2  
| options | {}  
| password_expires_at | None  
+-----+
```

FIGURE III.18 – Création de l'utilisateur : analyste2.

Attribution de rôles aux utilisateurs et au projet :

```
root@controller:~# microstack.openstack role add --project admin --user analyste1 membre
root@controller:~#
root@controller:~# microstack.openstack role add --project admin --user analyste2 membre
root@controller:~#
root@controller:~# microstack.openstack role add --project admin --user secrétaire lecteur
root@controller:~#
root@controller:~# microstack.openstack role add --project admin --user agent lecteur
root@controller:~#
```

FIGURE III.19 – Attribution des rôles aux utilisateurs et au projet "admin".

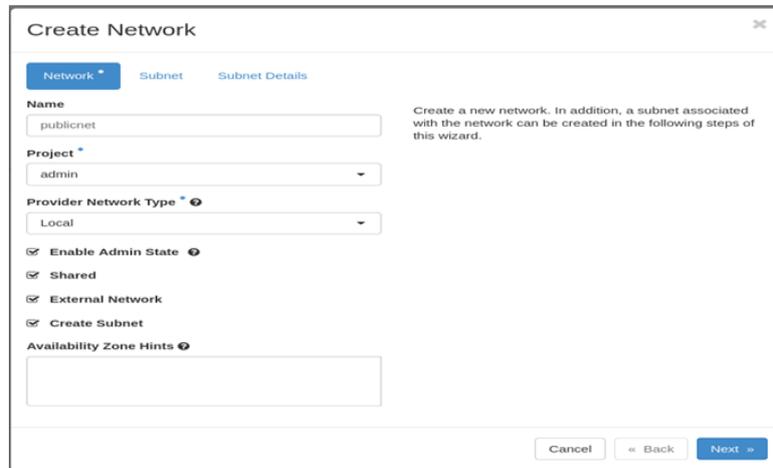
Lancement de l'instance de machine virtuelle

Via le Dashboard

Création de réseaux et routeur :

L'administrateur va maintenant créer deux réseaux l'un est public pour le siège officiel de l'entreprise et l'autre privé pour une annexe de l'entreprise avec différentes adresses IP et tester la connectivité entre les deux parties en les liant avec un routeur virtuel via la plateforme Openstack, la création se fait en suivant le chemin **Admin->Network->Networks- Create Network** et remplir les formulaires selon la configuration souhaitée :

- Création du réseau public "publicnet" et l'affecter au projet "admin" :



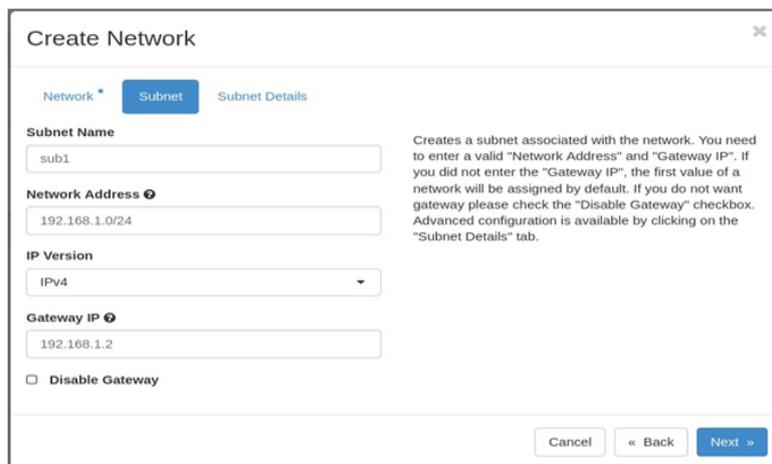
The screenshot shows the 'Create Network' wizard in the OpenStack dashboard. The 'Network' tab is selected. The form contains the following fields and options:

- Name:** publicnet
- Project:** admin
- Provider Network Type:** Local
- Enable Admin State**
- Shared**
- External Network**
- Create Subnet**
- Availability Zone Hints:** (empty text area)

Buttons at the bottom: Cancel, < Back, Next >

FIGURE III.20 – Création du réseau public.

- Création du sous/réseau de nom "sub1" sous l'@ : 192.168.1.0/24 :



The screenshot shows the 'Create Network' wizard in the OpenStack dashboard, now on the 'Subnet' tab. The form contains the following fields and options:

- Subnet Name:** sub1
- Network Address:** 192.168.1.0/24
- IP Version:** IPv4
- Gateway IP:** 192.168.1.2
- Disable Gateway**

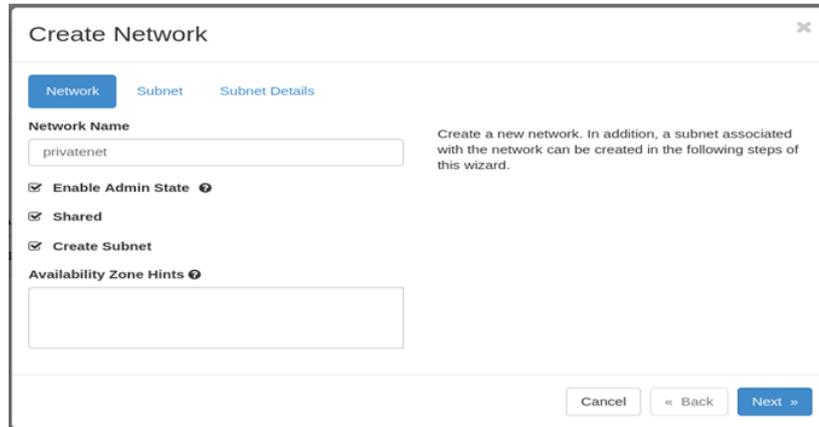
Buttons at the bottom: Cancel, < Back, Next >

FIGURE III.21 – Création du sous/réseau.

- Ensuite cliquer sur [Next] puis [Create].

La création du réseau privé se fait en suivant l'onglet : **Projet->Network->Networks- Create Network** et remplir les formulaires selon la configuration souhaitée :

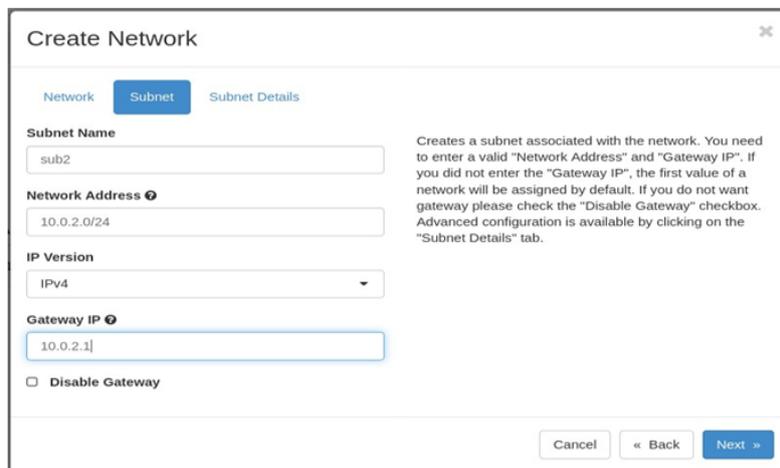
- Création du réseau privé "privatenet" et l'affecter au projet "admin" :



The screenshot shows the 'Create Network' wizard in OpenStack. The 'Network' tab is selected. The 'Network Name' field contains 'privatenet'. The 'Enable Admin State', 'Shared', and 'Create Subnet' checkboxes are checked. The 'Availability Zone Hints' field is empty. The 'Next' button is highlighted.

FIGURE III.22 – Création du réseau privé.

- Création du sous/réseau de nom "sub2" sous l'@ : 10.0.2.0/24 :



The screenshot shows the 'Create Network' wizard in OpenStack, now on the 'Subnet' tab. The 'Subnet Name' field contains 'sub2'. The 'Network Address' field contains '10.0.2.0/24'. The 'IP Version' dropdown is set to 'IPv4'. The 'Gateway IP' field contains '10.0.2.1'. The 'Disable Gateway' checkbox is unchecked. The 'Next' button is highlighted.

FIGURE III.23 – Création du sous/réseau.

Pour faire passer la connectivité entre les deux sièges, l'administrateur va relier les deux réseaux avec un routeur comme suit : aller à l'onglet : **Projet->Network->Routers>Create Router** puis le nommer "**myrouter**" tout

en associant le réseau public de l'entreprise :

- Création du routeur "myrouter" :

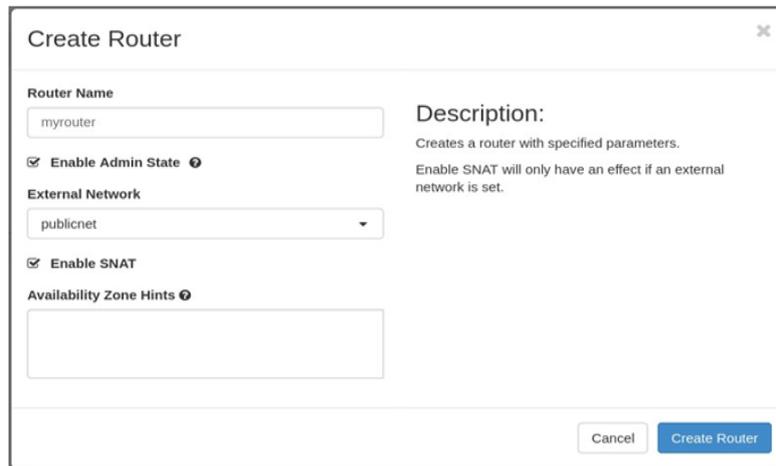


FIGURE III.24 – Création du routeur.

- Une fois le routeur créé il clique sur le nom "myrouter" puis sur le bouton [**Interfaces**] puis [**add Interface**] et il sélectionne le réseau privé de l'annexe comme suit :

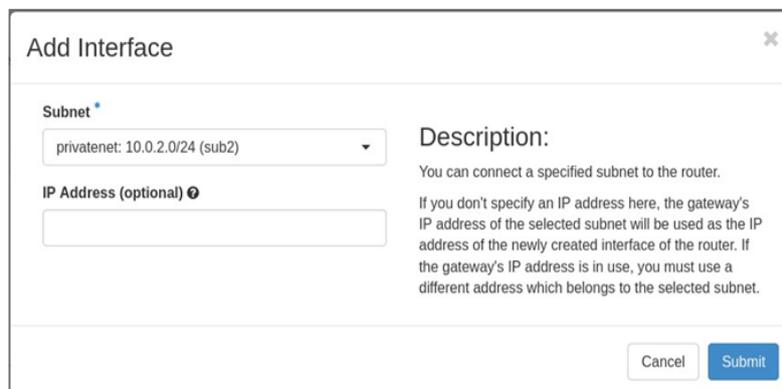


FIGURE III.25 – Ajout d'interface dans le routeur.

- L'administrateur peut maintenant s'assurer de la création de son réseautage dans l'onglet Network Topology :

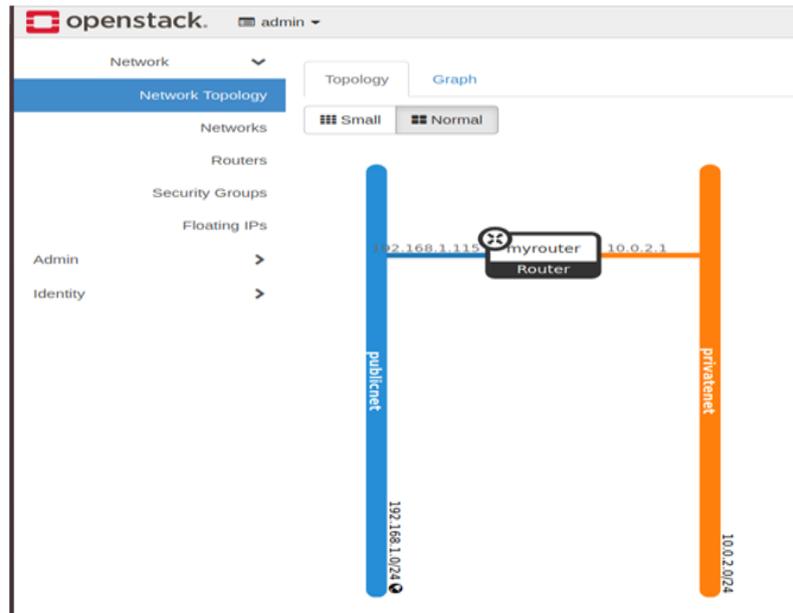


FIGURE III.26 – Vue globale de l'architecture réseau de l'entreprise.

Allocation d'une adresse IP flottante

Une adresse IP flottante permet un accès externe depuis des réseaux extérieurs ou Internet à une machine virtuelle Openstack.

Afin de créer des adresses IP flottantes, l'admin accède à l'onglet **Projet->Network->Floating IPs-Allocate IP To Project** puis sélectionner le réseau public dans l'espace **pool**.

Puis cliquer sur le bouton [**Allocate IP**] :

FIGURE III.27 – Allocation d'une @IP flottante à un pool externe.

Et l'ajout se fera ainsi :

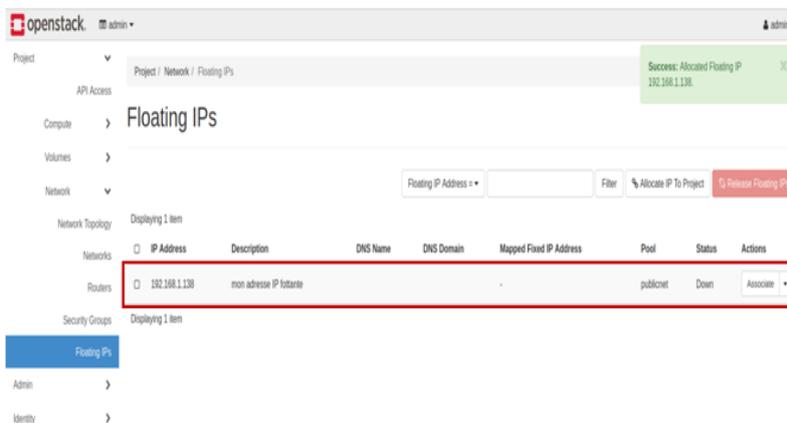


FIGURE III.28 – Confirmation de l'ajout.

Création de clé RSA et de groupes de sécurité

1. Paire de Clés :

- Pour créer une clé de type ssh il suffit d'aller à **Projet->Compute->Key Pairs- Create Key Pair** ensuite la renommer "mykey" et

désigner le type :



FIGURE III.29 – Création de clé SSH.

- Ensuite, télécharger le code de la clé publique sous forme de fichier mykey.pem :



FIGURE III.30 – Téléchargement de la clé.

2. Création de groupe de sécurité et règles :

Le responsable de la plateforme va créer deux groupes de sécurité l'un pour l'équipe technique dont les utilisateurs : `analyste1` et `analyste2` et l'autre pour l'équipe administrative dont les utilisateurs : `secrétaire` et `agent d'administration` et cela pour donner les privilèges qu'il souhaite à travers l'ajout d'autant de règles spécifiques pour chaque groupe.

- Pour se faire l'administrateur accède à **Projet->Network->Security group>Create Security Group** :

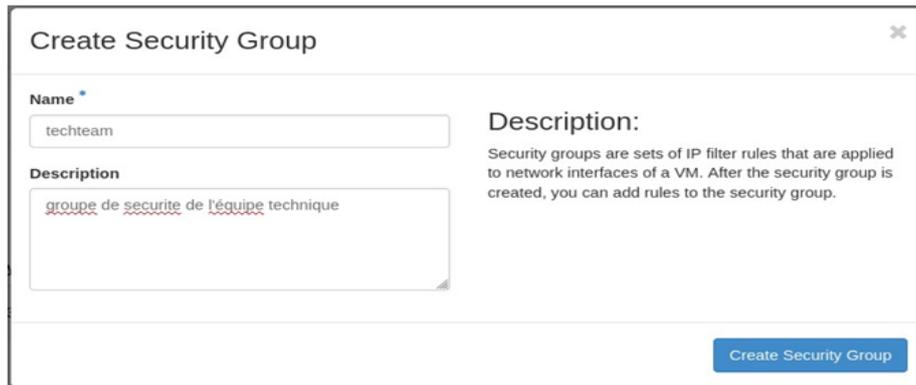
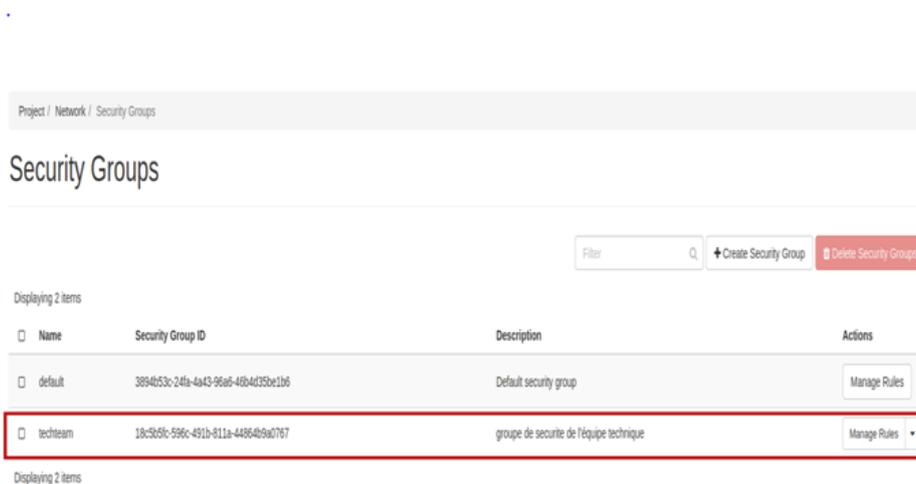


FIGURE III.31 – Création de groupe de sécurité "techteam".

- Puis pour ajouter des règles au groupe, il clique sur le bouton [Manage Rules] :



Name	Security Group ID	Description	Actions
default	3854053c-24fa-4a43-96a6-46b4d359e1b6	Default security group	Manage Rules
techteam	18c5256c-596c-491b-811a-4486409a0767	groupe de securite de l'équipe technique	Manage Rules

FIGURE III.32 – Confirmation de création de groupe.

- Ensuite pour ajouter une règle on clique sur [Add Rule] :

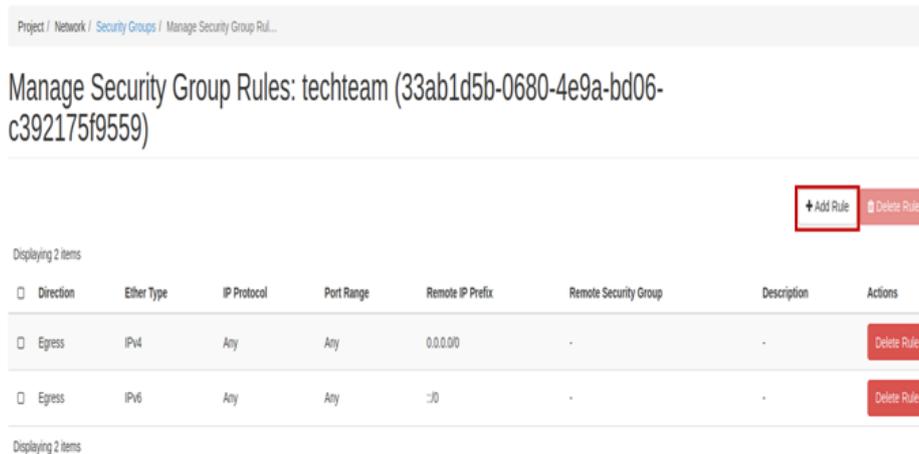


FIGURE III.33 – Espace de modification et ajout de règles.

- Après avoir cliquer sur [Add Rule], il l’a configure selon les permissions qu’il souhaite attribuer pour chaque groupe tout en gardant la sécurité des accès selon la hiérarchie de l’entreprise, puis cliquer sur [Add] :

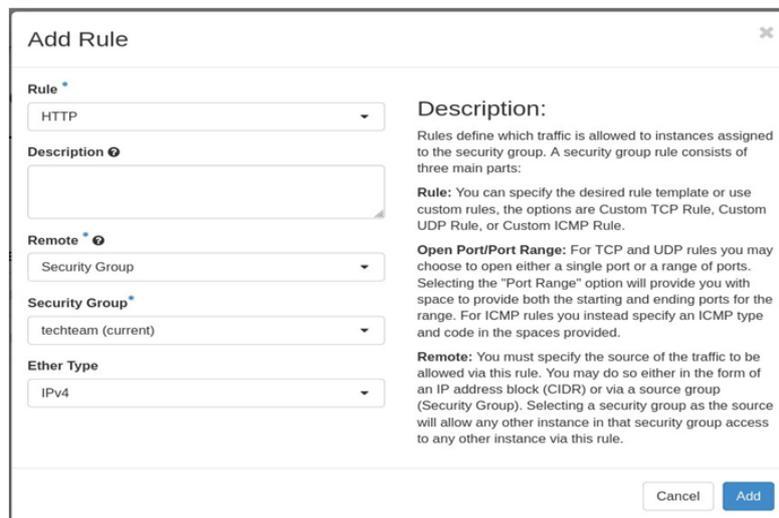
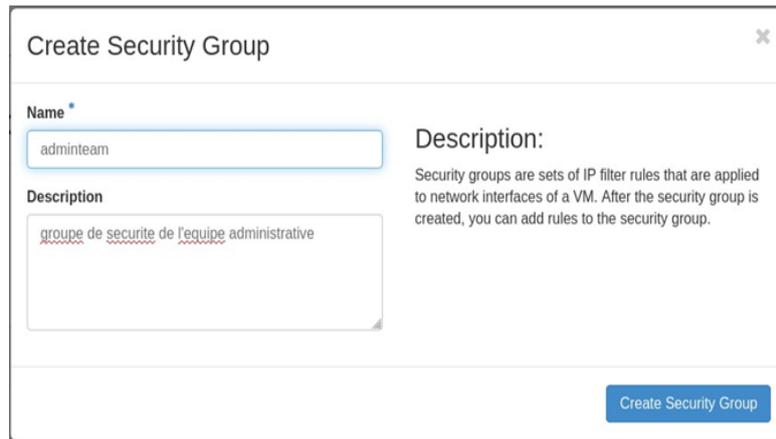


FIGURE III.34 – Ajout d’une règle au groupe.

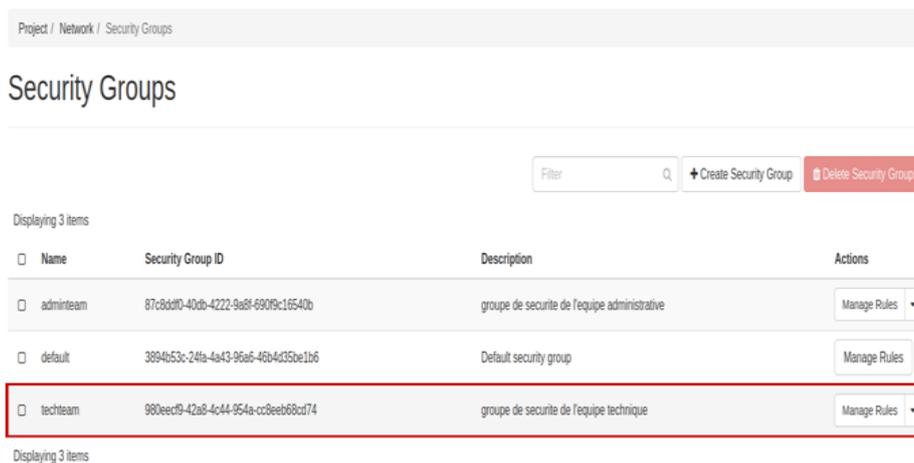
- Pour la deuxième équipe :



The screenshot shows a 'Create Security Group' dialog box. It has a title bar with a close button. Below the title bar, there are two main sections: 'Name' and 'Description'. The 'Name' section has a text input field containing 'adminteam'. The 'Description' section has a text area containing 'groupe de securite de l'equipe administrative'. To the right of the description text area, there is a 'Description:' label and a paragraph of text: 'Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group.' At the bottom right of the dialog, there is a blue button labeled 'Create Security Group'.

FIGURE III.35 – Création de groupe de sécurité "adminteam".

- Puis pour ajouter des règles au groupe il clique sur le bouton [Manage Rules] :



The screenshot shows the 'Security Groups' page in OpenStack. The page has a breadcrumb trail 'Project / Network / Security Groups'. Below the breadcrumb, there is a title 'Security Groups'. There are two buttons: '+ Create Security Group' and 'Delete Security Groups'. Below the buttons, there is a table with 4 columns: Name, Security Group ID, Description, and Actions. The table contains 3 rows. The first row is 'adminteam' with ID '87c8dd0-40db-4222-9a8f-6909c16540b' and description 'groupe de securite de l'equipe administrative'. The second row is 'default' with ID '3894b53c-24fa-4a43-96a6-46b4d35be1b6' and description 'Default security group'. The third row is 'techteam' with ID '980ecc9-42a8-4c44-954a-cc8eeb68cc74' and description 'groupe de securite de l'equipe technique'. The 'techteam' row is highlighted with a red box. Below the table, there is a 'Displaying 3 items' label.

Name	Security Group ID	Description	Actions
adminteam	87c8dd0-40db-4222-9a8f-6909c16540b	groupe de securite de l'equipe administrative	Manage Rules
default	3894b53c-24fa-4a43-96a6-46b4d35be1b6	Default security group	Manage Rules
techteam	980ecc9-42a8-4c44-954a-cc8eeb68cc74	groupe de securite de l'equipe technique	Manage Rules

FIGURE III.36 – Confirmation de création de groupe.

- Ensuite pour ajouter une règle on clique sur [Add Rule] :

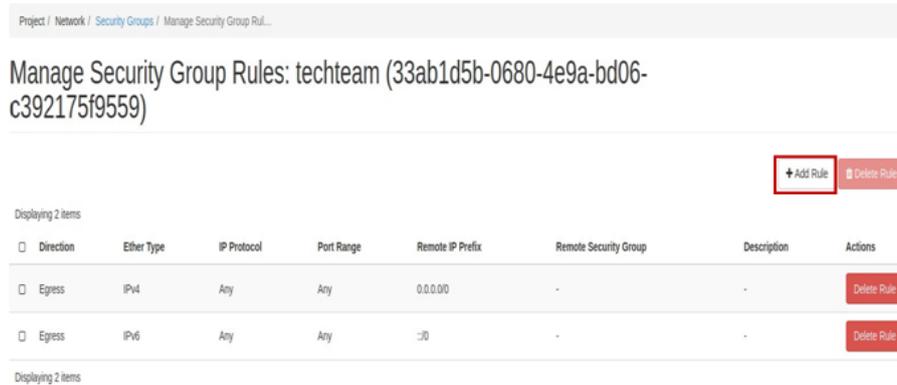


FIGURE III.37 – Espace de modification et ajout de règles.

- Ajout d’une règle pour le deuxième groupe :

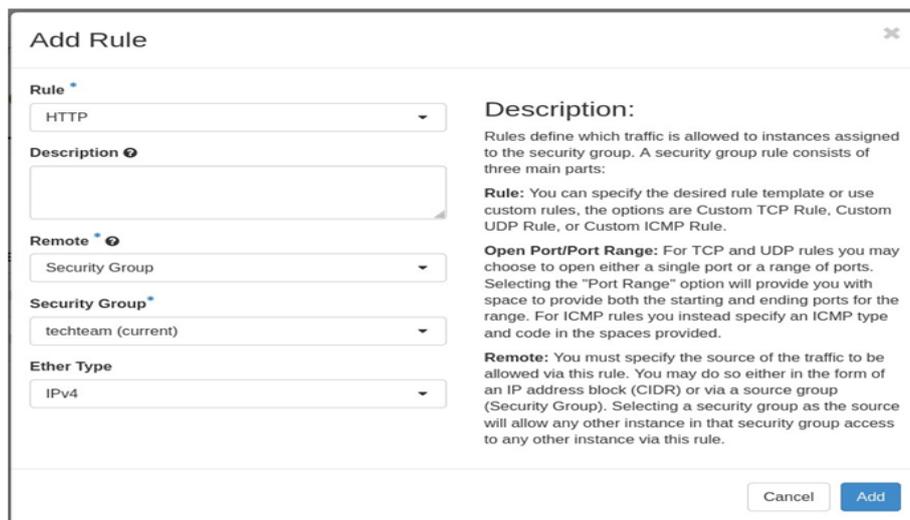


FIGURE III.38 – Ajout d’une règle au groupe.

Création de gabarit (flavor)

A défaut de ressources minimales pour la version test d’openstack, l’administrateur va créer une petite version de flavor qui consomme moins de la mémoire vive, suivant ce chemin **Admin->Flavors–Create Flavor** puis configurer également la RAM à 256Mb ainsi que des volumes minimaux pour

les autres périphériques :

The screenshot shows the 'Create Flavor' dialog box with the following fields and values:

- Name: myflavor
- ID: 6
- VCPUs: 1
- RAM (MB): 256
- Root Disk (GB): 1
- Ephemeral Disk (GB): 0
- Swap Disk (MB): 0
- RX/TX Factor: 1

Buttons: Cancel, Create Flavor

FIGURE III.39 – Création de flavor.

- Ensuite faire associer le gabarit uniquement pour le projet 'admin' et non pas publique :

The screenshot shows the 'Create Flavor' dialog box with the 'Flavor Access' tab active. It displays two lists of projects:

- All Projects: service
- Selected Projects: admin

Buttons: Cancel, Create Flavor

FIGURE III.40 – Attribution du flavor au projet "admin".

- Puis cliquer sur le bouton [Create Flavor] :

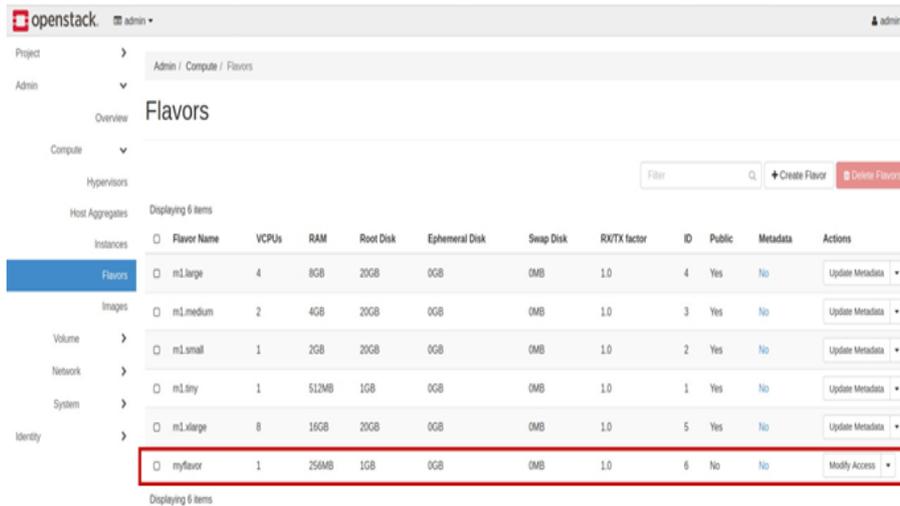


FIGURE III.41 – Confirmation de la création.

Création d'image

Pour la création d'image il faut d'abord télécharger une image système faite pour tester le fonctionnement d'Openstack appelée CirrOs qui est également une distribution Linux minimale.

Ensuite aller dans **Projet->Compute->Images>Create Image** et remplir le formulaire tout en chargeant l'image de CirrOs version 0.5.1 x86 et choisissant le type QCOW2-QEMU Emulator :

Figure III.42 shows the 'Create Image' form in OpenStack. The form is titled 'Create Image' and has a 'Close' button in the top right. It is divided into several sections: 'Image Details' (with a sub-section 'Metadata'), 'Image Source', 'Image Requirements', and 'Image Sharing'. In the 'Image Details' section, 'Image Name' is 'myimage' and 'Image Description' is empty. In 'Image Source', 'File' is 'cirros-0.5.1-x86_64-disk.img', 'Format' is 'QCOW2 - QEMU Emulator', and 'Image Description' is empty. In 'Image Requirements', 'Kernel' is 'Choose an image', 'Architecture' is empty, 'Ramdisk' is 'Choose an image', 'Minimum Disk (GB)' is '0', and 'Minimum RAM (MB)' is '0'. In 'Image Sharing', 'Visibility' has radio buttons for 'Private', 'Shared', 'Community', and 'Public', with 'Shared' selected. 'Protected' has radio buttons for 'Yes' and 'No', with 'No' selected. At the bottom, there are buttons for 'Cancel', '< Back', 'Next >', and 'Create Image'.

FIGURE III.42 – Création d'image système.

Puis cliquer sur [Create Image] :

Figure III.43 shows the OpenStack dashboard after the image creation. A green notification box at the top right says "Success: Image myimage was successfully created." Below the notification, there is a search bar and buttons for "+ Create Image" and "Delete Images". A table lists the created image "myimage" with columns for Owner, Name, Type, Status, Visibility, Protected, Disk Format, and Size. The image "myimage" is listed with Owner "admin", Name "myimage", Type "Image", Status "Saving", Visibility "Public", Protected "Yes", Disk Format "QCOW2", and Size "0 bytes". There is also a "Launch" button next to the image.

Owner	Name	Type	Status	Visibility	Protected	Disk Format	Size
admin	image2	Image	Active	Public	Yes	QCOW2	15.58 MB
admin	myimage	Image	Saving	Public	Yes	QCOW2	0 bytes

FIGURE III.43 – Confirmation de la création d'image.

Lancement de l'instance

Le lancement de l'instance se fait dans **Projet->Compute->Instances**—**Launch Instance** puis compléter plusieurs formulaires en attribuant à l'instance : une image, un gabarit, deux réseaux, deux groupes de sécurités et une paire de clés comme suit :

The screenshot shows the 'Launch Instance' form in OpenStack. The form is titled 'Launch Instance' and has a sidebar with tabs: Details (selected), Source, Flavor, Networks, Network Ports, Security Groups, Key Pair, Configuration, Server Groups, Scheduler Hints, and Metadata. The main area contains fields for Instance Name (myinstance), Description, Availability Zone (nova), and Count (1). A 'Total Instances (10 Max)' gauge shows 100% usage (1 Added, 9 Remaining). Navigation buttons include Cancel, Back, Next, and Launch Instance.

FIGURE III.44 – Création d'instance.

- Puis cliquer sur [Next] et désactiver le bouton [Create New Volume] et sélectionner l'image système créée précédemment en cliquant sur la flèche dessous :

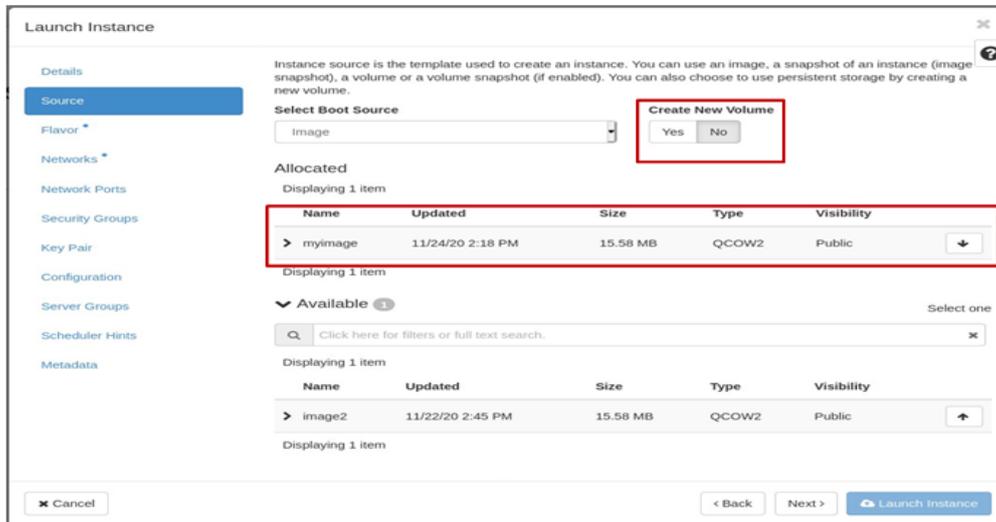


FIGURE III.45 – Attribution d'une image système.

- Puis cliquer sur [Next] et ajouter un un flavor adéquat :

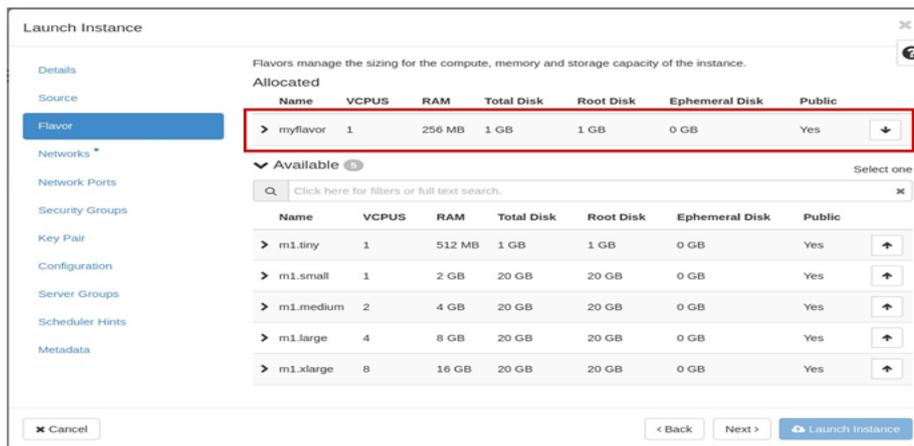


FIGURE III.46 – Attribution d'un gabarit.

- Ensuite attribuer les deux réseaux créés précédemment, l'admin va créer l'instance pour les deux sièges de l'entreprise et donc sur les deux réseaux :

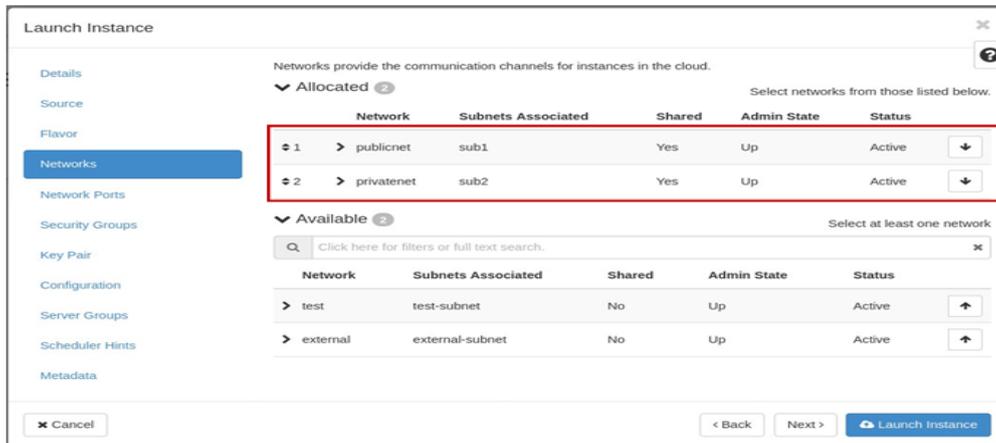


FIGURE III.47 – Attribution des réseaux à l'instance.

- Ensuite les deux groupes de sécurité :

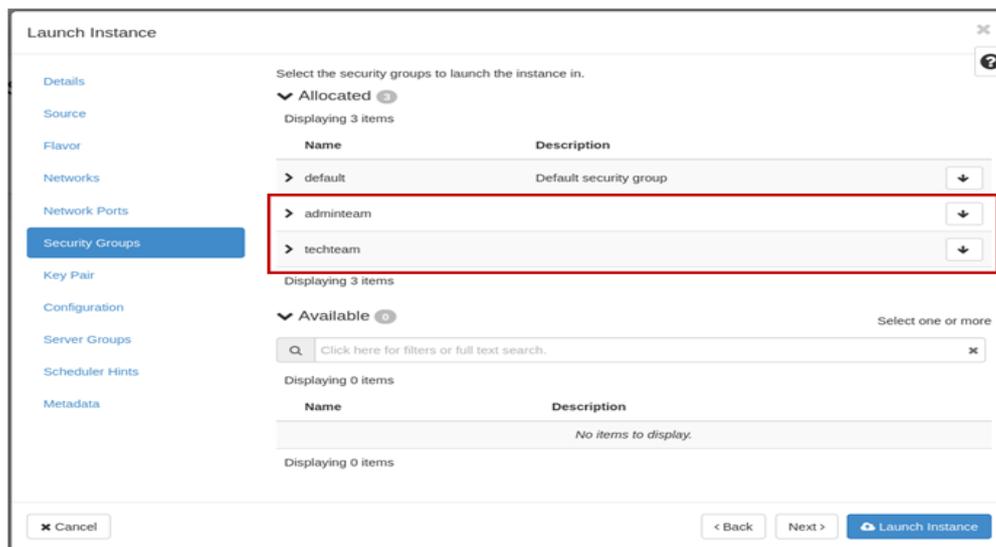


FIGURE III.48 – Attribution de groupes de sécurité.

- Et ajouter également à la fin la clé RSA qu'il a créée :

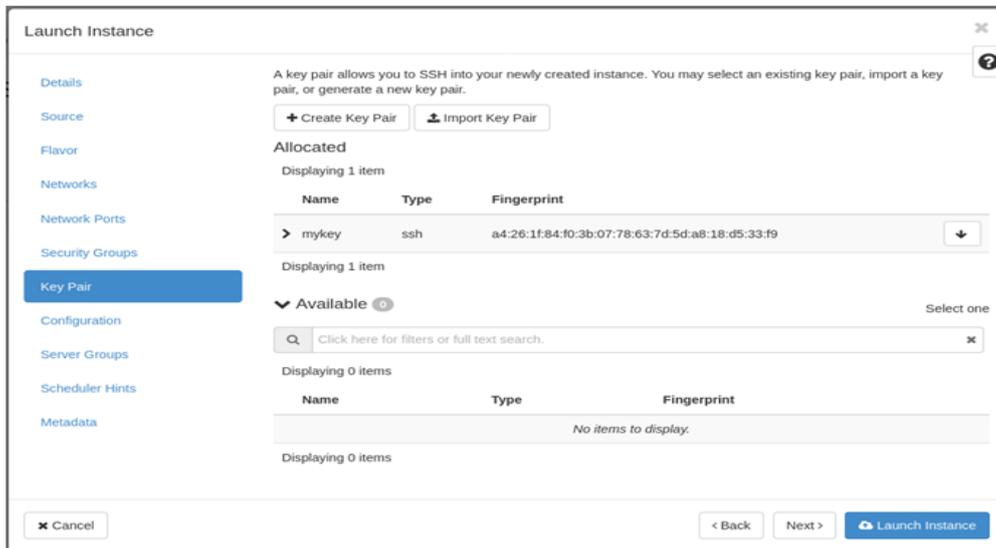


FIGURE III.49 – Attribution de la clé RSA.

- Enfin cliquer sur le bouton [Launch Instance]

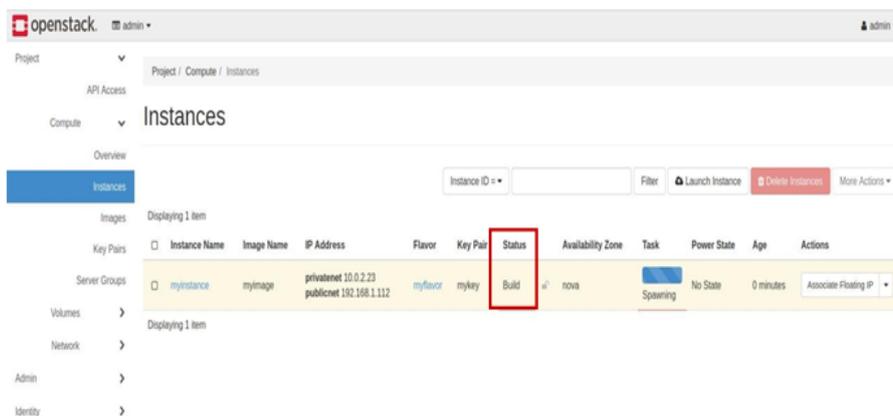


FIGURE III.50 – construction de l'instance.

- Après construction de l'instance son état par de [Build] à [Active] :

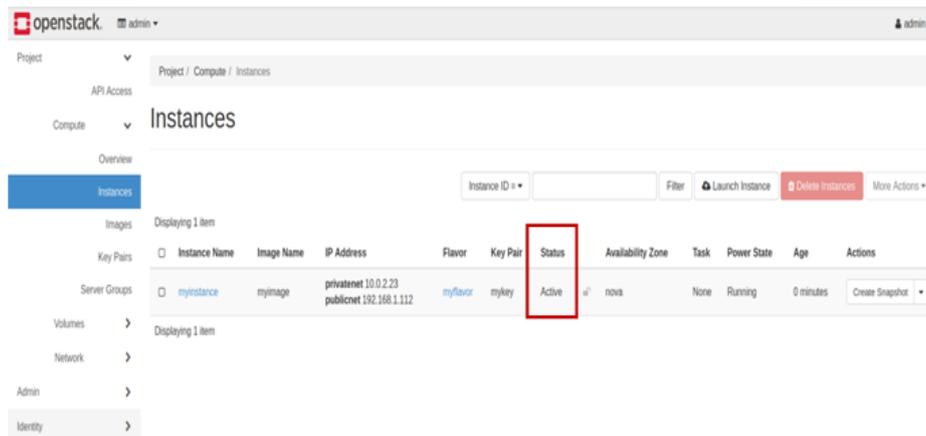


FIGURE III.51 – Création de l’instance avec succès.

Après le lancement de l’instance, l’administrateur va associer l’adresse IP flottante qu’il a créé précédemment à l’instance du réseau privé vu que l’adresse de l’instance créée est déjà dans le domaine du réseau public 192.168.0.0/24 comme suit :

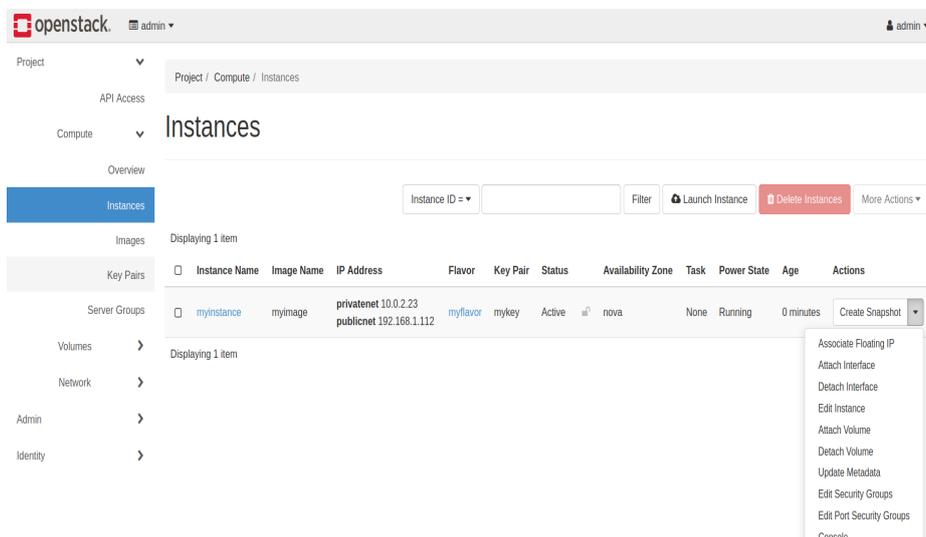


FIGURE III.52 – Configuration de l’instance.

- Après avoir cliquer sur le bouton [Associate Floating IP], il sélectionne l'adresse flottante et l'adresse privée de l'instance :

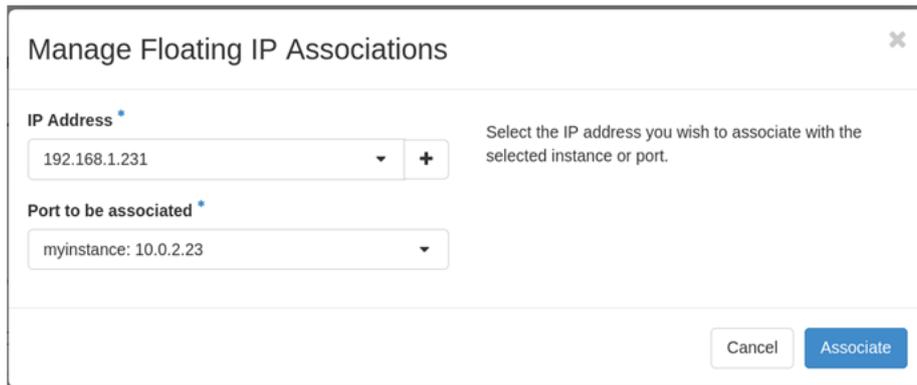


FIGURE III.53 – Association de l'@ flottante à l'instance.

- Ensuite Cliquer sur le bouton [Associate] :

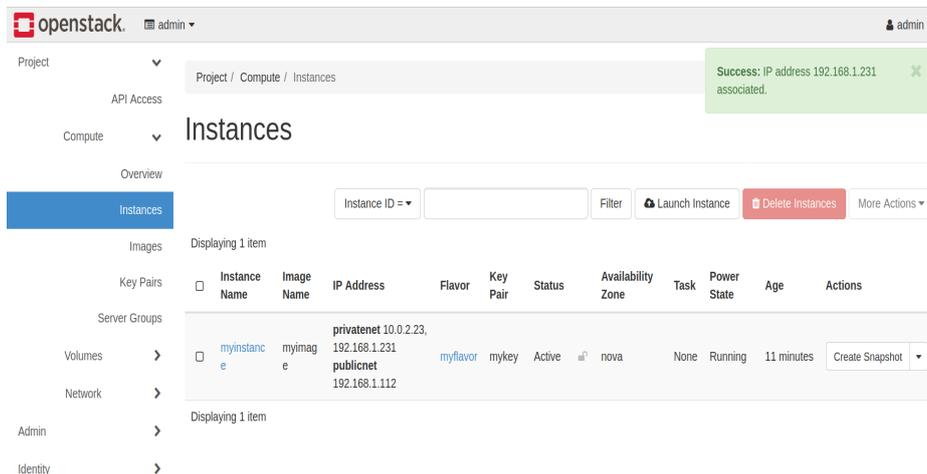


FIGURE III.54 – Association de l'adresse flottante avec succès.

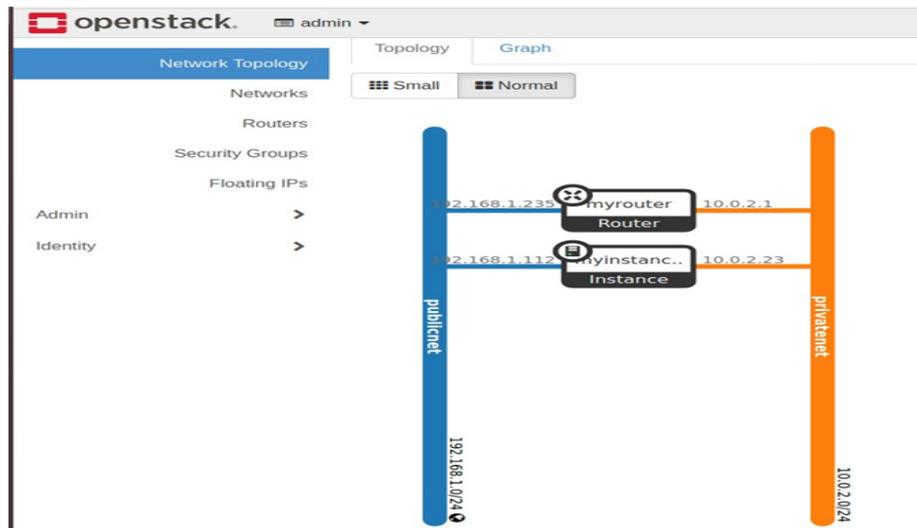


FIGURE III.55 – Architecture globale de l'instance reliée aux deux réseaux.

- Enfin, l'administrateur peut également voir son instance créée en mode CLI :

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions	
myinstance	myimage	privatenet 10.0.2.23, publicnet 192.168.1.231 publicnet 192.168.1.112	myflavor	mykey	Active	us-east-1a	nova	None	Running	19 minutes	Create Snapshot Disassociate Floating IP Attach Interface Detach Interface Edit Instance Attach Volume Detach Volume Update Metadata Edit Security Groups Edit Evt Security Groups Console View Log Rescue Instance Pause Instance

FIGURE III.56 – Passage au mode console.

- En mode CLI suivant, il peut s'identifier à sa machine virtuelle via le nom : cirros et le mot de passe : 'gocubsgo' et pouvoir vérifier l'adresse IP de son serveur par exemple :

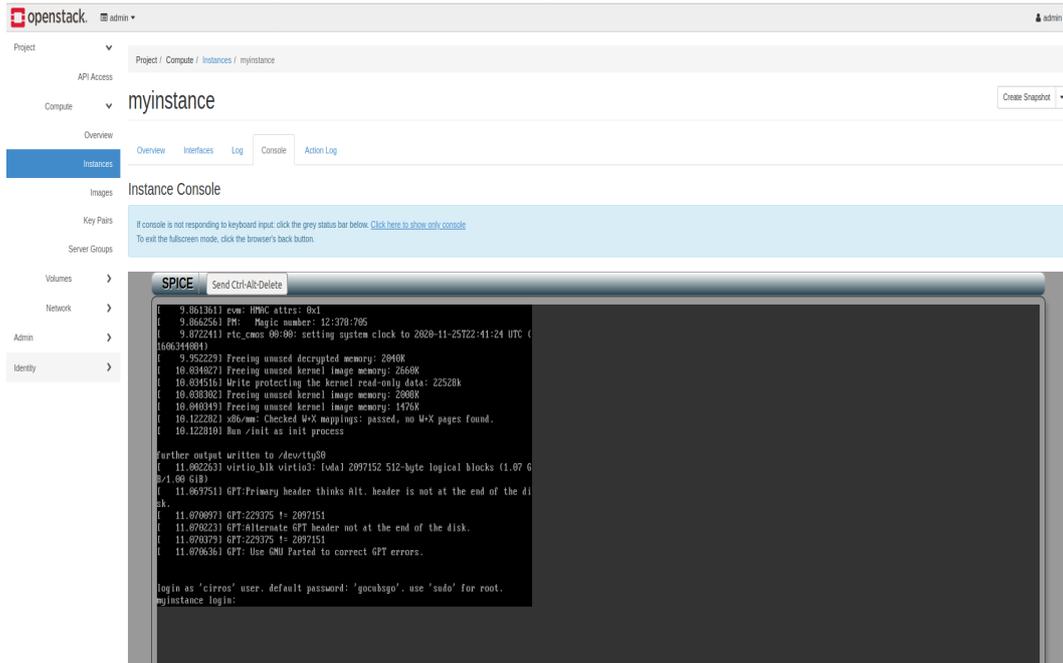


FIGURE III.57 – L'instance en mode console.

- Une fois accéder dans la machine, l'administrateur peut vérifier son adresse IP comme suit :

```
login as 'cirros' user. default password: 'gocubsgo'. use 'sudo' for root.
myinstance login: cirros
Password:
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:77:38:E3
          inet addr:10.0.2.23  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe77:38e3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1442  Metric:1
          RX packets:77 errors:0 dropped:0 overruns:0 frame:0
          TX packets:121 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10064 (9.8 KiB)  TX bytes:10758 (10.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$
```

FIGURE III.58 – Vérification de l'adresse IP de l'instance.

Via le Shell

Création de réseaux et routeur :

- Création du réseau public du domaine d'@ 192.168.1.0/24 :

```

root@controller:~# microstack.openstack network create \
> --project admin \
> --enable \
> --share \
> --external \
> --provider-network-type local \
> publicnet
-----
| Field | Value |
-----
| admin_state_up | UP |
| availability_zone_hints | |
| availability_zones | |
| created_at | 2020-11-26T16:05:22Z |
| description | |
| dns_domain | None |
| id | a4362f01-fdf7-4f36-9bca-8c4b7e7b767 |
| ipv4_address_scope | None |
| ipv6_address_scope | None |
| is_default | False |
| is_vlan_transparent | None |
| location | cloud=' ', project.domain_id=, project.domain_name='default', project.id='2d932182b4b04bb0bed498e5c4030b67', project.name='admin', region_name=' ', zone= |
| mtu | 1500 |
| name | publicnet |
| port_security_enabled | True |
| project_id | 2d932182b4b04bb0bed498e5c4030b67 |
| provider:network_type | local |
| provider:physical_network | None |
| provider:segmentation_id | None |
| qos_policy_id | None |
| revision_number | 1 |
| router:external | External |
| segments | None |
| shared | True |
| status | ACTIVE |
| subnets | |
| tags | |

```

FIGURE III.59 – Création du réseau public.

- Création du s/réseau "mysub1" :

```

root@controller:~# microstack.openstack subnet create \
> --subnet-range 192.168.0.0/24 \
> --dhcp \
> --gateway 192.168.0.1 \
> --network publicnet \
> mysub1
-----
| Field | Value |
-----
| allocation_pools | 192.168.0.2-192.168.0.254 |
| cidr | 192.168.0.0/24 |
| created_at | 2020-11-26T16:15:26Z |
| description | |
| dns_nameservers | |
| dns_publish_fixed_ip | None |
| enable_dhcp | True |
| gateway_ip | 192.168.0.1 |
| host_routes | |
| id | 8df23252-8289-4437-b85b-0bd12a465e61 |
| ip_version | 4 |
| ipv6_address_mode | None |
| ipv6_ra_mode | None |
| location | ccloud='', project.domain_id=, project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name='', zone= |
| name | mysub1 |
| network_id | a4362f01-fdf7-4f36-9bca-8cd4b7e7b767 |
| prefix_length | None |
| project_id | 2d932182b4b04bb0bed488e5c4030b67 |
| revision_number | 0 |
| segment_id | None |
| service_types | |
| subnetpool_id | None |
| tags | |
| updated_at | 2020-11-26T16:15:26Z |
-----

```

FIGURE III.60 – Création du sous/réseau.

- Création du réseau privé du domaine d'@ 10.0.2.0/24 :

```

root@controller:~# microstack.openstack network create \
> --project admin \
> --enable \
> --share \
> --internal \
> --provider-network-type local \
> privatenet
-----
| Field | Value |
-----
| admin_state_up | UP |
| availability_zone_hints | |
| availability_zones | |
| created_at | 2020-11-26T16:20:34Z |
| description | |
| dns_domain | None |
| id | f02f0a85-1c52-4d53-8056-43b31ccce91f |
| ipv4_address_scope | None |
| ipv6_address_scope | None |
| is_default | False |
| is_vlan_transparent | None |
| location | ccloud='', project.domain_id=, project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name='', zone= |
| mtu | 1500 |
| name | privatenet |
| port_security_enabled | True |
| project_id | 2d932182b4b04bb0bed488e5c4030b67 |
| provider:network_type | local |
| provider:physical_network | None |
| provider:segmentation_id | None |
| qos_policy_id | None |
| revision_number | 1 |
| router:external | Internal |
| segments | None |
| shared | True |
| status | ACTIVE |
| subnets | |
| tags | |
-----

```

FIGURE III.61 – Création du réseau privé.

- Création du s/réseau "mysub2" :

```
root@controller:~# microstack.openstack subnet create \  
> --subnet-range 10.0.2.0/24 \  
> --dhcp \  
> --gateway 10.0.2.1 \  
> --network privatenet \  
> mysub2
```

Field	Value
allocation_pools	10.0.2.2-10.0.2.254
cidr	10.0.2.0/24
created_at	2020-11-26T16:26:47Z
description	
dns_nameservers	
dns_publish_fixed_ip	None
enable_dhcp	True
gateway_ip	10.0.2.1
host_routes	
id	59b0bc1e-9635-4e4f-88b3-912d34163944
ip_version	4
ipv6_address_mode	None
ipv6_ra_mode	None
location	ccloud='', project.domain_id=, project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name='', zone=
name	mysub2
network_id	f82f0a85-1c52-4d53-8056-43b31ccce91f
prefix_length	None
project_id	2d932182b4b04bb0bed488e5c4030b67
revision_number	0
segment_id	None
service_types	
subnetpool_id	None
tags	
updated_at	2020-11-26T16:26:47Z

FIGURE III.62 – Création du sous/réseau.

- Création du routeur :

```
root@controller:~# microstack.openstack router create myrouter
```

Field	Value
admin_state_up	UP
availability_zone_hints	
availability_zones	
created_at	2020-11-26T16:45:23Z
description	
external_gateway_info	null
flavor_id	None
id	e709351d-85fd-4f2a-89b9-02c4ef40f578
location	cloud=' ', project.domain_id=' ', project.domain_name='default', project.id='2d932182b4b04bb0bed408e5c4030b67', project.name='admin', region_name=' ', zone=' '
name	myrouter
project_id	2d932182b4b04bb0bed408e5c4030b67
revision_number	1
routes	
status	ACTIVE
tags	
updated_at	2020-11-26T16:45:23Z

FIGURE III.63 – Création du routeur : "myrouter".

- Relier le routeur au réseau externe "publicnet" :

```
root@controller:~# microstack.openstack router set myrouter --external-gateway publicnet
root@controller:~#
```

- Relier le routeur au réseau interne en ajoutant son sous/réseau comme interface :

```
root@controller:~# microstack.openstack router add subnet myrouter mysub2
root@controller:~#
```

- Allocation d'une adresse IP flottante au projet :

```

root@controller:~# microstack.openstack floating ip create \
> --project admin \
> publicnet
-----+-----
| Field | Value |
-----+-----
| created_at | 2020-11-26T17:23:51Z |
| description | |
| dns_domain | |
| dns_name | |
| fixed_ip_address | None |
| floating_ip_address | 192.168.0.164 |
| floating_network_id | a4362f01-fdf7-4f36-9bca-8cd4b7e7b767 |
| id | 8e0ec405-0f98-4fa8-9b63-915788892612 |
| location | Munch({'cloud': '', 'region_name': '', 'zone': None, 'project': Munch({'id': '2d932182 |
| name | 192.168.0.164 |
| port_details | None |
| port_id | None |
| project_id | 2d932182b4b04bb0bed488e5c4030b67 |
| qos_policy_id | None |
| revision_number | 0 |
| router_id | None |
| status | DOWN |
| subnet_id | None |
| tags | [] |
| updated_at | 2020-11-26T17:23:51Z |
-----+-----

```

Création de clé RSA et de groupes de sécurité

- Création de clé RSA de type SSH :

```

root@controller:~# ssh-keygen -t rsa -f mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey.
Your public key has been saved in mykey.pub.
The key fingerprint is:
SHA256:X6h6luEfQGSjqZSbl2UxR3CpU2x+bCiJQYjMMb9xLAK root@controller
The key's randomart image is:
+---[RSA 2048]---+
| oEo o. B++ |
| ++.+. = *= |
| B =o+* o |
| . O.== o.+ |
| = o So.o. |
| . .+ . |
| ..oo |
| . = . |
| .o .. |
+---[SHA256]---+

```

FIGURE III.64 – Génération de clé SSH "mykey."

- Vérification de la création du fichier mykey.pub :

```
root@controller:~# ls
] mykey mykey.pub snap
```

- Récupération du code de la clé "mykey.pub" et copier le code :

```
root@controller:~# cat mykey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAm7F97BprhT7yZKcuMPWl8eLD72TJhWdqaXWlWVb/7qZJMOH/0pnUCn5CU2Q1ZZ40JgGtNTjHGxzdhL6CQqL0XGHe6LGP888vfrKzjU0dPm8sPrC
uyM4m2z+ndZK0F/lkzTo2x+GAQxYETxugV1MqshZ3zlu51S/SL/GnpQ+nVDfZa0KfGQHITuwYag+YLfbwpFgFMBK+glFPrf+CF3WVp5vww9LacrnXE7G4fNiambnN1rEb8oSuADVrdP3QAkAKfmB1V/
RjgryDMy8z1WQa847oelVfeH8Sb/J8S1TQr/pdI+oLfbhmzFncNAcr7B66MB1g10YCPU7CxyLr root@controller
```

- Récupérer l'identifiant de l'agent SSH :

```
root@controller:~# eval "$(ssh-agent)"
Agent pid 17387
```

- Puis aller au Dashboard pour importer la clé et coller le contenu qu'on a récupéré auparavant avec la commande `cat mykey.pub` puis cliquer sur **[Import Public Key]** :



FIGURE III.65 – Importation de la clé "mykey".

- Création des groupes de sécurité :

```
root@controller:~# microstack.openstack security group create techteam
```

Field	Value
created_at	2020-11-26T18:20:10Z
description	techteam
id	3c058723-24d6-4c38-8d5d-acc3dd519567
location	cloud=' ', project.domain_id=' ', project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name=' ', zone=' '
name	techteam
project_id	2d932182b4b04bb0bed488e5c4030b67
revision_number	1
rules	created_at='2020-11-26T18:20:10Z', direction='egress', ethertype='IPv6', id='96706096-5bc8-4983-adb7-b7af157950ae', updated_at='2020-11-26T18:20:10Z' created_at='2020-11-26T18:20:10Z', direction='egress', ethertype='IPv4', id='fff010e5c-b920-49cf-b2d5-07618cff3edf', updated_at='2020-11-26T18:20:10Z'
stateful	True
tags	[]
updated_at	2020-11-26T18:20:10Z

FIGURE III.66 – Création du groupe de sécurité "techteam".

```
root@controller:~# microstack.openstack security group create adminteam
```

Field	Value
created_at	2020-11-26T18:24:55Z
description	adminteam
id	de515858-6db8-4419-af24-e21edc0de48e
location	cloud=' ', project.domain_id=, project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name=' ', zone=
name	adminteam
project_id	2d932182b4b04bb0bed488e5c4030b67
revision_number	1
rules	created_at='2020-11-26T18:24:55Z', direction='egress', ethertype='IPv4', id='e3e10f59-41f7-4898-b0cc-ee4ba3403fe7', updated_at='2020-11-26T18:24:55Z' created_at='2020-11-26T18:24:55Z', direction='egress', ethertype='IPv6', id='ffd25486-4cde-4321-a039-37d93e4c19fb', updated_at='2020-11-26T18:24:55Z'
stateful	True
tags	[]
updated_at	2020-11-26T18:24:55Z

FIGURE III.67 – Création du groupe de sécurité "adminteam".

- Création des règles pour chaque groupe de sécurité :

```
root@controller:~# microstack.openstack security group rule create --proto tcp techteam
```

Field	Value
created_at	2020-11-26T18:23:31Z
description	
direction	ingress
ether_type	IPv4
id	c27d4db0-81d4-4c8a-93e9-598a7b593758
location	cloud=' ', project.domain_id=, project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name=' ', zone=
name	None
port_range_max	None
port_range_min	None
project_id	2d932182b4b04bb0bed488e5c4030b67
protocol	tcp
remote_group_id	None
remote_ip_prefix	0.0.0.0/0
revision_number	0
security_group_id	3c058723-24d6-4c38-8d5d-acc3dd519567
tags	[]
updated_at	2020-11-26T18:23:31Z

FIGURE III.68 – Création d'une règle dans le groupe de sécurité "techteam".

```
root@controller:~# microstack.openstack security group rule create --proto udp adminteam
```

Field	Value
created_at	2020-11-26T18:28:26Z
description	
direction	ingress
ether_type	IPv4
id	3a89766b-042e-4f67-a72e-509fe71e4cee
location	cloud=' ', project.domain_id=' ', project.domain_name='default', project.id='2d932182b4b04bb0bed488e5c4030b67', project.name='admin', region_name=' ', zone=' '
name	None
port_range_max	None
port_range_min	None
project_id	2d932182b4b04bb0bed488e5c4030b67
protocol	udp
remote_group_id	None
remote_ip_prefix	0.0.0.0/0
revision_number	0
security_group_id	de515058-6db8-4419-af24-e21edc0de48e
tags	[]
updated_at	2020-11-26T18:28:26Z

FIGURE III.69 – Création d'une règle dans le groupe de sécurité "adminteam".

Création de gabarit (flavor)

- Création d'un mini flavor :

```

root@controller:~# microstack.openstack flavor create \
> --id 6 \
> --ram 256 \
> --disk 1 \
> --swap 0 \
> --vcpus 1 \
> --rxtx-factor 1 \
> --public \
> myflavor
+-----+-----+
| Field | Value |
+-----+-----+
| OS-FLV-DISABLED:disabled | False |
| OS-FLV-EXT-DATA:ephemeral | 0 |
| disk | 1 |
| id | 6 |
| name | myflavor |
| os-flavor-access:is_public | True |
| properties | |
| ram | 256 |
| rxtx_factor | 1.0 |
| swap | |
| vcpus | 1 |
+-----+-----+

```

FIGURE III.70 – Création d'un gabarit (flavor).

- Création d'image :

Pour la création de l'image système, il faut d'abord télécharger la version minimale de distribution Linux "Cirros" puis lui faire appel dans la commande de création :

```

root@controller:~# microstack.openstack image create \
> --disk-format qcow2 \
> --container-format bare \
> --file /home/controller/Images/cirros-0.5.1-x86_64-disk.img \
> --public \
> myimage

```

```

root@controller:~# microstack.openstack image create \
> --disk-format qcow2 \
> --container-format bare \
> --file /home/controller/images/cirros-0.5.1-x86_64-disk.img \
> --public \
> myimage
+-----+
| Field          | Value
+-----+
| checksum       | 1d3062cd89af34e419f7100277f38b2b
| container_format | bare
| created_at     | 2020-11-27T14:06:31Z
| disk_format    | qcow2
| file           | /v2/images/3d168fa0-afa9-4083-bdd7-c544b7f4a0d9/file
| id             | 3d168fa0-afa9-4083-bdd7-c544b7f4a0d9
| min_disk       | 0
| min_ram        | 0
| name           | myimage
| owner          | 2d932182b4b04bb0bed488e5c4030b67
| properties     | os_hash_algo='sha512', os_hash_value='553d220ed58cfee7dafe003c446a9f197ab5edf8ffc0939d
0277f38b2b', owner_specified.openstack.object='images/myimage', owner_specified.openstack.sha256='c4110030e
| protected      | False
| schema         | /v2/schemas/image
| size           | 16338944
| status         | active
| tags           |
| updated_at     | 2020-11-27T14:06:32Z
| visibility     | public
+-----+

```

FIGURE III.71 – Création de l'image "myimage" avec succès.

Lancement de l'instance

Pour lancer l'instance l'administrateur va faire appel dans sa commande de base de lancement d'instance aux ressources créées auparavant via leurs noms sauf le réseau par son identifiant et vu que l'adresse IP flottante est du domaine du réseau publique 192.168.0.0/24 donc il va créer l'instance sur le réseau privé @10.0.2.0/24 puis associer l'adresse IP flottante à l'instance :

- Récupération de l'identifiant du réseau privé "privatenet" :

```
root@controller:~# microstack.openstack network list
```

ID	Name	Subnets
41663105-c957-48b5-828d-947e3dbe2069	test	afa01678-80f3-47d8-b2c6-cce6c4d5e6eb
a4362f01-fdf7-4f36-9bca-8cd4b7e7b767	publicnet	8df23252-8289-4437-b85b-0bd12a465e61
b4cb47f5-6db2-44ae-a9e1-04837ac25f1e	external	699e3c24-e5ce-4f25-a909-287a16b38f89
f82f0a85-1c52-4d53-8056-43b31ccce91f	privatenet	59bdbc1e-9635-4e4f-88b3-912d34163944

FIGURE III.72 – Aperçu des réseaux.

- La commande qui permet de lancer l'instance du nom "myinstance" :

```
root@controller:~# microstack.openstack server create \
> --flavor myflavor \
> --image myimage \
> --nic net-id=f82f0a85-1c52-4d53-8056-43b31ccce91f \
> --security-group techteam \
> --security-group adminteam \
> --key-name mykey \
> myinstance
```

FIGURE III.73 – Commande de création de l'instance.

- Lancement de l'instance de la machine virtuelle :

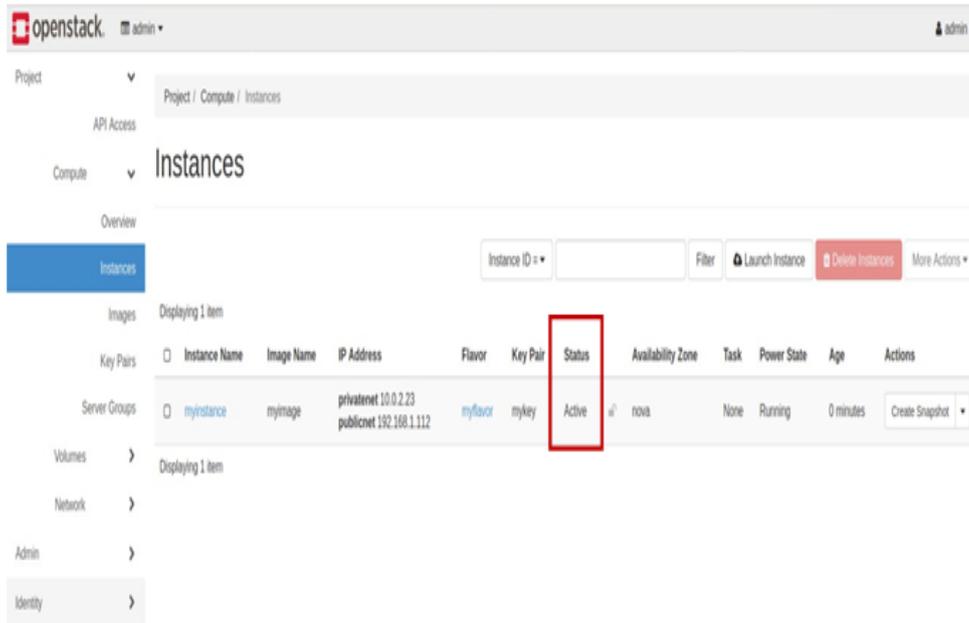


FIGURE III.74 – Création de l’instance avec succès.

- Vérification des serveurs existants :

```

root@controller:~# microstack.openstack server list
+-----+-----+-----+-----+-----+-----+
| ID          | Name      | Status | Networks      | Image  | Flavor  |
+-----+-----+-----+-----+-----+-----+
| 8b1e1f58-4687-4e0c-b454-2cd6e07fbae7 | myinstance | ACTIVE | privatenet=10.0.2.79 | myimage | myflavor |
+-----+-----+-----+-----+-----+-----+
    
```

FIGURE III.75 – Aperçu des instances.

L’instance créée n’as qu’une seule adresse IP, l’administrateur va associer l’adresse IP flottante :

- Récupération de l’adresse IP flottante :

```
root@controller:~# microstack.openstack floating ip list
```

ID	Floating IP Address	Fixed IP Address	Port	Floating Network	Project
8e0ec405-0f98-4fa8-9b63-915788892612	192.168.0.164	None	None	a4362f01-fdf7-4f36-9bca-8cd4b7e7b767	2d932182b4b04bb0bed488e5c4030b67

FIGURE III.76 – Récupération de l’adresse IP flottante : @192.168.0.164..

- Associer l’adresse IP flottante à l’instance :

```
root@controller:~# microstack.openstack server add floating ip myinstance 192.168.0.164
root@controller:~#
```

- Vérifier les adresses attribuées à l’instance qu’elle en a maintenant deux adresses de deux plages différentes :

```
root@controller:~# microstack.openstack server list
```

ID	Name	Status	Networks	Image	Flavor
95286897-4d51-4e62-b18a-47c82d2dba54	myinstance	ACTIVE	privatenet=10.0.2.136, 192.168.0.164	myimage	myflavor

FIGURE III.77 – Vérification d’adressage IP de l’instance.

III.5 Migration de machines virtuelles

Dans le cas d'une panne au niveau d'une machine virtuelle ou changement de serveur afin d'équilibrer la charge l'administrateur doit procéder à une migration de la VM.

Il existe deux types de migration :

III.5.1 Migration à froid

Cette stratégie est la plus basique, il est nécessaire de mettre la machine virtuelle hors tension afin de copier l'état de sa mémoire sur l'hôte cible, Elle est utilisée lorsque l'hôte reçoit plus de requêtes qu'il ne sait en gérer. La décision de migration se fait sur base d'un calcul des requêtes entrantes, de l'utilisation du processeur, des ressources disponibles sur l'hôte principal ainsi que de la consommation énergétique permise[23].

Pour lancer une migration à froid il faut que les nœuds source et destination puissent effectués une connexion secrète SSH. Avant de lancer une migration, il faut en premier lieu connaitre les instances existantes.

- La figure ci-dessous donne une liste de tout les clusters :

```
root@controller:~# microstack.openstack server list
+-----+-----+-----+-----+-----+-----+
| ID           | Name      | Status | Networks          | Image  | Flavor  |
+-----+-----+-----+-----+-----+-----+
| 8b1e1f58-4687-4e0c-b454-2cd6e07fbae7 | myinstance | ACTIVE | privatenet=10.0.2.79 | myimage | myflavor |
+-----+-----+-----+-----+-----+-----+
```

FIGURE III.78 – Liste des instances.

- Ensuite il affiche, l'hôte sur lequel réside son instance :

```

root@controller:~# microstack.openstack server show 95286897-4d51-4e62-b18a-47c82d2dba54
-----
| Field | Value |
-----
| OS-DCF:diskConfig | AUTO |
| OS-EXT-AZ:availability_zone | nova |
| OS-EXT-SRV-ATTR:host | controller |
| OS-EXT-SRV-ATTR:hypervisor_hostname | controller |
| OS-EXT-SRV-ATTR:instance_name | instance-00000008 |
| OS-EXT-STS:power_state | Shutdown |
| OS-EXT-STS:task_state | None |
| OS-EXT-STS:vm_state | stopped |
| OS-SRV-USG:launched_at | 2020-11-27T21:11:25.000000 |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | privatenet=10.0.2.136, 192.168.0.164 |
| config_drive | |
| created | 2020-11-27T21:11:11Z |
| flavor | myFlavor (6) |
| hostid | 407365683d0899df1c66e51551fd60af40a764316af1dba0a34e92ae |
| id | 95286897-4d51-4e62-b18a-47c82d2dba54 |
| image | myImage (3d168fa0-afa9-4083-bdd7-c544b7f4a0d9) |
| key_name | mykey |
| name | myInstance |
| project_id | 2d932182b4b04bb0bed488e5c4030b67 |
| properties | |
| security_groups | name='techteam'  
name='default'  
name='adminTEAM' |
| status | SHUTOFF |
| updated | 2020-11-28T09:56:35Z |
| user_id | d7848e30fb4947f0aa23680b8d28828e |
| volumes_attached | |
-----

```

FIGURE III.79 – Aperçu d'informations de l'instance à migrer.

Comme cette figure ci-dessus le montre, l'instance réside sur le nœud principal "controller".

Dans le cas de plusieurs machines, on suppose que l'hôte "compute2" existe dans le siège annexe de l'entreprise donc l'administrateur peut lancer la migration de son instance vers l'hôte "compute2" via la commande suivante :

```

root@controller:~# microstack.openstack server migrate 95286897-4d51-4e62-b18a-47c82d2dba54
root@controller:~#

```

FIGURE III.80 – Commande de la migration à froid.

L'instance maintenant est migrée sur le nœud destination "compute2"; la vérification de la réussite de l'opération peut se faire via le nœud principal. Donc le résultat sera de ce genre : que l'instance a bien changé d'hôte et son état est "active".

```

root@controller:~# openstack server show 8dc3411c-1ca2-448e-bdb3-4090d87d621b
+-----+-----+
| Field | Value |
+-----+-----+
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | nova |
| OS-EXT-SRV-ATTR:host | compute2 |
| OS-EXT-SRV-ATTR:hypervisor_hostname | compute2 |
| OS-EXT-SRV-ATTR:instance_name | instance-0000002a |
| OS-EXT-STS:power_state | Running |
| OS-EXT-STS:task_state | None |
| OS-EXT-STS:vm_state | active |
| OS-SRV-USG:launched_at | 2019-08-31T17:42:36.000000 |
| OS-SRV-USG:terminated_at | None |
| accessIPv4 | |
| accessIPv6 | |
| addresses | provider=10.0.0.102 |
| config_drive | |
| created | 2019-08-31T17:34:40Z |
| flavor | m1.nano (0) |
| hostId | 7c1385eda2303f8c63e0e7c0bf6b559133233b4d0dbc1fe5b85e4f97 |
| id | 8dc3411c-1ca2-448e-bdb3-4090d87d621b |
| image | cirros (4a6cf4da-8c01-430a-86ab-c0c05f8471a2) |
| key_name | mykey |
| name | instance1 |
| progress | 0 |
| project_id | bec9bd23182447399f3b6a013d613031 |
| properties | |
| security_groups | name= 'default' |
| status | ACTIVE |
| updated | 2019-08-31T17:46:48Z |
| user_id | 08c43dcba2024b6aa5cd021e908def71 |
| volumes_attached | |
+-----+-----+

```

FIGURE III.81 – Confirmation de déplacement de l’instance.

III.5.2 Migration à chaud

La migration à chaud est une stratégie utilisée pour pallier les problèmes de la migration à froid. Elle permet à un hyperviseur de transférer, au travers d’une connexion TCP, une machine virtuelle d’un hôte à un autre sans être obligé de l’arrêter.

Supposant qu’il existe plusieurs instances de machines dans ce cas, l’administrateur lance la migration via le Dashboard de controller tout en gardant les machines fonctionnelles (lancées). Il choisit l’instance (VM) à migrer dans ce cas on va migrer l’instance test-1, puis dans la colonne « Actions » il choi-

sit « Live migrate instance » comme le montre la figure suivante :

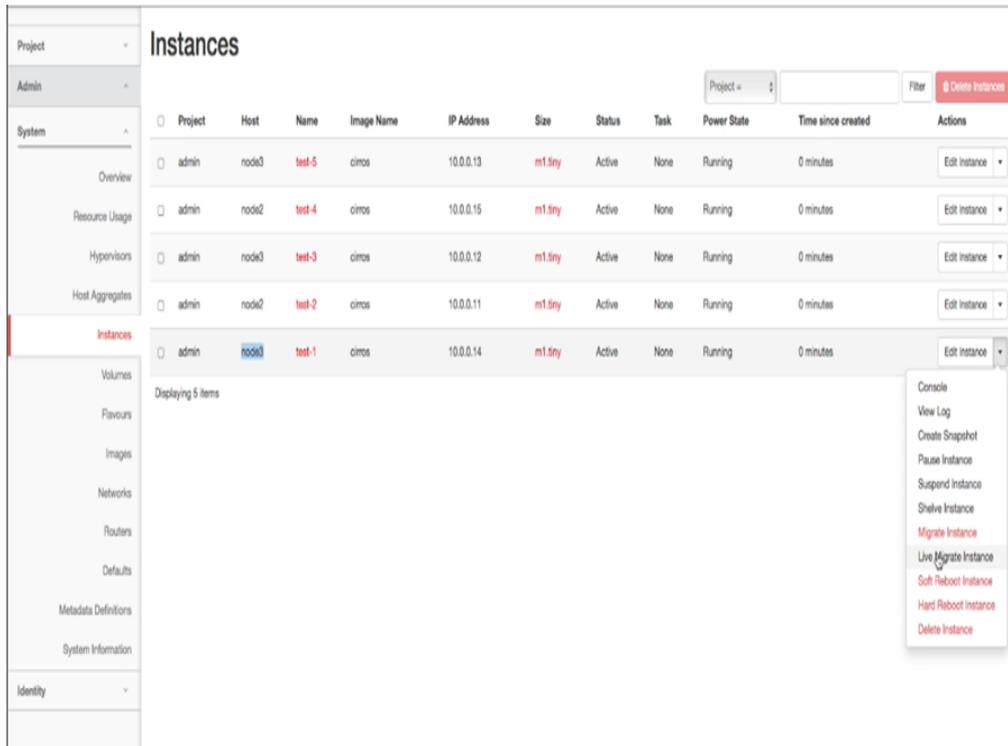


FIGURE III.82 – Liste des instances et options possibles.

Ensuite cette fenêtre (figure) apparaisse pour choisir vers quelle machine migrer dans ce cas la migration est faite de l'hôte "node3" vers l'hôte "node2".

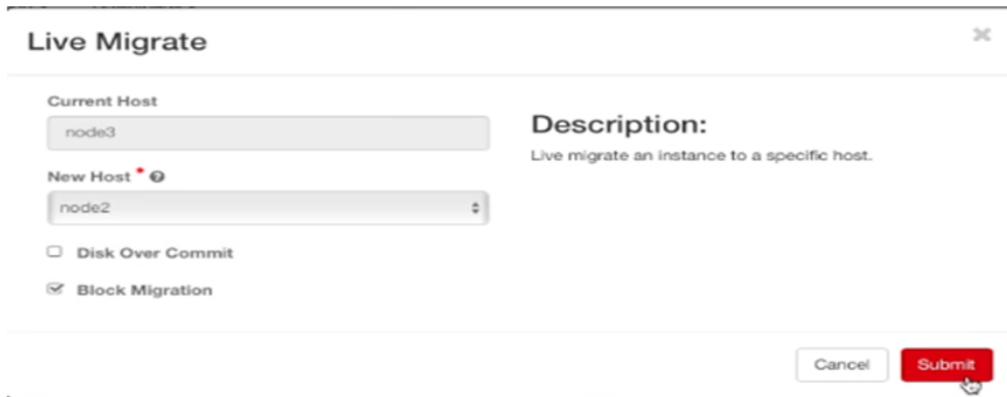


FIGURE III.83 – Choix de l’hôte courant et l’hôte destinataire.

Processus de la migration :



Fin de la migration :



III.6 Conclusion

Dans ce chapitre, nous avons mis en place notre plateforme pour construire un cloud privé au sein d’une entreprise, à travers cette installation nous avons pu dégager l’intérêt du cloud, et ce que peut nous apporter de bénéfices et augmentation de performances.

Mais la mise en place de cette technologie nécessite un matériel puissant et supporte la virtualisation avec un bon accès internet.

Conclusion et perspectives

Le Cloud Computing est une technologie émergente et innovante, qui permet d'offrir des ressources informatiques sous forme de services à la demande accessibles de n'importe où, n'importe quand et par n'importe qui. La fiabilité et la robustesse de ces services deviennent des éléments de premières importances.

Au cours de ce mémoire, nous avons fait une étude et mise en place d'une solution open source du Cloud Computing pour une entreprise, on a commencé par donner les définitions de base nécessaires à la compréhension du Cloud, son architecture et ses services (IaaS,PaaS,SaaS) ainsi que ses différents types de déploiement (privé, public, hybride) ensuite on a présenté le fondement du cloud qui est la virtualisation.

Et on a présenté et détaillé les différentes solutions permettant de mettre en place un Cloud ceci nous a permis d'avoir une idée précise et complète sur les solutions disponibles du Cloud et surtout de choisir celle qui nous convient le mieux.

Dans notre projet, Openstack est la plus répondue à nos critères, car son but principal et de construire un Cloud privé pur. Pour mettre en œuvre cette solution, on a procédé à la création d'un environnement Cloud sur Openstack, cette étape était la plus longue et la plus complexe à cause du manque de ressources et surtout nécessite une grande connaissance en réseau et en lignes de commandes linux.

Tous ces problèmes nous ont montrés la complexité d'utiliser une telle plateforme, et leurs résolutions nous a souvent retardés et pénalisés mais nous a amené à chercher et essayer de nombreuses méthodes et à expérimenter le travail d'un informaticien digne de ce nom, qui propose une solution à tout problème quelque soit sa complexité.

Ce projet a été pour nous une chance et une formidable opportunité de découvrir un environnement informatique nouveau, complexe et vaste, ce qui nous a permis d'acquérir de l'expérience en administration systèmes et réseaux et d'approfondir nos connaissances dans le domaine de la virtualisation et du Cloud Computing.

En perspectives, nous proposons la mise en place de la solution de Cloud Computing Openstack sur un environnement plus large à savoir sur un ser-

veur pour l'université Mouloud Mammeri de Tizi-Ouzou, qui permettra aux étudiants de profiter des avantages d'un Cloud et de créer leur propres machines virtuelles selon des règles dictées par l'administrateur.

Avec ça, l'étudiant peut continuer ses modifications sur un TP même en dehors des murs de l'université, il ne se soucie pas de la perte de ses données et rend ses ressources disponibles de partout. Cela donnera lieu à l'installation et configuration des composants : Quantum, Swift et Cinder qui sont nécessaire dans ce cas pour assurer le bon fonctionnement d'un tel environnement et prochainement l'installation sur un environnement multi nœuds qui va permettre de voir de prêt comment remédier à une surcharge de demandes (migration des VMs) et de tester toutes ses fonctionnalités.

Bibliographie

- [1] [Online]. Available : <https://blog.advancia-itsystem.com/les-chiffres-du-cloud-computing/.X8Nt6c6gIU>
- [2] *Mcgrawhill-cloud computing.*
- [3] J. Maher, *mémoire de master : Mise en place d'une plateforme de virtualisation et déploiement d'une solution Cloud privé open source*, 2015.
- [4] Site officiel openstack. [Online]. Available : <https://www.openstack.org/>
- [5] [Online]. Available : <https://www.ibm.com/fr-fr/cloud/learn/benefits-of-cloud-computing>
- [6] [Online]. Available : https://www.cisco.com/c/fr_fr/solutions/cloud-systemsmanagement/intelligentautomation-cloud/index.html
- [7] N. GREVET, *Le Cloud Computing : Evolution ou Révolution.*
- [8] *L'évolution maîtrisée vers le IaaS/PaaS*, Novembre.
- [9] [Online]. Available : <http://www.xen.org/>
- [10] [Online]. Available : <http://www.arumtec.net/fr/outils-virtualisation/outils-devirtualisation/vmware-vsphere4.1/>
- [11] [Online]. Available : <http://www.microsoft.com/hyper-v-server/en/us/default.aspx>
- [12] [Online]. Available : <https://www.salesforce.com/fr/>
- [13] [Online]. Available : <https://aws.amazon.com/fr/>
- [14] [Online]. Available : <https://cloud.google.com/appengine?hl=fr>
- [15] [Online]. Available : <https://azure.microsoft.com/fr-fr/>

- [16] [Online]. Available : <https://aws.amazon.com/fr/ec2/>
 - [17] [Online]. Available : <https://www.alibabacloud.com/fr>
 - [18] [Online]. Available : <https://www.ibm.com/cloud>
 - [19] [Online]. Available : <https://www.eucalyptus.cloud/>
 - [20] [Online]. Available : <https://opennebula.io/>
 - [21] [Online]. Available : <https://fr.wikipedia.org/wiki/OpenStackHistorique>
 - [22] [Online]. Available : <https://ubuntu.com/tutorials/microstack-get-started>
 - [23] K. Abedelaziz, *vers une gestion de migrations des machines virtuelles pour améliorer la qualité de service dans le Cloud Computing*, thèse de doctorat, 2015.
-