

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

Université Saad Dahleb Blida1

Faculté des sciences.

Département informatique.



Mémoire de fin d'études en vue de l'obtention du Diplôme

MASTER INFORMATIQUE

Option : Systèmes informatiques et Réseaux

Thème

**Étude et conception d'un Cloud privé basé sur OpenStack et
migration d'un Environnement virtuel classique d'ELIT.**

Présenté par : M^{lle} AHMED EL ARBI Amira.

Le : 10/09/2020.

Devant le jury :

- Mme CHIKHI Imane.
- Mme MANCER Yasmine.

Encadré par :

- Mr BENYAHIA Mohamed (USDB).
- Mr MELLOUK Ouali(ELIT).

Année universitaire : 2019/2020.

Résumé

Avec le développement constant des technologies dans le monde de l'IT, les entreprises dans les milieux informatiques commencent à pencher majoritairement vers la virtualisation et le Cloud Computing afin de bénéficier des nombreux avantages incontestés qu'ils offrent. Dans cette même approche nous rencontrons les Cloud privés qui malgré l'apparition de leurs fournisseurs depuis quelques années, leurs implémentations au sein des architectures entrepreneuriales sont en croissance constante.

Le présent mémoire introduit un projet actuel à acquérir dans les Datacenter d'Elit, qui consiste à l'ajout de plateforme de Cloud privé connu par OpenStack, afin d'optimiser et de maximiser l'utilisation des ressources tout en gardant un contrôle et une gestion totale.

Mots clés

Virtualisation, centre de données, Cloud privé, OpenStack, Open source, VMware, haute disponibilité.

Abstract

With the constant development of technologies in the IT world, companies in the IT world are beginning to lean towards virtualization and Cloud Computing in order to benefit from the many undisputed advantages that they offer. In this same approach, we have encountered private clouds which, despite the appearance of their suppliers in recent years, their implementation within entrepreneurial architectures is in constant suspicion.

This thesis develops a current project acquired in ELIT's data centers, which consists of the addition of private Cloud platform known by OpenStack, in order to optimize and maximize the use of resources while maintaining control and total management.

Keywords

Virtualization, data center, private cloud, OpenStack, Open source, VMware, high availability.

ملخص

مع التطور المستمر للتقنيات في عالم تكنولوجيا المعلومات، بدأت الشركات في عالم تكنولوجيا المعلومات تميل نحو الافتراضية والحوسبة السحابية من أجل الاستفادة من العديد من المزايا المقدمة والتي لا جدال فيها. في هذا النهج نفسه، واجهنا غيوما خاصة، على الرغم من ظهور مورديها بكثرة في السنوات الأخيرة، إلا أن تنفيذها داخل بنى تنظيم المشاريع يضل موضع شك دائم.

طورت هذه الأطروحة مشروعا حاليا يتم العمل عليه في مراكز البيانات في ELIT، والذي يتمثل في إضافة منصة سحابية خاصة معروفة بـ 'اوبن ستاك' OpenStack، من أجل تحسين استخدام الموارد في حده الأقصى، مع الحفاظ على التحكم والإدارة الكاملة.

الكلمات المفتاحية

المحاكاة الافتراضية، مركز البيانات، السحابة الخاصة، اوبن ستاك OpenStack، المصدر المفتوح، في ام وير VMWare، التوفر العالي.

Dédicaces

À mes chers parents, aucune dédicace ne saurait exprimer l'amour, le respect et ma considération envers vous pour les sacrifices que vous avez fait pour mon bien-être. Votre générosité et bonneté ont été toujours un exemple pour moi. Trouvez en ce travail, le fruit de votre patience et confiance que vous m'avez accordées.

À mon petit frère, qui a été toujours présent pour me supporter. Je te dédie ce travail avec tous mes vœux de réussite dans ton bac et rendre notre joie doublée.

À mes grands-parents, mes oncles, mes tantes et mes cousins et cousines.

À mes professeurs, à mes chers ami(e)s et collègues.

A tout ceux qui m'ont aidé de près o de loin, je dédie ce travail avec hommage.



Remerciement

Avant tout je remercie dieu qui m'a donné le courage et la volonté de réaliser ce modeste travail surtout dans les circonstances approuvées par le monde.

Je viens d'abord, à remercier Monsieur **Mohamed Benyahia**, enseignant à l'université de blida-1 pour ces précieux conseils, son estimable aide et le suivi qu'il m'a apporté.

J'adresse mes profondes gratitudees à Madame **Sehad Imane**, pour la confiance qu'elle m'a fait en me donnant l'opportunité à travailler sur ce projet. Ainsi qu'au Monsieur **Mellouk Ouali** et Monsieur **Moussaoui Mohamed** pour l'aide et les connaissances qui ont partagés avec moi et toute l'équipe du Datacenter.

Je tiens également à remercier « **TOUTS** » mes professeurs qui m'ont enseigné pendant mes cinq ans d'études, pour leurs conseils et orientations.

Enfin, un grand merci pour mes parents, sans vous je ne pourrais jamais y arriver ici.

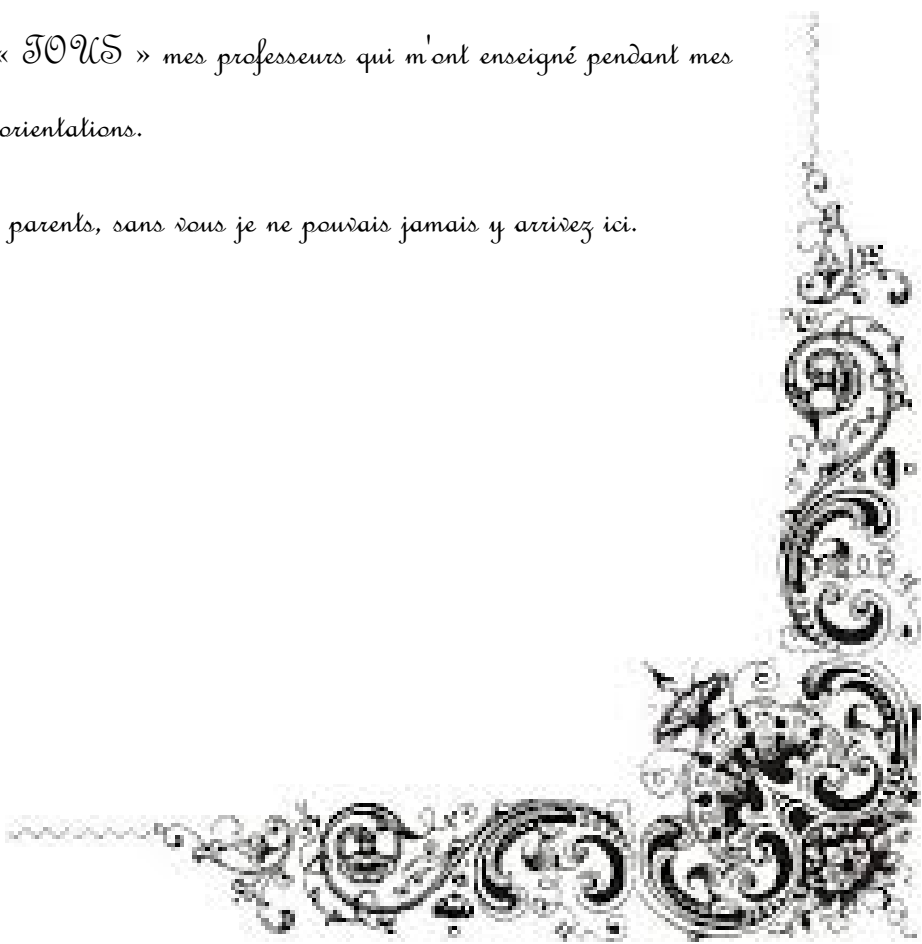


Table des matières

Introduction	1
Problématique	1
Objectifs	2
Structure du mémoire	2
Partie 1.....	3
État de l'art.....	3
Chapitre 1: Cloud Computing	4
1.1 Introduction	5
1.2 Cloud Computing	5
1.2.1 Historique	5
1.2.2 Définition	5
1.2.3 Caractéristiques du Cloud	6
1.3 Modèles des prestations des services.....	7
1.3.1 Infrastructure en tant que service(IaaS)	7
1.3.2 Plateforme en tant que service(PaaS).....	8
1.3.3 Software en tant que service(SaaS).....	8
1.3.4 Everything as a service (XaaS)	10
1.4 Modèles de déploiement des Cloud.....	10
1.4.1 Cloud public	10
1.4.2 Cloud privé.....	11
1.4.3 Cloud hybride.....	11
1.4.4 Cloud communautaire	12
1.5 Avantages et inconvénients	12
1.5.1 Avantages	12
1.5.2 Inconvénients	13
1.6 Éléments constitutifs du Cloud Computing :.....	13
1.6.1 Data Center.....	13
1.6.2 Virtualisation.....	14
1.6.3 Mécanisme	15
1.6.3.1 Hyperviseur de type 01	15
1.6.3.2 Hyperviseur de type02.....	16
1.6.4 Techniques de virtualisation.....	16
1.6.4.1 Virtualisation complète.....	16
1.6.4.2 La paravirtualisation	17

1.6.4.3	L'isolation	18
1.6.5	Types de virtualisation	19
1.6.5.1	Virtualisation de matérielle	19
1.6.5.2	Virtualisation des applications.....	19
1.6.5.3	Virtualisation des postes de travail.....	19
1.6.5.4	Virtualisation du stockage :	19
1.6.5.5	Virtualisation du réseau	19
1.6.5.6	Virtualisation de serveur.....	19
1.6.6	Avantages et inconvénients	20
1.6.6.1	Avantages	20
1.6.6.2	Inconvénients.....	20
1.6.7	Éditeurs et solutions	20
1.6.7.1	VMware	21
1.6.7.2	Citrix.....	21
1.6.7.3	Microsoft	21
1.7	Conclusion.....	21
Chapitre 2: Étude comparative et choix de la solution.....		22
2.1	Introduction	23
2.2	Solutions du Cloud Computing	23
2.3	Solutions propriétaires	23
2.3.1	Microsoft Azure	23
2.3.2	VCloud	24
2.3.3	Amazon EC2	24
2.4	Solutions Open Source	24
2.4.1	Eucalyptus	25
2.4.2	OpenNebula.....	25
2.4.3	CloudStack	25
2.4.4	OpenStack	26
2.5	Étude comparative des plateformes de solutions open source	26
2.5.1	Présentation d'hyperviseur.....	28
2.5.2	Réseau	29
2.6	Choix de la solution IaaS.....	30
2.7	Étude comparative des hyperviseurs	31
2.7.1	Xen	31
2.7.2	KVM.....	31

2.7.3	VSphere/ESXi	31
2.7.4	Hyper-V	31
2.8	Choix d'hyperviseur	32
2.9	Conclusion	32
Chapitre 3: Étude de la solution OpenStack		33
3.1	Introduction	34
3.2	Présentation d'OpenStack	34
3.3	Architecture OpenStack.....	36
3.3.1	Architecture conceptuelle.....	36
3.3.2	Architecture logique	36
3.4	Composants OpenStack.....	38
3.4.1	Horizon – Tableau de bord.....	38
3.4.2	Keystone – Identité.....	38
3.4.3	Nova – Calcul (Compute)	39
3.4.4	Glance- image	41
3.4.5	Neutron – réseau.....	42
3.4.6	Cinder- bloc de stockage	44
3.4.7	Swift- Stockage d'Objets	45
3.5	Conclusion.....	47
PARTIE 2.....		48
Conception, implémentation et test de la solution adaptée.		48
Chapitre 4: Conception de la solution		49
4.1	Introduction	50
4.2	Spécification des besoins.....	50
4.2.1	Besoins fonctionnels	50
4.2.1.1	Gestion d'images	50
4.2.1.2	Gestion d'instances.....	50
4.2.1.3	Gestion des volumes	50
4.2.1.4	Gestion des flavors	50
4.2.1.5	Gestion des projets.....	51
4.2.1.6	Gestion des utilisateurs	51
4.2.1.7	Gestion de la sécurité et de l'accès	51
4.2.2	Besoins non fonctionnels	51
4.3	Identification des acteurs	51
4.4	Spécification technique (Environnement matériel)	52

4.4.1	Partie stockage.....	52
4.4.1.1	Baie de stockage	52
4.4.2	Partie compute.....	52
4.4.2.1	Châssis	52
4.4.2.2	Serveurs blades	53
4.4.3	Partie réseau :	53
4.5	Architecture de Solution Openstack	53
4.5.1	Réseau Option 1 : Réseaux fournisseurs	54
4.5.2	Réseau Option 2 : Réseaux libre-service.....	55
4.6	Diagramme de cas d'utilisation global :.....	56
4.7	Diagramme de séquence	57
4.7.1	Diagramme de séquence « Créer projet »	57
4.7.2	Diagramme de séquence « Déployer VM »	58
4.8	Conclusion	59
Chapitre 5 : Implémentation et test de la solution OpenStack		60
5.1	Introduction	61
5.2	Installation de l'hyperviseur ESXi	61
5.3	Installation du VCenter.....	65
5.4	Préparation des VMS dans le cluster.....	68
5.5	Préparation des VMs avant les services OpenStack.....	76
5.6	Déploiement des services	83
5.7	Installation OpenStack.....	87
5.8	Déploiement de l'interface graphique web d'OpenStack.....	91
5.9	Conclusion	96
Conclusion Générale		97
Bibliographie.....		98

Liste des figures

Figure 1: Les services du Cloud Computing. [8]	9
Figure 2 : Le service Everything as a Service. [9]	10
Figure 3: Modèles de déploiement de Cloud. [14].....	12
Figure 4: Une salle de serveurs dans un data center. [16].....	14
Figure 5: Mécanisme de virtualisation. [18]	15
Figure 6: Hyperviseur de type 01. [20]	16
Figure 7: Hyperviseur de type 02. [20]	16
Figure 8: Virtualisation complète. [21]	17
Figure 9: Paravirtualisation. [21]	17
Figure 10: Mécanisme de redirections des appels. [21]	18
Figure 11: Isolation. [21].....	18
Figure 12: L'utilisation des solutions Open Source dans le marché. [31]	31
Figure 13: Architecture Conceptuelle d'OpenStack. [35]	36
Figure 14: Architecture logique d'OpenStack. [36]	37
Figure 15: Services et fonctions des projets OpenStack. [33].....	38
Figure 16: Le fonctionnement d'OpenStack Keystone. [39].....	39
Figure 17: Architecture d'OpenStack Nova. [41].....	41
Figure 18: Architecture d'OpenStack Glance. [43]	42
Figure 19: Architecture d'OpenStack Neutron. [45]	44
Figure 20: Architecture d'OpenStack Cinder. [47]	45
Figure 21: Architecture d'OpenStack Swift. [47].....	47
Figure 22: Architecture de solution. [49]	54
Figure 23: Option 1 de réseaux solution: Réseaux fournisseurs. [49]	55
Figure 24: Option 2 de réseaux solution: Réseaux libre-service. [49]	55
Figure 25: Diagramme de cas d'utilisations general.	56
Figure 26: Diagramme de séquence Créer projet.....	58
Figure 27: Diagramme de séquence Déployer une VM.....	59
Figure 28: Chargement d'installation d'ESXi.....	61
Figure 29: Démarrage d'installation d'ESXi.....	62
Figure 30: Contrat de licence utilisateur final.....	62
Figure 31: Sélection de disque.	63
Figure 32: Sélection de la disposition de clavier.....	63
Figure 33: Saisir le mot de passe.....	64
Figure 34: Confirmation d'installation d'ESXi.....	64
Figure 35: Fin d'installation d'ESXi.	65
Figure 36: Menu principale de VCenter.....	66
Figure 37: Choix de serveur ESXi.	66
Figure 38: Sélection de la taille de VCenter.	67
Figure 39: Configuration d'adresse.	67
Figure 40: Paramètres finales.	68
Figure 41: Vision sur le Cluster.	68
Figure 42: Le premier serveur.	69
Figure 43: Le deuxième serveur.....	69
Figure 44: Configuration du réseau du premier serveur.	70

Figure 45: Configuration du réseau du deuxième serveur.	70
Figure 46: Caractéristiques VM OpenStack Controlleur1	71
Figure 47: Caractéristiques VM OpenStack Compute1	72
Figure 48: Caractéristiques VM OpenStack block1	72
Figure 49: Caractéristiques VM OpenStack Object1	73
Figure 50: Caractéristiques VM OpenStack Object2	73
Figure 51: Configuration des ports pour le premier serveur.	74
Figure 52: Configuration des ports pour le deuxième serveur.	74
Figure 53: Configuration DRS du Cluster	75
Figure 54: Configuration HA.	75
Figure 55: Éditer les paramètres du cluster.	76
Figure 56: Création d'interface réseau sur le nœud compute1.	76
Figure 57: Création d'interface réseau pour le nœud controller1.	76
Figure 58: Configuration de la carte réseau "ens160" du nœud compute1.	77
Figure 59: Configuration de la carte réseau "ens160" du nœud controller1.	77
Figure 60: Configuration de la carte réseau "ens192" du nœud compute1.	78
Figure 61: Configuration de la carte réseau "ens192" du nœud controller1.	78
Figure 62: Configuration de la résolution de noms	79
Figure 63: Sauvegarde des données en utilisant Maria dB.	79
Figure 64: Configuration de MySQL.	80
Figure 65: Les bases de données de chaque service	80
Figure 66: Installation de RabbitMQ.	81
Figure 67: Installation de l'outil Memcached	81
Figure 68: Configuration du Memcached.	82
Figure 69: Installation d'ETCD.	82
Figure 70: Configuration de base de données pour chaque service.	83
Figure 71: Attribution des droits sur les bases de données.	84
Figure 72: Configuration d'accès à la base de données	84
Figure 73: Configuration du Compute1 pour s'authentifier à Keystone.	85
Figure 74: Configuration du Controller1 pour s'authentifier à Keystone.	85
Figure 75: Création du projet service.	86
Figure 76: La liste des services OpenStack	87
Figure 77: Liste des utilisateurs OpenStack	87
Figure 78: OpenStack Dashboard.	88
Figure 79: Vue d'ensemble du Dashboard	88
Figure 80: Etape1 de création d'instances.	89
Figure 81: Etape2 de création d'instances.	89
Figure 82: Etape3 de création d'instances.	90
Figure 83: Etape4 de création d'instances.	90
Figure 84: État d'instance.	91
Figure 85: Installation et configuration du service de serveur web.	91
Figure 86: Configurations de service web	92
Figure 87: Interface graphique de service web.	92
Figure 88: Interface projet administratif d'OpenStack.	93
Figure 89: Informations du projet.	93
Figure 90: Informations d'utilisateur.	94
Figure 91: Interface projet	94

Figure 92: Reconfiguration du site web officiel de Sonelgaz.....	95
Figure 93: Site web déployé.....	95

Liste des tableaux

Tableau 1: Différentes fonctionnalités des plateformes. [29]	27
Tableau 2: Comparaison des caractéristiques des solution Cloud [29].....	28
Tableau 3: Les différentes hyperviseurs supportées par les solutions Cloud open source [30].	29
Tableau 4: Comparaison des fonctionnalités réseaux selon les solutions IaaS. [30]	30
Tableau 5: Les principales caractéristiques de chaque hyperviseur [32].	32
Tableau 6: Tableau des versions d'OpenStack. [34]	35

Liste des Acronymes

AMD: Advanced Micro Device.

AMI: Amazon Machine Images.

AMQP: Advanced Message Queuing Protocol.

API: Application Programming Interface.

ASP: Application Service Provider.

AWS: Amazon Web Services.

BSD: Berkeley Software Distribution.

CISCO: Commercial & Industrial Security Corporation.

CLI: Command Line Interface.

DHCP: Dynamic Host Configuration Protocol.

DRS: Distributed Resources Scheduler.

DSI: Directeur du System d'Information.

EBS: Elastic Black Store.

EC2: Elastic Compute Cloud.

EFA: Elastic Fabric Adapters.

EMC: Richar Egain, Roger Marino, John Curly.

EOL: End of Life.

FAH: Fournisseur d'Application Hébergées.

FWaaS: Firewall as a Service.

GPFS: General Parallel File System.

GPL: General Public License.

GRE: Generic Routing Encapsulation.

HPC: High Performance Compute.

HTML: Hypertext Markup Language.

HTTP: Hypertext Transfer Protocol.

IaaS: Infrastructure as a Service.

IBM: International Business Machines corporation.

IBM: International Business Machines.

IoT: Internet of Things.

IP: Internet Protocol.

iSCSI: Internet Small Computer System Interface.

KVM: Kernel-based Virtual Machine.

LBaaS: Load Balancer as a Service.

LDAP: Lightweight Directory Access Protocol.

LUN: Logical Unit Number.

LVM: Logical Volume Management.

LXC: Linux Containers.

NASA: National Aeronautics and Space Administration.

NAT: Network Address Translation.

NFS: Network File System.

NIST: National Institute of Standards and Technology.

OCCI: Open Cloud Computing Interface.

ODL: Object Definition Language.

OS: Operating System.

OVF: Open Virtualization Format.

OVN: Open Virtual Network.

PaaS: Platform as a Service.

QCOW: QEMU Copy On Write.

QoS: Quality of Service.

RADOS: Reliable Autonomic Distribution Object Store.

REST: Representational State Transfer.

S3: Simple Storage Service.

SaaS: Software as a Service.

SAN: Storage Area Network.

SCP: Secure Copy.

SDN: Software Defined Networking.

SGBD: Système de Gestion de Base de Données.

SLA: Service Level Agreement.

SPICE: Simple Protocol for Independent Computing Environment.

SPOF: Single Point of Failure.

SQL: Structured Query Language.

SUN: Stanford University Network.

TBD: To Be Determined.

VCNS: Virtual Cloud Networking and Security.

VCPU: Virtual Central Processing Unit.

VLAN: Virtual Local Area Network.

VM: Virtual Machine.

VNC: Virtual Network Computing.

VPN: Virtual Private Network.

VXLAN: Virtual Extensible LAN.

WSGI: Web Server Gateway Interface.

XaaS: Everything as a Service.

XCP: eXplicit Control Protocol.

Introduction

Le Cloud Computing, ce terme qui est apparu depuis longtemps fait aujourd'hui un grand pas en approuvant des milliers de recherches ; et de développements faites dans ce domaine. Ceci n'est que le début, le Cloud Computing devient une nécessité dans le domaine informatique, pourquoi ? Simplement car il offre ce que veut le développeur, l'ingénieur, le chercheur...etc. Tous sont des clients qui utilisent les services offerts que peut importer le fournisseur de ces derniers. De plus la technologie se développe de plus que les sociétés se battent pour avoir les nouveautés en exclusivité cherchant la sécurité, la disponibilité et la flexibilité.

D'autre part, la virtualisation a fait son apparition de nouveau avec plusieurs mises à jour, qui suivent le rythme de développement surtout au niveau des entreprises qui veulent bénéficier des avantages de cette rénovation.

D'un point de vue général le Cloud Computing qui est basé sur la virtualisation a envahi le monde de commercialisation et s'est présenté comme solution délivrer jusqu'au client. Malgré ses défaillances, elle a été abordé à cause du coût réduit. Cela mène à dire que les entreprises ont besoin d'acquérir cette technologie afin d'assurer la gestion de leurs projets, productivité et équipes.

Sur ce point, nous allons assurer depuis notre projet une plateforme Cloud Optimale et hautement disponible à des fins d'utilisation pure, après avoir bien étudié l'architecture mise en place au niveau du centre des données, l'hébergeur de ce projet.

Problématique

Le défi majeur auquel nous ferons face sera d'intégrer cette solution de Cloud Open source sans support de déploiement contrairement aux cas classiques de production, et de l'adapter au niveau de l'environnement virtuel d'Elit, dans le but de bénéficier des avantages que fournit celui-ci à notre solution Cloud.

La problématique de ce sujet consiste à tirer profit des avantages de la solution OpenStack, en minimisant les inconvénients pour assurer une infrastructure de production organisé, optimal et hautement disponible.

Objectifs

1. Étude comparative des solutions existantes pour le Cloud Computing et définition d'une solution adéquate.
2. Étude des différentes fonctionnalités OpenStack.
3. Validation de la solution en la testant.

Structure du mémoire

Ce mémoire est composé de cinq chapitres, il commence par une introduction générale où sont décrites le contexte, la problématique et les objectifs.

- **Chapitre 1** : présente une description de la société d'accueil en spécifiant ses missions, les sociétés en participation ainsi de suite.
- **Chapitre 2** : présente un aperçu des concepts de base de la virtualisation, définition, mécanisme, technique et types, avantages et inconvénients ainsi que les éditeurs et solutions en faisant le passage vers le Cloud Computing en déterminant la relation entre la virtualisation et ce dernier, définition du Cloud Computing, caractéristiques, modèle de prestation de service, modèle de déploiement, avantages et inconvénients et l'évolution bien sûr.
- **Chapitre 3** : contient une présentation des solutions propriétaires et Open Source, étude comparative des plateformes des solutions Open Source, choix de la solution IaaS, étude comparative des hyperviseurs et choix de ce dernier.
- **Chapitre 4** : dans ce chapitre nous présentons la solution OpenStack, architecture et les composants.
- **Chapitre 5** : le dernier chapitre illustre la mise en œuvre de la solution choisie.

Le mémoire se termine par une conclusion générale qui rappelle le travail réalisé et propose des perspectives futures.

Partie 1
État de l'art.

Chapitre 1: Cloud Computing



1.1 Introduction

Avec l'évolution des technologies on se trouve devant des données massives qui ne peuvent pas être stockées ou utilisées depuis une seule source de données. Pour cela les Clouds ont apparu comme solution pour le stockage le traitement et la diffusion de ces informations. Basé sur la virtualisation surtout, le Cloud Computing est devenu la révolution dans le monde de la technologie informatique. Dans ce chapitre nous allons présenter le concept du Cloud Computing ainsi que les éléments principaux pour ce dernier.

1.2 Cloud Computing

1.2.1 Historique

Le concept de Cloud Computing est loin d'être nouveau, il est exploité depuis les années 2000, 'est la cinquième génération de l'informatique après les Mainframes, les PCs, les Clients/Serveurs et le Web [1], les changements qui ont permis l'apparition du Cloud Computing sont nombreux. Aussi il y a le concept de virtualisation qui permet une mutualisation des serveurs et offre donc une mise en production simplifiée et une meilleure utilisation des ressources. Le Cloud Computing est donc la juxtaposition de ces technologies pour passer à la vitesse supérieure sur l'exploitation de données à travers Internet. Le concept du Cloud Computing a été mis en œuvre en 2002 par Amazon, un leader du business, pour absorber la charge importante des commandes faites. [2]

1.2.2 Définition

Le terme Cloud Computing semble provenir de diagrammes de réseaux informatiques qui représentent internet sous forme de cloud. La plupart des grandes sociétés informatiques et sociétés d'études de marché telles que IBM, SUN Microsystems, Gartner et Forrester Research ont produit des livres blancs qui tentent de définir le sens de terme.

Selon NIST¹, le Cloud Computing est un modèle qui permet un accès réseau pratique et à la demande à un pool partagé de ressources informatiques configurables (par exemple : le réseau, les serveurs, le stockage, des applications et des services) qui peuvent être rapidement provisionnées et libéré avec un minimum d'effort de gestion ou d'interaction avec le fournisseur de service. [3]

¹ NIST: National Institute of Standards and Technology.

Autres comme Vaquero et al. Ont émulé avec la définition du NIST ou ils ont étudié 22 définitions du Cloud Computing et ont proposé la suivante, les nuages sont un vaste pool de ressources virtualisées facilement utilisable et accessibles (telles que le matériel, les plateformes de développement et/ou des services). Ces ressources peuvent être reconfigurables dynamiquement pour s'ajouter à une charge variable (échelle), permettant également une utilisation optimale des ressources. Ce pool de ressources est généralement exploité par un modèle de paiement à l'utilisation dans lequel des garanties sont offertes par le fournisseur d'infrastructure au moyen de SLA personnalisés. [4]

Donc le Cloud Computing est un concept qui consiste à la mise à disposition de ressources matérielles ou logicielles sous forme de service à la demande. De cette façon les applications et les données ne se trouvent pas sur des serveurs locaux mais dans le Cloud composé d'un certain nombre de serveurs distants interconnecté. Le Cloud Computing comme une nouvelle façon de délivrer les ressources informatiques, peut être vu comme un réseau informatique distribué, dont les ressources peuvent être fournies de manière dynamique et reposant sur un contrat de service entre le fournisseur et le client pour assurer le paiement d'utilisation des services.

1.2.3 Caractéristiques du Cloud

Pour considérer un service, une solution ou un environnement comme étant un Cloud Computing, il doit répondre aux caractéristiques bien précises. Selon NIST, cinq caractéristiques essentielles sont soutenues : [3]

- **Self-service à la demande :** un utilisateur peut allouer des capacités informatiques (serveurs, réseau, stockage, environnement d'exécution et application) selon les besoins sans nécessiter d'interaction humaine avec chaque fournisseur de service.
- **Accès aux réseaux étendu :** les capacités sont disponibles sur le réseau et accessibles via des mécanismes standard qui favorise leurs utilisations par des plateformes clients hétérogènes (par exemple : téléphones mobiles, tablettes, ordinateurs portables et postes de travail).
- **Mise en commun des ressources :** les ressources informatiques du fournisseur sont regroupées pour servir plusieurs consommateurs à l'aide d'un modèle multi-locataires. Différentes ressources physiques et virtuelles sont allouées et libérées dynamiquement selon la demande des consommateurs. Généralement le client n'a

aucun contrôle ou connaissance de l'emplacement exact des ressources fournies, mais peut être en mesure de spécifier l'emplacement géographique à un niveau d'abstraction plus élevé (par exemple : par pays, continent ou Datacenter). Des exemples de ressources incluent le stockage, le traitement, la mémoire et la bande passante du réseau.

- **Flexibilité des ressources :** les ressources sont allouées et libérées d'une façon élastique dans certains cas automatiquement pour évoluer rapidement à la demande, qu'elle soit croissante ou décroissante. Pour que les approvisionnements semblent souvent illimités et peuvent être adoptés à n'importe quelle quantité à tout moment.
- **Un service mesuré :** les systèmes en nuage contrôlent et optimisent automatiquement l'utilisation des ressources en tirant parti d'une capacité de mesure à un certain niveau d'abstraction adapté au type de service (par exemple : stockage, traitement, temps de calcul et bande passante). L'utilisation des ressources peut être surveillée, contrôlée et signalée en fournissant une transparence à la fois au fournisseur et au consommateur du service utilisé.

1.3 Modèles des prestations des services

La plupart des services de Cloud Computing peuvent être classés en trois grandes catégories. On les appelle parfois "pile" de Cloud Computing car elles s'empilent les unes sur les autres.

1.3.1 Infrastructure en tant que service(IaaS)

C'est la catégorie la plus basique des services du Cloud Computing. Les fournisseurs d'IaaS offrent aux utilisateurs un accès aux ressources tels que des serveurs, du stockage et l'équipement réseau. On peut caractériser ce service par : [5]

- L'utilisateur paie à la demande de l'IaaS au lieu d'acheter du matériel et donc une économie des coûts d'achat et de maintenance.
- L'indisponibilité des données ne peut pas être entraînée par des défaillances car elles sont stockées dans le cloud.

Parmi les principaux fournisseurs IaaS, on retrouve Amazon Web Services, Windows Azure, Google Compute Engine, Rackspace Open Cloud ou encore IBM SmartCloud

Entreprise. Pour les consommateurs de ce service on trouve les fournisseurs de développement et fournisseurs informatiques.

Citons quelques services IaaS ouvert au grand public qui permettent le stockage en ligne : Dropbox, Google Drive etc. Eucalyptus est aussi l'une des solutions IaaS mais qui est privée.

1.3.2 Plateforme en tant que service(PaaS)

Qualifie les services de Cloud Computing qui fournit aux utilisateurs un environnement dans lequel ils peuvent développer, tester, gérer et fournir des applications logicielles. En outre, les utilisateurs peuvent créer leurs applications web ou mobiles sans se soucier de la configuration ou de la gestion de l'infrastructure des ressources nécessaires. Les principales caractéristiques de ce service sont : [6]

- Offrir une plateforme avec des outils pour tester, développer et héberger les applications, donc ne pas se soucier de l'infrastructure sous-jacente, aussi il facilite le travail collaboratif même entre les équipes à distance.
- Les fournisseurs gèrent la sécurité, les systèmes d'exploitation, les logiciels de serveurs et les sauvegardes.

Parmi les principaux fournisseurs PaaS, on compte Force.com qui propose une plateforme CRM aux entreprises. Les autres plateformes PaaS de développement et de gestion de logiciel sont Apper IQ, Mendix, Amazon Web Services Elastic Beanstalk, Google App Engine ou encore Heroku. Pour les consommateurs on trouve les développeurs web et développeurs d'application.

1.3.3 Software en tant que service(SaaS)

C'est une méthode de diffusion d'application logicielle via Internet à la demande et sur abonnement. Les utilisateurs donc n'installent pas les applications sur leurs terminaux mais plutôt les utilisent pour stocker et analyser des données ainsi qu'il collabore des projets. [7]

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

CHAPITRE 1: CLOUD COMPUTING

navigateur internet. Le déploiement, la configuration, la maintenance, la sauvegarde etc. sont désormais de la responsabilité du fournisseur du service, le client est alors consommateur.

Les caractéristiques de ce service se présente comme suit :

- Les logiciels et applications sont proposés par les fournisseurs selon un modèle par abonnement et ils se chargent de la gestion, l'installation et la mise à niveau de ceux-ci.
- Les données sont sécurisées, donc on ne risque pas de perdre des informations lors d'une panne d'un équipement.
- Les applications sont accessibles depuis n'importe quel terminal connecté à internet et virtuellement n'importe où dans le monde, or l'utilisation peut être dimensionnée en fonction des besoins.

Pour les fournisseurs de ce service on trouve presque tous les fournisseurs en ligne. Citons par exemple Google App et « Exchange Online »² de Microsoft. Et pour les consommateurs c'est la quasi-totalité des internautes. Gmail est un exemple de SaaS très connu, très utilisé et gratuit.

La figure 1 ci-dessous montre les différents services du Cloud Computing et ses utilisateurs.

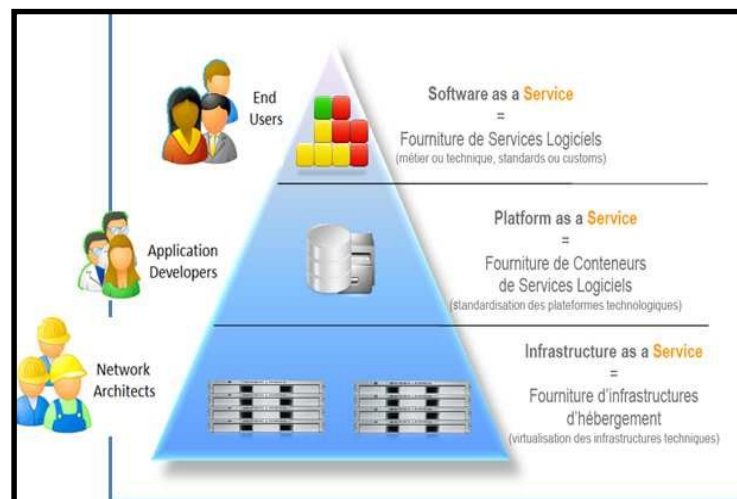


Figure 1: Les services du Cloud Computing. [8]

² « Exchange Online » permet de bénéficier d'une messagerie fournie en tant que service hébergé par Microsoft sans devoir la gérer en interne.

Bien que les trois types de services de Cloud Computing mentionnés avant sont les plus connus, ils peuvent être tous inclus dans un seul type qui est Everything as a service.

1.3.4 Everything as a service (XaaS)

Appelé aussi “anything as a service”. L'acronyme (XaaS) est un terme général qui englobe une variété de produits, d'outils et de technologie qui apparaissent comme des offres de service.

L'apparition de ces nouveaux services ainsi que leurs acronymes associés a fait apparaître des annuaires aaS (as a Service) pour garder une trace. Chacun de ces services de Cloud est une industrie. Parmi les services les plus populaires, citons : Desktop as a service (DaaS), Artificial intelligence as a service (AIaaS) et Unified Communication as a service (UaaS).

La figure 2 ci-dessous montre le service XaaS qui englobe tous les services du Cloud Computing.

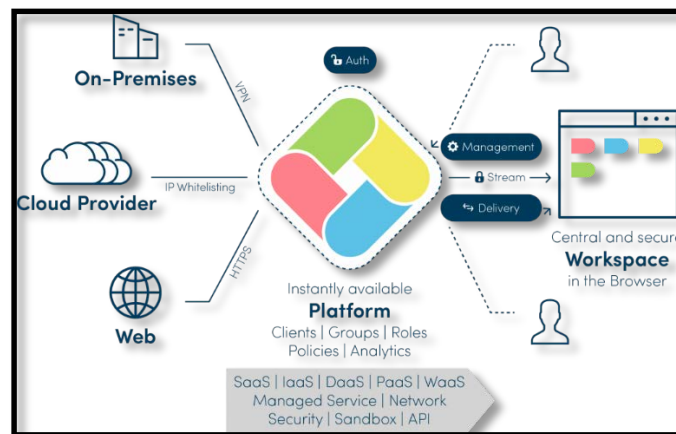


Figure 2 : Le service Everything as a Service. [9]

1.4 Modèles de déploiement des Cloud

1.4.1 Cloud public

Est un pool de ressources virtuelles, créé à partir de matériel détenu et géré par une entreprise tierce ; qui sont automatiquement approvisionnées et allouées à différents clients via une interface en libre-service, de telle sorte que les clients ne doivent payer que pour les cycles de processeur, le stockage ou la bande passante qu'ils consomment. [10]

Alibaba cloud, Microsoft Azure, Google Cloud, Amazon web services(AWS) et IBM Cloud sont les principaux fournisseurs de Cloud public.

Les aspects majeurs d'un Cloud public sont :

- **Allocation des ressources** : les clients partagent les services Cloud et les ressources virtuelles qui proviennent de l'infrastructure, des plateformes et des logiciels de celui-ci.
- **Accords d'utilisation** : les ressources sont distribuées en fonction des besoins, mais le modèle de paiement à l'utilisation n'est pas appliqué. Certains organismes de recherches qui ont recours au Massachusetts Open Cloud utilisent les Cloud publics sans payer de frais.
- **Gestion** : le prestataire assure au moins la maintenance du matériel sur lequel repose le cloud, prend en charge le réseau et gère le logiciel de virtualisation.

1.4.2 Cloud privé

Est généralement défini comme environnement réservé à l'utilisateur final ou à un groupe d'utilisateurs à l'intérieur de pare-feu. Ce type de Cloud offre aux entreprises un bon nombre d'avantages (dont le libre-service, l'extensibilité et l'élasticité) auxquels s'ajoutent les possibilités de contrôle et un niveau de sécurité et de confidentialité supérieure. [11]

Le Cloud privé ne repose désormais plus forcément sur l'infrastructure sur site, car les entreprises créent des Cloud privés dans des Datacenter hors site et loués à des fournisseurs. Cette tendance a fait naître différents sous-types de Cloud privés, notamment :

- **Cloud privé géré** : ce type est créé et utilisé par les clients, tandis qu'il est déployé, configuré et géré par un fournisseur tiers.
- **Cloud dédié** : il s'agit d'un Cloud au sein d'un autre Cloud. Un Cloud dédié peut être déployé dans un Cloud public ou dans un Cloud privé.

1.4.3 Cloud hybride

C'est une forme d'architecture qui offre un certain degré de portabilité, d'orchestration et de gestion de la charge de travail entre deux ou plusieurs environnements. Les caractéristiques de ce type sont complexes et les exigences peuvent varier selon l'utilisateur qui les définit. [12]

Un Cloud hybride peut inclure :

- Au moins un Cloud privé et un Cloud public.
- Au moins deux Cloud publics ou deux Cloud privés.
- Un environnement nu ou virtuel connecté à au moins un Cloud privé ou public.

1.4.4 Cloud communautaire

Plus rarement utilisé, il consiste à partager un espace donné entre plusieurs entreprises ayant les mêmes exigences en matière de sécurité et de confidentialité. Il s'apparente donc à un Cloud privé partagé. Parmi ses caractéristiques : [13]

- Il peut être utilisé pour des applications génériques, mais qui ont des spécificités adaptées aux contraintes du groupe.
- Il peut aussi héberger une application " métier " très spécialisée, mais commune à de très nombreuses entreprises.

Amadeus et Cmed sont deux exemples de Cloud communautaires.

La figure 3 suivante montre les différents types de Cloud dont on a parlé.

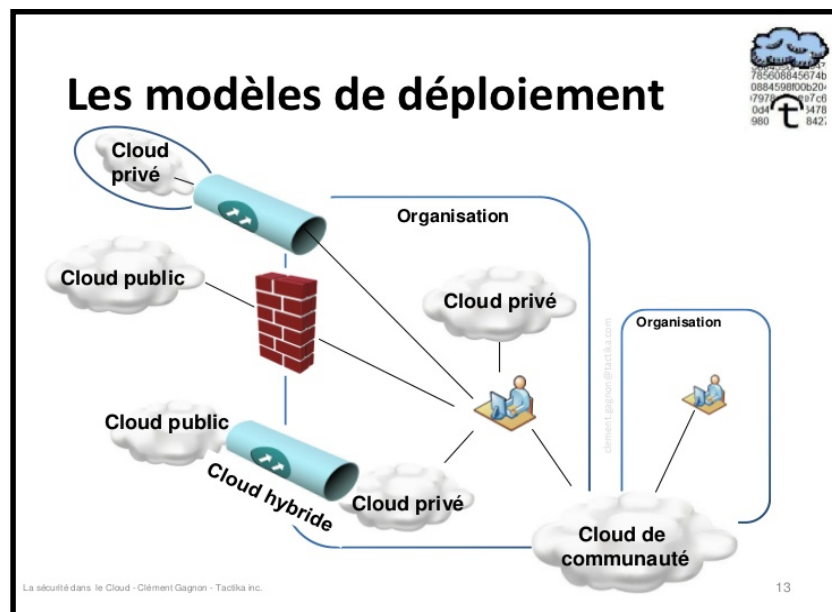


Figure 3: Modèles de déploiement de Cloud. [14]

1.5 Avantages et inconvénients

1.5.1 Avantages

- **Rentabilité** : la principale raison du passage au Cloud Computing est qu'il nécessite des coûts nettement inférieurs à ceux d'une technologie sur site. De ce fait le Cloud offre un espace énorme de stockage, ce qui permet d'économiser de l'argent et des ressources sur les entreprises.
- **Mise à jour et évolutivité** : il suffit de mettre à jour l'application réseau et tous les utilisateurs bénéficient des nouveautés et des corrections.

- **Mise en commun des ressources** : chaque utilisateur contribue à l'enrichissement des données et des expériences de l'ensemble si des outils collaboratifs sont mises en place. Cet avantage facilite la gestion et la transmission des connaissances dans les entreprises.
- **Sécurité** : on évite le problème de perte des données confidentielles car elles ne sont pas présentes en local.
- **Mobilité** : l'utilisateur peut y accéder à tout moment et de n'importe quel type d'appareil qui est doté d'un navigateur à ses applications et son workflow.
- **Haute vitesse** : le Cloud Computing permet de déployer rapidement le service en moins de clic. Ce déploiement rapide permet d'obtenir les ressources nécessaires pour le système en moins d'une minute.

1.5.2 Inconvénients

- **Sécurité** : la plateforme cloud, si elle est externe, doit être suffisamment sécurisée pour éviter le risque d'intrusion ou de vol des données par piratage.
- **Faible bande passante** : parfois la bande passante est faible car plusieurs utilisateurs accèdent au Cloud en même temps, ce qui la réduit. Avec moins de rapidité, les avantages du Cloud Computing ne peuvent pas être réalisées.
- **Problèmes de flexibilité** : les services de Cloud fonctionnent sur des serveurs distants, ce qui complique le contrôle des logiciels et du matériel par les entreprises.
- **Incompatibilité** : étant donné que toute l'infrastructure est virtualisée, l'incompatibilité peut survenir à des moments où de sérieux problèmes apparaissent concernant le bon fonctionnement des services.
- **L'interopérabilité** : peut être prise comme un obstacle que d'inconvénient, elle s'agit de la capacité des ressources d'un fournisseur de Cloud à communiquer avec des ressources d'un autre fournisseur de cloud.

1.6 Éléments constitutifs du Cloud Computing :

1.6.1 Data Center

Un data center ou centre de données est une infrastructure physique sécurisé où regroupe un ensemble d'éléments comme du matériel informatique (ordinateurs, serveurs, unités de stockage, etc...), des équipements réseaux et de télécommunication, des systèmes de contrôle et de sécurité (climatisation, incendie...). [15]

Cette infrastructure peut être utilisée pour organiser, traiter, stocker et entreposer des grandes quantités de données la figure 4 suivante montre un exemple d'une salle de serveurs dans un data center. La taille d'un Datacenter peut varier d'une simple pièce à un immeuble de plusieurs étages, et va permettre à une entreprise ou une administration de fonctionner à l'ère d'internet et du cloud. [15]



Figure 4: Une salle de serveurs dans un data center. [16]

Les data centers sont aussi considérés comme des sites physiques ultra sécurisés car ils traitent et protègent des données sensibles. Ils doivent fonctionner sans interruption étant garants du bon fonctionnement de tout l'écosystème web. La qualité des performances de ces infrastructures et le niveau de sécurité élevé sont les grands avantages des data centers. [15]

1.6.2 Virtualisation

La virtualisation c'est la technique qui permet à plusieurs utilisateurs d'exploiter des ressources matérielles, logicielles ou environnementales à partir d'une seule machine physique, en se basant sur deux principes fondamentaux : [17]

Le cloisonnement : chaque système d'exploitation fonctionne indépendamment des autres et ne peut pas inférer les autres.

La transparence : le fonctionnement en mode virtualisé ne change rien au fonctionnement du système d'exploitation ou des applications.

1.6.3 Mécanisme

Le principe de virtualisation repose sur le mécanisme suivant :

- Un système "hôte" qui est un simple système d'exploitation installé sur un serveur physique unique, va servir afin d'accueillir plusieurs autres systèmes d'exploitation par un logiciel appelé "hyperviseur".
- L'hyperviseur est donc un logiciel de virtualisation qui est directement installé sur le système d'exploitation principal ou hôte. Celui-ci va permettre de créer plusieurs environnements clos et indépendants qui pourront à leur tour héberger d'autres systèmes invités. Les environnements indépendants ainsi créés grâce à l'hyperviseur sont des machines virtuelles.

La figure 5 suivante montre le mécanisme de la virtualisation :

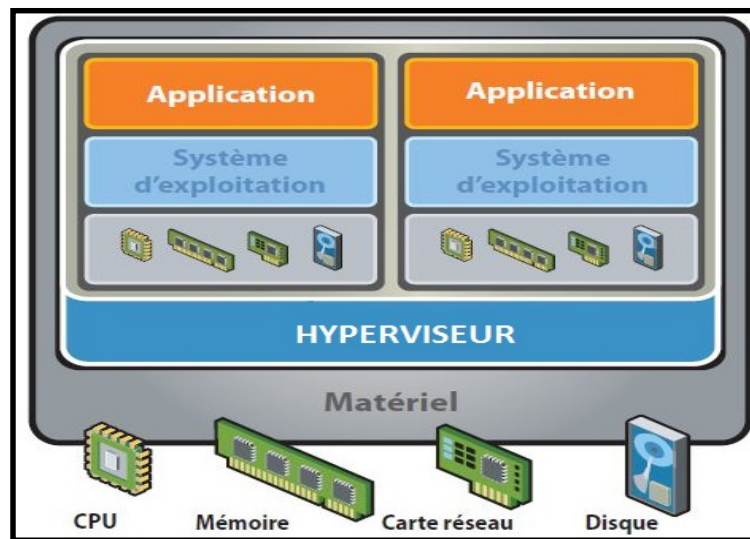


Figure 5: Mécanisme de virtualisation. [18]

Donc le système hôte va créer plusieurs environnements entièrement indépendants grâce à un hyperviseur. Ces machines virtuelles vont pouvoir communiquer avec les ressources physiques et le hardware du serveur physique (mémoire, espace disque etc...).

Les hyperviseurs peuvent être divisés en deux types :

1.6.3.1 Hyperviseur de type 01

Noyau allégé et optimisé aussi appelé natif, est un logiciel qui s'exécute directement sur une plateforme matérielle comme le montre la figure 6 ci-dessous, elle est considérée comme outil de contrôle de système d'exploitation. De ce fait un système d'exploitation secondaire peut être exécuté au-dessus du matériel. [19]

Parmi les hyperviseurs de type 01, on trouve : ESXi Server, Xen, Hyper-V etc...

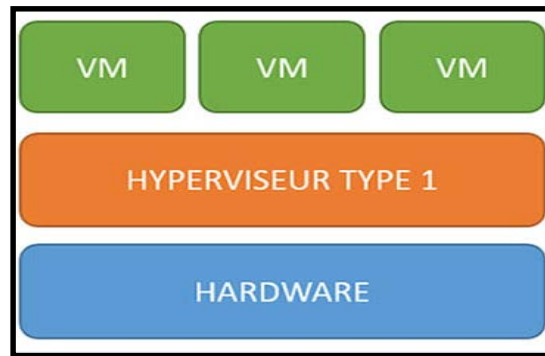


Figure 6: Hyperviseur de type 01. [20]

1.6.3.2 Hyperviseur de type 02

Dit "hosted" ou architecture invitée, est un logiciel qui s'exécute à l'intérieur d'un autre système d'exploitation comme illustré dans la figure 7 ci-dessous, un système d'exploitation invité s'exécutera en troisième niveau au-dessus du matériel. L'intérêt qu'on peut trouver c'est le fait de pouvoir exécuté plusieurs hyperviseurs simultanément vu qu'ils ne sont pas liés à la couche matérielle. [19]

Parmi les hyperviseurs de type 2 on trouve : VMware Player, VMware Workstation, Virtual Box etc...

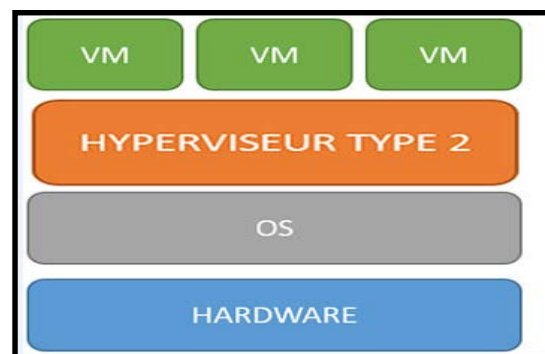


Figure 7: Hyperviseur de type 02. [20]

1.6.4 Techniques de virtualisation

Il existe différentes techniques de virtualisation, citons par niveau d'abstraction décroissant :

1.6.4.1 Virtualisation complète

Virtualisation est dite complète lorsque le système d'exploitation invité n'a pas conscience d'être virtualisé, donc n'a aucun moyen de savoir qu'il partage le matériel avec d'autres OS.

Ainsi, l'ensemble des systèmes d'exploitation virtualisés s'exécute sur un ordinateur unique, pouvant fonctionner de manière totalement indépendante les uns des autres et être vu comme des ordinateurs à part entière sur réseau. [21] La figure 8 montre l'architecture de la virtualisation complète.

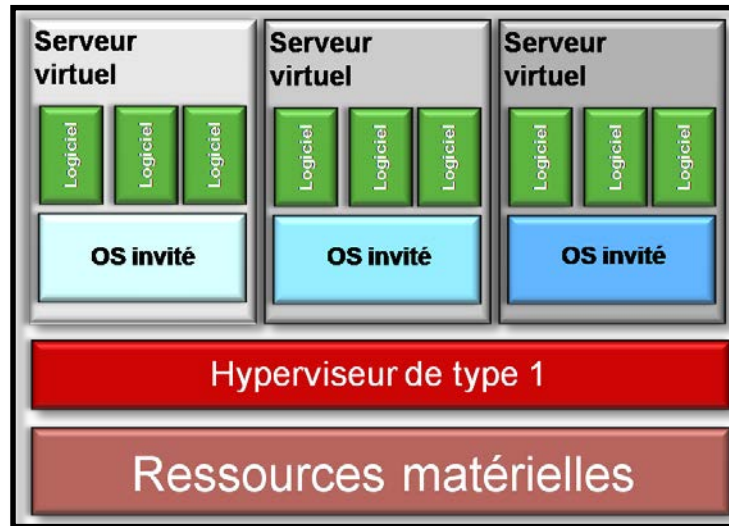


Figure 8: Virtualisation complète. [21]

1.6.4.2 La paravirtualisation

Par opposition à la virtualisation, on parle de la paravirtualisation lorsque le système d'exploitation doit être modifié pour fonctionner sur un hyperviseur de paravirtualisation. Les modifications sont en fait des insertions de divers permettant de rediriger les appels système au lieu de les traduire (par des drivers "back-end" et "front-end"). [21] L'architecture de paravirtualisation est montrée dans la figure 9 qui suit.

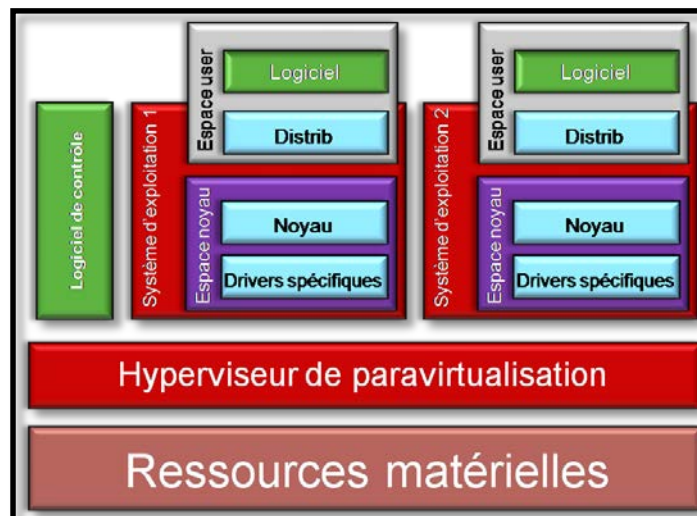


Figure 9: Paravirtualisation. [21]

Le mécanisme de redirection des appels système est expliqué ici dans la figure 10 :

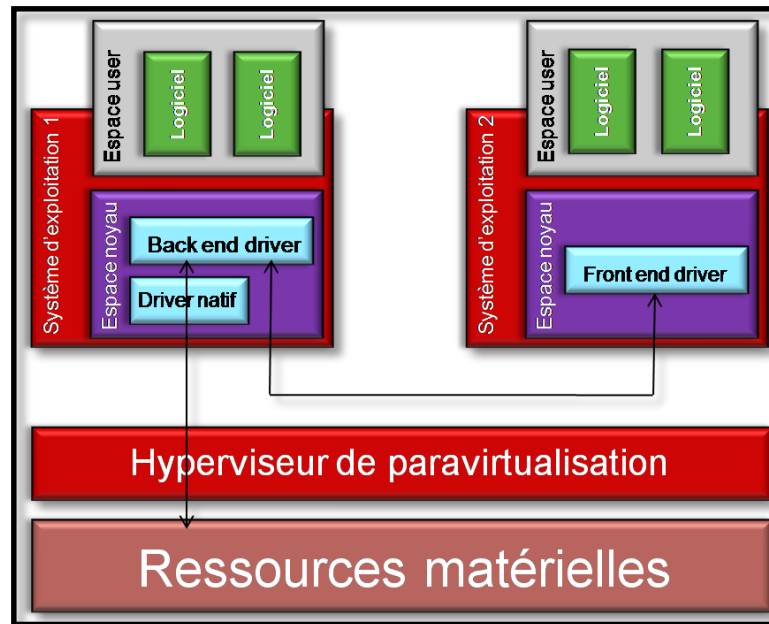


Figure 10: Mécanisme de redirections des appels. [21]

1.6.4.3 L'isolation

Consiste à mettre en place, sur un même noyau de système d'exploitation, une séparation forte entre différents contextes logiciels. Il s'agit de la technique de virtualisation la plus "légère" qui existe [17]. La figure 11 suivante montre l'architecture d'Isolation.

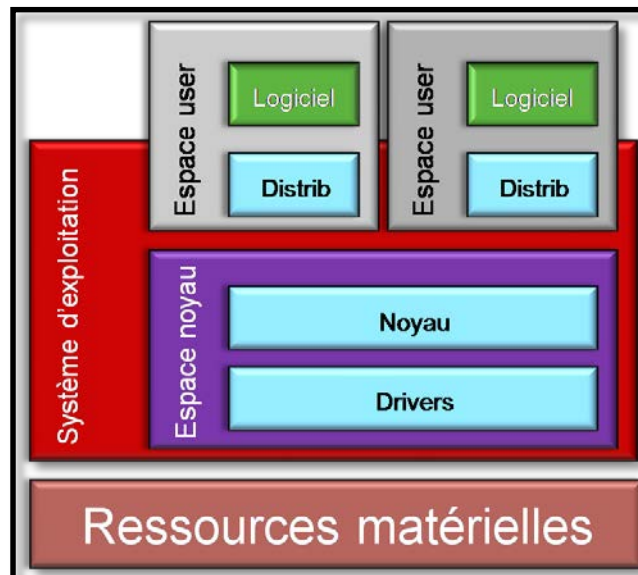


Figure 11: Isolation. [21]

1.6.5 Types de virtualisation

Il existe différents types de virtualisation. On recense les principaux comme suit :

1.6.5.1 Virtualisation de matérielle

C'est le type le plus courant, dans ce cas un gestionnaire de machine virtuelle reflète les versions virtuelles des systèmes d'exploitation et les fusionne en un énorme serveur physique unique. Il aide à exécuter plusieurs systèmes d'exploitation sur un même périphérique en même temps.

1.6.5.2 Virtualisation des applications

Une application peut être utilisée sur un autre hôte à partir duquel elle est installée dans différentes tailles. Ce type est également appelé virtualisation du service d'application.

1.6.5.3 Virtualisation des postes de travail

L'environnement de bureau de travail est séparé de la machine physique et organisé en infrastructure de postes de travail virtuels. Elle aide les utilisateurs à accéder à leurs bureaux à partir de n'importe quel endroit.

1.6.5.4 Virtualisation du stockage :

Consiste à regrouper l'ensemble des périphériques de stockage physique en un seul périphérique simulé centralisé. Ce dernier est géré depuis une console centrale. La manipulation du stockage dans le Cloud est surtout utilisée pour la sauvegarde, l'archivage et la récupération des données en masquant l'architecture de stockage réelle et physique complexe.

1.6.5.5 Virtualisation du réseau

C'est le processus reproduisant un réseau physique et ses différentes composants (ports, routeurs...), cette technique permet de mettre à disposition des réseaux isolés, cloisonnés mais fonctionnant sur une infrastructure mutualisée ; aussi aide à mieux superviser et à identifier l'utilisation des données. Elle assure également la sécurité en limitant le mouvement des fichiers sur plusieurs réseaux.

1.6.5.6 Virtualisation de serveur

Par un principe d'émulation, une couche logicielle "hyperviseur" isole les ressources physiques des systèmes d'exploitation et les regroupent en un seul serveur. Par ce principe, plusieurs systèmes d'exploitation peuvent cohabiter sur une même machine, indépendamment l'un de l'autre.

1.6.6 Avantages et inconvénients

1.6.6.1 Avantages

Faire le choix de la virtualisation présente les avantages :

- Coûts réduits car elle implique moins de matériel donc moins d'espace et de maintenance.
- Économie d'énergie.
- Meilleure exploitation des ressources.
- Continuité d'activité c'est-à-dire en cas d'interruption, la virtualisation facilite le plan de reprise d'activité.
- Meilleure agilité car elle encourage la flexibilité des processus et la mobilité des équipes.

1.6.6.2 Inconvénients

On peut trouver les inconvénients suivants :

- La mise en œuvre de la virtualisation induit une complexité de gestion et apporte son lot de risque inhérent au "tout virtuel".
- La gestion des données qui sont partagées sur plusieurs disques physiques et stockées sur des volumes logiques, donc il faut faire une cartographie détaillée à un instant T. Les LUN (Logical Unit Number) doivent être normés, nommés et numérotés.
- La gestion des pannes : Comme on le sait, la virtualisation masque souvent l'origine des pannes grâce aux fonctionnalités automatiques de répartition de charge, donc le système continue de fonctionner mais s'il survient une panne importante, la localisation de cette panne devient très difficile.

1.6.7 Éditeurs et solutions

Le marché des solutions de virtualisation est extrêmement concurrentiel. Parmi les principaux acteurs nous parlons des trois principaux éditeurs à savoir VMware, Citrix et Microsoft. [22]

1.6.7.1 VMware

VMware est probablement le plus vieux et le plus gros éditeur sur le marché de la virtualisation. Filiale du groupe EMC³ et fondé en 1998, cette compagnie propose une gamme de produits et services complète associée à la virtualisation d'architecture x86. [22]

1.6.7.2 Citrix

Citrix a rejoint le marché de la virtualisation en 2007 suite à l'achat de XenSource, une compagnie ayant développé son propre hyperviseur. L'offre de virtualisation de Citrix contient différents produits, Citrix a également noué de solides partenariats avec l'éditeur Microsoft permettant de faciliter l'intégration de ses solutions de virtualisation au sein d'une architecture Microsoft. [22]

1.6.7.3 Microsoft

Microsoft est arrivé sur le marché de la virtualisation récemment, avec son Hyperviseur baptisé Hyper-V qui se présente sous la forme d'un rôle disponible avec la dernière édition du système d'exploitation de la gamme serveurs de la firme, à savoir Windows 2008 Server. [22]

1.7 Conclusion

Dans ce chapitre nous avons donné une étude théorique sur les technologies de virtualisation et du Cloud Computing, en citant tous les éléments qui peuvent être utiles.

Dans le chapitre suivant, nous allons comparer les différentes solutions et nous allons préciser le choix de la meilleure.

³ EMC : connu aujourd'hui par Dell EMC, les trois lettres signifient les noms des membres fondateurs, Richard Egan, Roger Marino et John Curly.

Chapitre 2: Étude comparative et choix de la solution



2.1 Introduction

Dans ce chapitre, nous allons étudier et comparer les différentes solutions du Cloud Computing propriétaire et open source. Nous choisissons par la suite la solution la plus optimale et adéquate pour notre projet.

2.2 Solutions du Cloud Computing

Les services Cloud open source et propriétaires visent à fournir des logiciels ou services fiables aux utilisateurs finaux. Certains préfèrent travailler avec des grandes entreprises comme Amazon ou Microsoft, avec des programmes et services compatibles, d'autres préfèrent l'interopérabilité et la flexibilité de l'alternative Open Source comme OpenStack ou Eucalyptus. Pour ceux, qui veulent des solutions plus gérées, peuvent choisir le logiciel propriétaire mais pour les autres le logiciel open source est le choix évident.

Généralement les modèles open source sont gratuits et ne facturent pas l'utilisation de logiciels, alors que les modèles propriétaires offrent des forfaits gratuits au début comme des mises à jour, mais finissent toujours par coûter au client dans le cas de la mise à niveau et la possibilité d'ajouter de nouveaux packages.

2.3 Solutions propriétaires

Parmi les solutions propriétaires les plus communs au marché, on a :

2.3.1 Microsoft Azure

Regroupe plusieurs services de cloud, on retrouve notamment un service de stockage, des machines virtuelles, et des réseaux de diffusion de contenu. Azure propose également des services exploitant les technologies propriétaires de Microsoft. [6]

Azure a été annoncé en Octobre 2008 et publié le 1er février 2010 sous le nom de "Windows Azure " avant d'être renommé "Microsoft Azure" le 25 mars 2014.

Parmi les services proposé, Remote App qui permet le déploiement de programmes Windows sur différents systèmes d'exploitation par le biais d'une machine virtuelle, Azure IoT permettant de connecter et surveiller les objets connectés, et le service HDInsight es un déploiement customisé de la plateforme BigData Hadoop.

2.3.2 VCloud

VMware vCloud suite est un package qui inclut l'hyperviseur vSphere Entreprise ; la plateforme de gestion du Cloud hybride avec les nouvelles unités de licence portables de VMware qui permettent à vCloud suite de créer et gérer à la fois des Cloud privés basés sur vSphere et des Cloud hybrides multifournisseurs. [23]

Les principaux composants sont :

- **VSphere/ESXI** : l'hyperviseur qui est chargé sur les serveurs physiques.
- **VCenter** : la console de gestion centralisée pour tous les hôtes vSphere et les machines virtuelles.
- **VCloud Director** : le portail de libre-service Cloud privé.
- **VCloud Networking and Security (VCNS)** : précédemment appelé vSphere, VCNS est ce qui permet de sécuriser l'infrastructure.

2.3.3 Amazon EC2

Amazon Elastic Compute Cloud est un service web qui fournit une capacité de calcul sécurisée et redimensionnable dans le Cloud. Il est conçu pour faciliter l'accès aux ressources de Cloud Computing à l'échelle du web destiné aux développeurs. L'interface du service web Amazon EC2 permet d'obtenir et configurer des capacités avec un minimum d'efforts. [24]

Les modules de ce service sont :

- **Système AWS Nitro** : plateforme structurelle de la dernière génération d'EC2 ; permet d'innover rapidement, réduire les coûts et offrir des avantages supplémentaires comme la sécurité.
- **Amazon Elastic Block Store (EBS)** : propose un stockage par blocs "hautes performances" pour toutes les charges de travail.
- **Elastic Fabric Adapter (EFA)** : interface réseau qui permet à des applications comme "Calcul à Hautes Performances(HPC)" de communiquer entre plusieurs instances à hauteur de 100Gb/s.

2.4 Solutions Open Source

On va présenter maintenant les solutions Open source du Cloud Computing les plus connus.

2.4.1 Eucalyptus

Issue d'un projet de recherche de l'université de Californie, cette plateforme Cloud open source est intégrée dans Ubuntu server et Debian. Écrite en c, java et python, permet de créer des Cloud privés ou hybrides et supporte des machines virtuelles linux ainsi que les hyperviseurs Xen et KVM, compatible aussi avec EC2 d'Amazon. [25]

Il existe également une version propriétaire commercialisée par la société Eucalyptus system.

Cette solution apporte de fonctionnalités supplémentaires comme le support de VMware, celui des machines virtuelles Windows et l'intégration SAN.

2.4.2 OpenNebula

OpenNebula, purement Open source permet de déployer des Cloud privés, hybrides et publics. Écrite en c++, Ruby et Shell, elle supporte les plateformes de virtualisation Xen, KVM et VMware ainsi que le service "On-demand" d'Amazon EC2. Le projet est publié sous licence Apache 2.0. [26]

Parmi ses fonctionnalités : gestion centralisée des machines virtuelles et des ressources physiques, répartition des charges, extension des capacités par ajout de serveurs physiques.

La majorité de ces solutions sont des solutions d'infrastructure permettant une gestion simplifiée d'architectures complexes.

2.4.3 CloudStack

CloudStack, issu du rachat de Cloud.com par Citrix, a été conçu pour permettre le déploiement et la gestion d'importants réseaux de machines virtuelles. Il supporte les principaux moteurs de virtualisation à savoir : vSphere, Oracle VM, KVM, Xen mais aussi les services Cloud d'Amazon. [27]

Les caractéristiques principales de CloudStack sont les suivants :

- Réseau virtuel (Support VLAN).
- Piscine de ressources (permet aux administrateurs de limiter les ressources virtuelles).
- Routeurs virtuels, une pare-feu et un équilibreur de charge.
- Migration en direct avec le maintien d'accueil.

2.4.4 OpenStack

C'est le résultat de fusion de deux projets portés l'un par la NASA⁴ et l'autre issu de l'effort de l'hébergeur Américain Rackspace Cloud Server. L'ensemble de la plateforme est sous licence Apache 2.0. [28]

OpenStack est un logiciel libre qui permet la construction de Cloud privé et public. Il est aussi une communauté et un projet en plus qu'un logiciel qui a pour but d'aider les organisations à mettre en œuvre un système de serveur de stockage virtuel.

Il s'installe sur un système d'exploitation libre comme Ubuntu ou Debian et se configure entièrement en ligne de commande. C'est un système robuste qui a fait ses preuves auprès des professionnels du domaine.

La technologie est populaire parmi une vaste communauté de spécialistes et soutenue par des sociétés telles que Cisco, Dell, NASA, Intel, AMD, Citrix, Rackspace et Right Scale.

Les principales caractéristiques d'OpenStack sont les suivantes :

- Capacité à gérer les ressources du serveur de produits virtuels.
- Capacité à gérer les réseaux locaux.
- Gestion de l'image de la machine virtuelle.
- Groupes de sécurités.
- Contrôle d'accès basé sur les rôles.
- VNC proxy via un navigateur web.

2.5 Étude comparative des plateformes de solutions open source

Afin de prendre le choix convenable et le plus adéquat, pour la construction de notre cloud, on a fait une étude comparative des solutions open source définie dans la section précédente en se basant sur plusieurs critères.

⁴ NASA : National Aeronautics and Space Administration.

CHAPITRE 2: ÉTUDE COMPARATIVE ET CHOIX DE LA SOLUTION

Le tableau 1 ci-dessous résume les principales fonctionnalités de chaque solution :

	CloudStack	Eucalyptus	OpenStack	OpenNebula
Date de sortie	4 mai 2010	29 mai 2008	21 octobre 2010	Mars 2008
Licence	Licence Apache	Propriétaire, GPL v3	Licence Apache	Licence Apache
Modèle de déploiement	Cloud public et privé	Cloud privé et hybride	Cloud public et hybride	Cloud privé, public et hybride
Type de service	IaaS	IaaS	IaaS	IaaS
Interface d'accès d'utilisateur	Gestion riche, interface utilisateur en libre-service	Service web, ligne de commande	Interface web	Libvirt, EC2, OCCl API
genre de service	Offre de service disque, réseau et Template	Compute, Storage	Compute(Nova), Storage (Swift)	Compute, storage
Compatibilité	Supporte Amazon EC2 et S3 API	Supporte EC2, S3	Supporte différentes plateformes	Ouvert, multiplateformes
Os support	Windows, Linux et différentes versions BSD	Linux	Linux, Ubuntu	Linux
Langages de programmation	Java, Python	C, Java	Python	C++, C, Ruby, Java, Shell Scripts
Installation	Facile à installer avec les paramètres par défaut	Dépend de l'environnement réseau et matériel	Difficile à installer plusieurs composants et plusieurs fichiers de configuration	Manuelle, installation facile sur les distributions supportées
documentation	La documentation ne traite pas des questions complexes dans leurs intégralité	Correcte, complète mais pas toujours à jour	Excellente, une documentation officielle disponible et très détaillée	Complète, documentations, références de tous les fichiers de configuration

Tableau 1: Différentes fonctionnalités des plateformes. [29]

CHAPITRE 2: ÉTUDE COMPARATIVE ET CHOIX DE LA SOLUTION

Le tableau 2 ci-dessous, définit la comparaison des caractéristiques des plateformes open Cloud :

	CloudStack	Eucalyptus	OpenStack	OpenNebula
Options d'image disque	Les utilisateurs peuvent gérer leurs Cloud facilement par l'utilisation d'une interface web, outils ligne de commande ou RESTful API	Options fixés par l'administrateur	Glance à une RESTful API	Dans les Cloud privés, la plupart des options libvirt sont ouverts
Stockage d'image disque	iSCSI ou NFS	Wabus, qui imite Amazon S3	Nova	Un système de fichiers partagées, par défaut NFS ou SCP
Hyperviseurs	VMware, KVM, XenServer, Xen Cloud Plateforme (XCP) et hyper-V	Xen, KVM, (VMware en non-open source)	Open virtualisation format(OVF)	Xen, KVM, VMware
Caractéristiques uniques	LVM en cluster, support NetScaler et intégration LDAP	Interface web de gestion des utilisateurs	Système d'authentification unifié	Migration de VM prise en charge

Tableau 2: Comparaison des caractéristiques des solutions Cloud [29].

2.5.1 Présentation d'hyperviseur

Un hyperviseur est une plateforme de virtualisation, qui permet de faire fonctionner plusieurs systèmes d'exploitation sur une même machine physique. Dans notre cas, il permet de créer des instances dans le cloud. L'IaaS est capable de générer des instances ou d'en

CHAPITRE 2: ÉTUDE COMPARATIVE ET CHOIX DE LA SOLUTION

supprimer selon un certain gabarit. Le tableau 3 suivant liste les différents hyperviseurs disponibles pour les différents solutions Cloud Computing [30]

	CloudStack	Eucalyptus	OpenStack	OpenNebula
KVM	√	√	√	√
XenServer	√		√	
Xen	√	√	√	√
EMV	√		√	
LXC	√		√	
VMware	√	√	√	√
Hyper-v	√		√	
Docker			√	
Power KVM			√	
Haute disponibilité d'instances	√		√	√
Auto-scaling d'instances	√	√	√	√

Tableau 3: Les différentes hyperviseurs supportées par les solutions Cloud open source [30].

2.5.2 Réseau

Un ensemble d'instances virtuelles sans connectivité est une infrastructure morte. Certaines solutions IaaS actuellement proposent l'utilisation de SDN (Software-Defined Networking) qui permet de créer un nombre assez important de scénarios réseaux virtuels à l'aide des routeurs virtuels. [30]

Certaines solutions offrent la possibilité de créer des balancements de charge pour les instances, d'autres permettant de créer des accès sécurisés à une infrastructure privée sans point d'accès évident via VPN.

CHAPITRE 2: ÉTUDE COMPARATIVE ET CHOIX DE LA SOLUTION

Le tableau 4 suivant présente une comparaison des fonctionnalités réseaux selon les solutions IaaS.

	CloudStack	Eucalyptus	OpenStack	OpenNebula
Isolation simple(VLAN)	√	√	√	√
SDN(Open Flow/ open vSwitch)			√	√
Support solutions matérielles			√	
Qos			√	
Système de détection d'intrusion			√	
Équilibrage des charges	√		√	
Pare-feu	√		√	√
VPN	√		√	

Tableau 4: Comparaison des fonctionnalités réseaux selon les solutions IaaS. [30]

2.6 Choix de la solution IaaS

Dans cette section, il est question de faire le choix de solution qui convient le mieux. En se basant sur l'étude comparative faite dans ce chapitre, nous avons fixé notre choix sur OpenStack pour les raisons suivantes :

- Disponible librement(gratuit) et une nouvelle version est lancé tous les six mois ; ainsi que la disponibilité de la documentation.
- La conception modulaire d'OpenStack permet une évolutivité souple. Il est aussi possible d'ajouter des ressources pour avoir une interface uniforme pour l'utilisateur finale.
- La solution Open source la plus utilisée. La figure 12 illustre le pourcentage d'utilisation des solutions open source pour la mise en place d'un environnement Cloud où on peut distinguer que la plus grande part de marché utilise OpenStack.

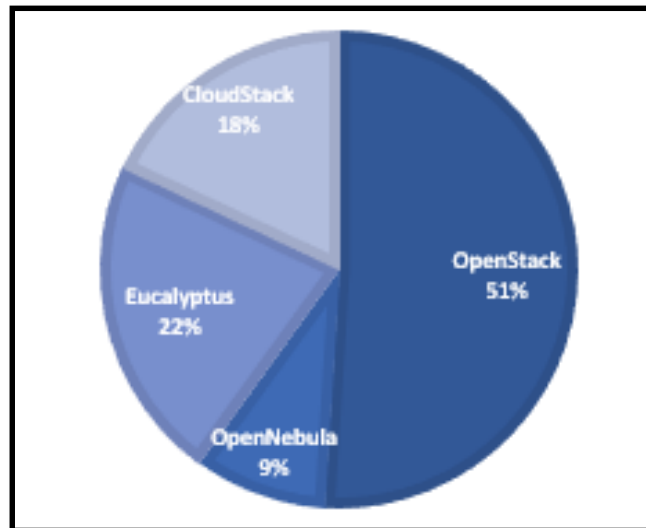


Figure 12: L'utilisation des solutions Open Source dans le marché. [31]

2.7 Étude comparative des hyperviseurs

Après avoir choisi la solution la plus efficace on va maintenant définir quelques hyperviseurs supportés par la solution choisie :

2.7.1 Xen

Est un hyperviseur de type 1 (Bare Metal) qui a débuté comme un projet de recherche à l'université de Cambridge. La société XenSource a été créée et a poursuivi le développement. Il est intégré partiellement dans la partie principale du noyau Linux depuis la version 3.0.

2.7.2 KVM

Kernel-based Virtual Machine est un hyperviseur libre de type 1 (Bare Metal) pour Linux développé par RedHat. KVM est intégré dans le noyau Linux depuis la version 2.6.201.

2.7.3 VSphère/ESXi

VMware vSphere est un logiciel d'infrastructure de Cloud Computing de l'éditeur VMware, c'est un hyperviseur de type 1 (Bare Metal).

2.7.4 Hyper-V

Également connu sous le nom de Windows server virtualisation, est un système de virtualisation basé sur hyperviseur 64 bits de la version de Windows server 2008.

CHAPITRE 2: ÉTUDE COMPARATIVE ET CHOIX DE LA SOLUTION

Le tableau 5 présenté ci-dessous résume les principales caractéristiques de chaque hyperviseur.

	Xen	KVM	vSphere	Hyper-v
Société	XenSource	RedHat	VMware	Microsoft
Développé par	PME personnelle	PME personnelle	Entreprise	Entreprise
VCPUs par hôte	512	512	512	512
VCPUs par VM	8windows/ 32 linux	16	16	4
RAM par hôte	512GB	1TB	1TB	1TB
RAM par VM	32GB	256GB	256GB	256GB
NICs virtuelles	7 par client	4 par client	10 par client	8 synthétique , 4 émulé
Support SE client	CentOS, Debian, RedHat, Windows	RedHat, Windows	La plupart des systèmes d'exploitation x86	RedHat, Windows

Tableau 5: Les principales caractéristiques de chaque hyperviseur [32].

2.8 Choix d'hyperviseur

Nous avons pris le choix d'utiliser l'hyperviseur vSphere car il est globalement plus performant que les autres hyperviseurs, ainsi que le partenariat avec VMware vu qu'on pourra avoir une assistance technique et des conseils si on a besoin.

2.9 Conclusion

Après une étude comparative détaillée on a pu prendre le choix convenable de la solution la plus adéquate pour notre projet.

Dans le chapitre suivant nous allons focaliser sur notre solution choisie "OpenStack".

Chapitre 3: Étude de la solution OpenStack



3.1 Introduction

Dans ce chapitre nous allons présenter la solution OpenStack, ainsi qu'on va définir les composants de l'architecture de celle-ci.

3.2 Présentation d'OpenStack

OpenStack est une plateforme de Cloud Computing Open Source qui supporte tous types d'environnements cloud. Elle fournit une solution d'infrastructure-as-a-service à travers plusieurs services complémentaires où chacun d'eux offre une API qui facilite cette intégration

Le cycle de développement d'OpenStack est de 6mois avec la parution d'une nouvelle version qui peut être améliorée par des points de révisions stables. Le nom de la version correspond au nom de la ville qui accueille les rencontres officielles OpenStack. [33]

Le tableau 6 ci-dessous représente les différentes versions d'OpenStack ainsi que leurs état, la date initiale de sortie, la prochaine phase qu'elles va entamé ainsi que la date de fin de vie .

CHAPITRE 3: ETUDE DE LA SOLUTION OPENSTACK

Séries	Statu	Initial release date	NEXT PHASE	Eol Date
victoria	Développent	2020-10-14 estimatif	Maintien estimatif 2020- 10-14	
Ussuri	Maintien	2020-05-13	2021-11-12	
Train	Maintien	2019-10-16	2021-11-16	
Stein	Maintien	2019-04-10	Maintien étendu estimatif 2020-11-11	
Rocky	Maintien étendu	2018-08-30	Non maintien TBD	
Queens	Maintien étendu	2018-02-28	Non maintien TBD	
Pike	Maintien étendu	2017-08-30	Non maintien TBD	
Ocata	Maintien étendu	2017-02-22	Non maintien Estimé 2020- 05-04	
Newton	Eol	2016-10-06		2017-10-25
Mitaka	Eol	2016-04-07		2017-4-10
Liberty	Eol	2015-10-15		2016-11-17
Kilo	Eol	2015-04-30		2016-05-06
Juno	Eol	2014-10-16		2015-12-07
Icehouse	Eol	2014-04-17		2015-07-02
Havana	Eol	2013-10-17		2014-09-30
Grizzly	Eol	2013-04-04		2014-03-29
Falson	Eol	2012-09-27		2013-11-19
Essese	Eol	2012-04-05		2013-05-06
Diablo	Eol	2011-09-22		2013-05-06
Cactus	Eol	2011-04-15		
Bexar	Eol	2011-02-03		
Austin	Eol	2010-10-21		

Tableau 6: Tableau des versions d'OpenStack. [34]

3.3 Architecture OpenStack

3.3.1 Architecture conceptuelle

OpenStack possède une architecture modulaire qui comprend de nombreux composants. Le diagramme illustré dans la figure 13 montre les relations entre les services OpenStack :

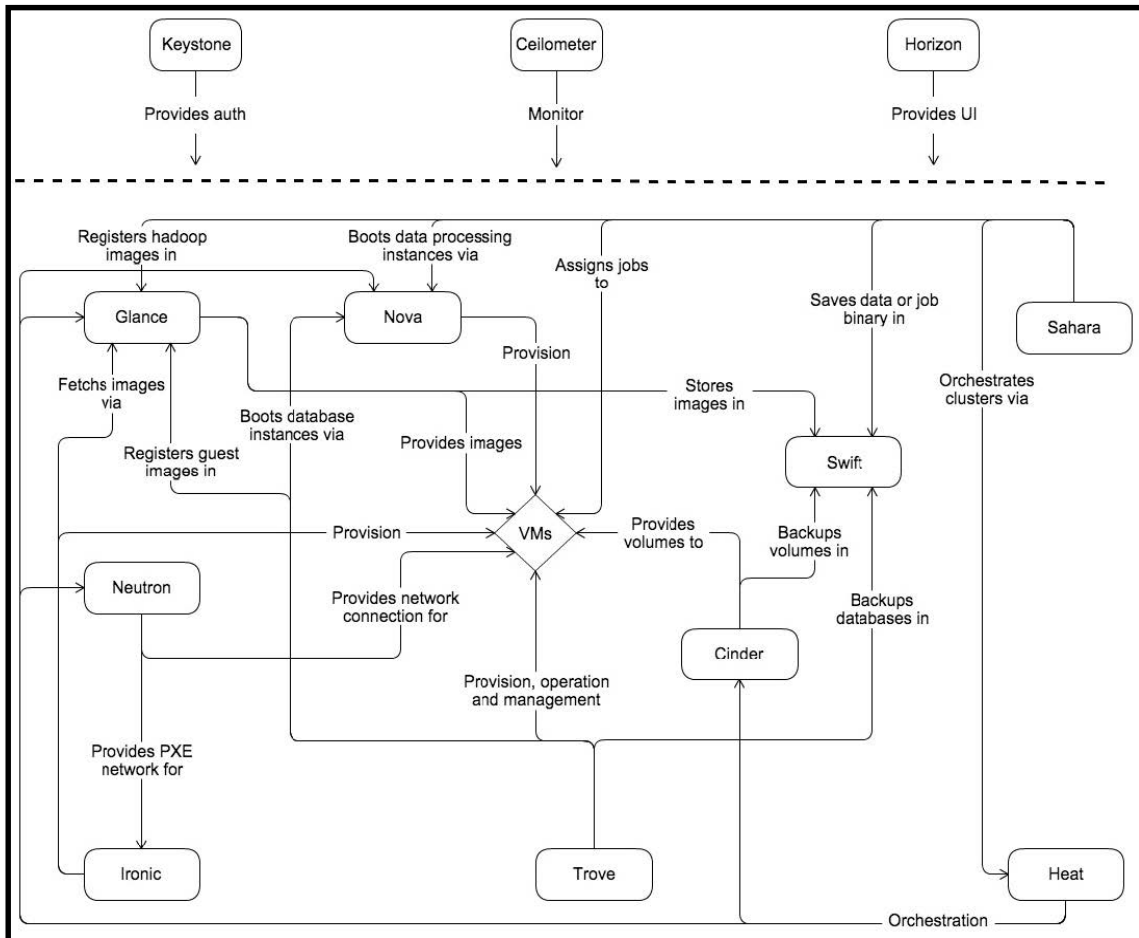


Figure 13: Architecture Conceptuelle d'OpenStack. [35]

3.3.2 Architecture logique

Pour concevoir, déployer et configurer OpenStack, les administrateurs doivent comprendre l'architecture logique qui se présente de suite dans la figure 14 :

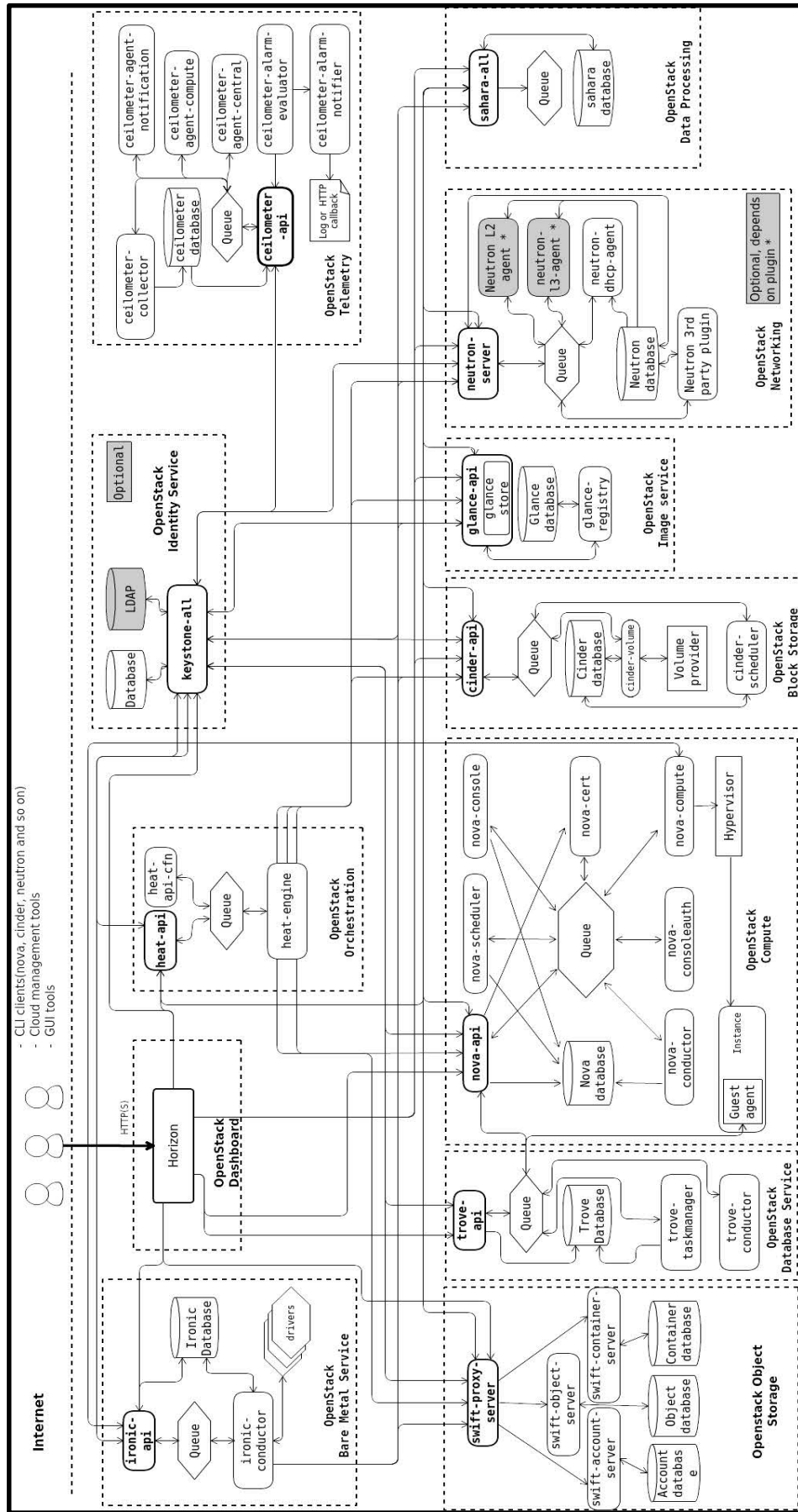


Figure 14: Architecture logique d'OpenStack. [36]

3.4 Composants OpenStack

OpenStack possède une architecture modulaire qui comprend plusieurs composants où chacun a une fonction particulière offrant un service particulier illustré dans la figure 15 ci-dessous.

Le lancement d'une instance c'est-à-dire machine virtuelle nécessite de nombreuses interactions entre plusieurs services.

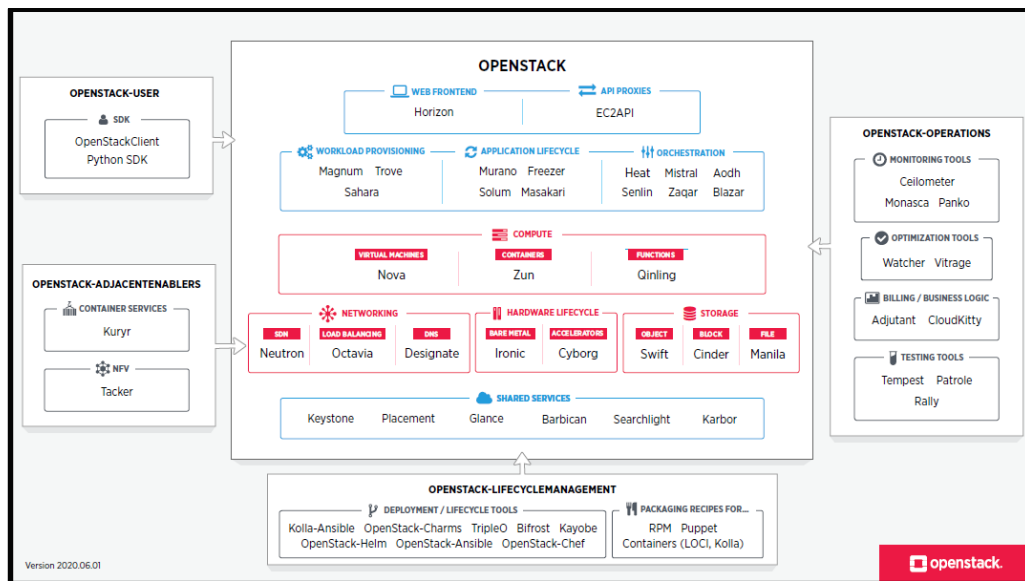


Figure 15: Services et fonctions des projets OpenStack. [33]

3.4.1 Horizon – Tableau de bord

Le Dashboard OpenStack, également appelé horizon est une interface web qui permet aux administrateurs et aux utilisateurs de Cloud de gérer les différentes ressources et services OpenStack (lancement d'une instance, attribution d'adresse IP, définition des contrôles d'accès etc...). Cette interface web est développée en python, basée sur le Framework Django, tire parti des API REST et elle utilise le module WSGI d'Apache. [37]

3.4.2 Keystone – Identité

Keystone est un Framework d'authentification et d'autorisation pour tous les services OpenStack. Ce service gère les demandes API et fournit des services d'identité configurables de multiples formes y compris le standard de nom, mot de passe et les systèmes à base de jetons. [38]

L'authentification vérifie que la demande provient en réalité de celui qu'il prétend être. L'autorisation consiste à vérifier si l'utilisateur authentifié a accès à des services auxquels il souhaite accéder. Le service d'identité peut également s'intégrer à certains systèmes de gestion d'utilisateurs externes (tels que LDAP).

La figure 16 montre le fonctionnement de Keystone ; ce dernier s'appuie sur un SGBD soit de type SQLite, MYSQL ou PostgreSQL et utilise par défaut le port 35357.

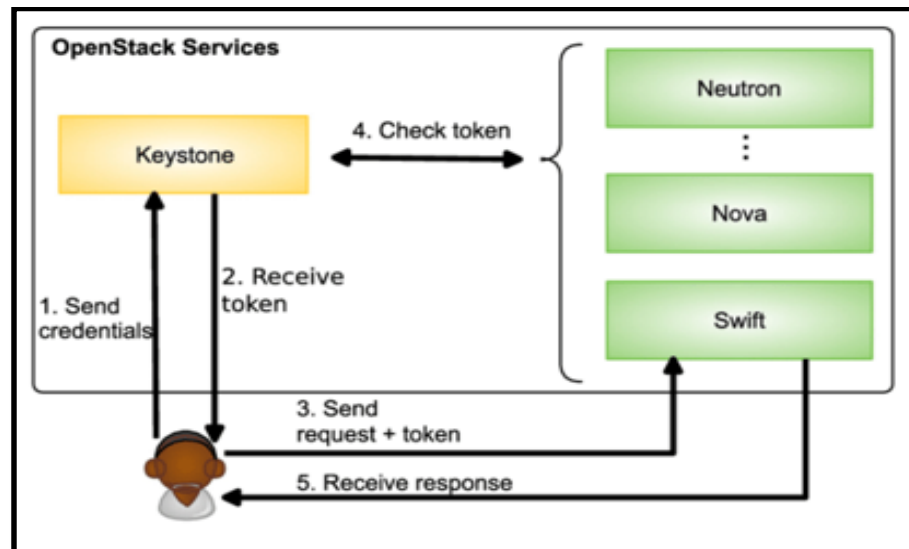


Figure 16: Le fonctionnement d'OpenStack Keystone. [39]

3.4.3 Nova – Calcul (Compute)

Le composant principal le plus compliqué et le plus distribué d'OpenStack, Nova, assure les opérations de calcul et de traitement au sein du cloud. Ce module offre les fonctionnalités proposées par EC2 d'AWS. Il permet de définir les groupes de sécurité pour effectuer les contrôles d'accès, les keys paires pour accéder aux instances et choisir la taille de ces dernières à partir des flavors, en associant aussi un nombre de VCPUs, de l'espace de stockage et une quantité de RAM. Tout cela à partir de la ligne de commande où à partir du tableau de bord du composant horizon. [40]

OpenStack Compute contrôle les hyperviseurs par l'intermédiaire de la "libvirt" ou directement par les API de certains hyperviseurs pris en charge tels que KVM, QEMU, LXC, Xen Server, Hyper-v et VMware. Il repose sur le langage python, les Framework Tornado et Twisted ainsi que le protocole ouvert pour les systèmes de messagerie AMQP. Nova utilise par défaut le port 8774.

Ce nœud est constitué de différents processus mentionnés ainsi que leurs fonctions :

- **Nova-api** : c'est un service web API RESTful qui accepte les commandes entrantes pour interagir avec le Cloud OpenStack, il initie la plupart des activités d'orchestration.
- **Nova-metadata-api** : accepte les requêtes métadonnées des instances, ce service généralement utilisé à l'exécution du mode multi-hôtes avec les installations nova-network.
- **Nova-Compute** : c'est un daemon de travail (worker) qui gère les instances de machine virtuelle via les API d'hyperviseurs. Par exemple :
 - Xen API pour Xen Server/XCP.
 - Libvirt pour KVM ou QEMU.
 - VMware API pour VMware.
- **Nova-placement-api** : suit l'inventaire et l'utilisation de chaque fournisseur.
- **Nova-Scheduler** : il reçoit une requête de la file d'attente et détermine l'hôte sur lequel il doit fonctionner.
- **Nova-conductor** : il fournit des services de nova calcul, tels que la mise à jour des bases de données et la gestion des tâches de longues durées.
- **Nova-cert** : démon de travail qui sert les certificats x509, utilisé pour générer les certificats pour euca-bundle-image. Utilisé seulement pour API d'EC2.
- **Nova-consoleauth** : un démon qui gère l'autorisation des jetons pour les utilisateurs.
- **Nova-novncproxy** : un démon qui fournit un proxy pour l'accès aux instances en cours d'exécution via une connexion VNC. Prend en charge les clients NOVNC basé sur un navigateur.
- **Nova-spicehtml5proxy** : fournit un proxy pour l'accès aux instances en cours d'exécution via une connexion SPICE. Prend en charge les clients HTML5 basé sur un navigateur.
- **Nova-XVNCProxy** : fournit un proxy pour l'accès aux instances en cours d'exécution via une connexion SPICE. Prend en charge un client Java spécifique à OpenStack.
- **La queue** : un hub central pour passer des messages entre les démons, habituellement implémenté avec RabbitMQ, peut également être implémenté avec une autre file d'attente de messages AMQP telle que ZeroMQ.

- **Base de données SQL** : stocke la plupart des états de construction et d'exécution d'une infrastructure Cloud, notamment :
 - Types d'instance disponibles.
 - Instances en cours d'utilisation.
 - Réseaux disponibles.
 - Projets.

La figure 17 montre l'architecture d'OpenStack Nova ainsi que tous ces composants.

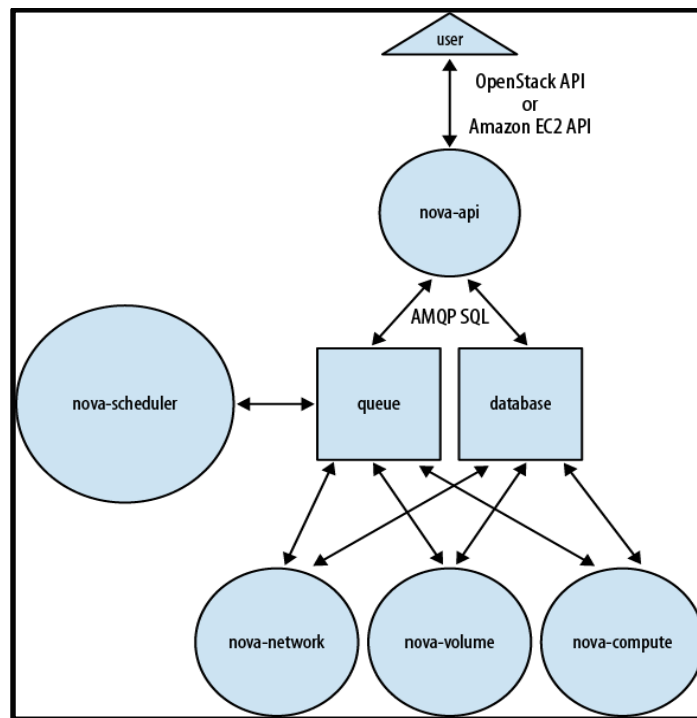


Figure 17: Architecture d'OpenStack Nova. [41]

3.4.4 Glance- image

Ce module fournit des services de découverte, d'enregistrement et de livraison pour les images de disque vers les instances. Il stocke 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 stockages décentralisés comme Ceph. Ce service ne stocke pas seulement des images mais aussi des informations sur celles-ci, les métadonnées, qui sont par exemple le format du disque (comme Qcow2 ou Raw) ou les conteneurs de celle-ci (OVF par exemple). [42]

Glance fournit également une API compatible REST permettant d'effectuer des requêtes dans une base de données pour récupérer les informations sur les images disques des machines virtuelles. Il s'agit des images d'OS (UBUNTU, Cirros, Windows, etc...) à partir desquelles on peut démarrer les instances .il s'agit de la même brique que les AMIs d'AWS.

OpenStack Imaging service se décline en sous composants suivants :

- **Glance-api** : il accepte les appels d'API image pour la découverte, la récupération et le stockage d'image. Par défaut, glance-api écoute sur le port 9292.
- **Glance-registry** : stocke, traite et récupère les métadonnées sur les images (taille, type, etc...). Par défaut, glance-registry écoute sur le port 9191.
- **Glance data base** : une base de données pour stocker les métadonnées des images.

Glance prend en charge les systèmes de fichiers normaux, les périphériques de blocs RADOS, Amazon s3, http et Swift. La figure 18 montre l'architecture d'OpenStack Glance ainsi que tous ces composants.

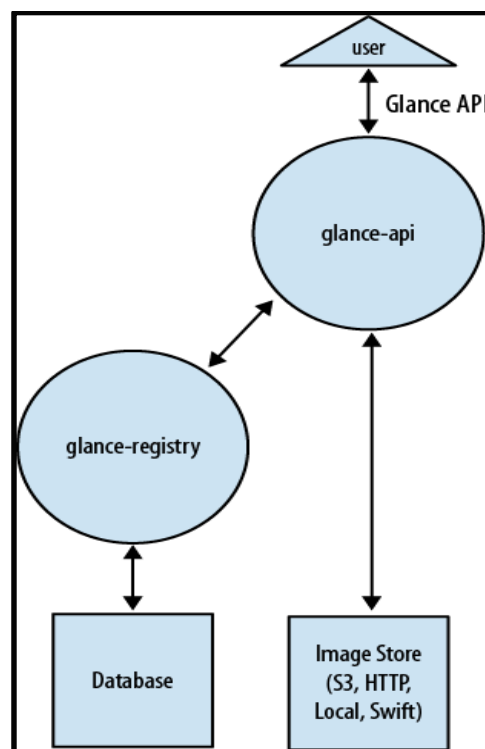


Figure 18: Architecture d'OpenStack Glance. [43]

3.4.5 Neutron – réseau

Le service Neutron d'OpenStack permet de gérer et manipuler les réseaux et l'adressage IP au sein d'OpenStack (statique ou par l'intermédiaire du service DHCP ; ainsi que le service d'adresses IP flottante). Les utilisateurs peuvent créer leurs propres réseaux, contrôler le trafic à travers des groupes de sécurité et connecter leurs instances à un ou plusieurs réseaux. [44]

Ce module fournit différents types de déploiement réseau en fonction de l'infrastructure cible tels que les réseaux plats (flat network), les réseaux VLAN, VXLAN ou à tunnel GRE.

Neutron gère ses déploiements grâce aux modules complémentaires permettant la communication aux équipements ou logiciels de gestion. Les plug-ins les plus utilisés sont Open vSwitch, ML2, linux Bridge mais aussi CISCO NEXUS, juniper, Open Contrail et d'autres. [33]

Dans son architecture, neutron a été construit en suivant les réseaux de nouvelle génération dites SDN. Il utilise par défaut le port 9696, et se décline en plusieurs sous composants :

- **Neutron-server** : il s'agit du démon qui expose aux autres composants l'API neutron et va passer les requêtes utilisateurs vers les autres démons. Il agit comme le contrôleur du réseau.
- **Neutron-plugin agent** : fonctionne sur chaque nœud de traitement pour gérer la configuration du commutateur virtuel local (vSwitch). Le plug-in que vous utilisez détermine les agents qui s'exécutent. Ce service nécessite l'accès à la file d'attente des messages et dépend du plug-in utilisé. Certains plug-ins comme OpenDaylight (ODL) et Open Virtuel Network (OVN) ne nécessitent aucun agent python sur les nœuds de calcul.
- **Neutron-DHCP agent** : fournit des services DHCP aux réseaux de locataires. Cet agent est le même pour tous les plug-ins et est responsable de la gestion de la configuration.
- **Neutron- L3 agent** : fournit le transfert L3 /NAT pour l'accès réseau externe des machines virtuelles sur les réseaux de clients. Nécessite l'accès à la file d'attente des messages en option selon le plug-in.
- **Network-provider-service (SDN)** : fournit des services de mise en réseau supplémentaires aux réseaux de locataires. Ces services SDN peuvent interagir avec des agents neutron-server, neutron-plugin-in et plug-in-agent via des canaux de communication tels que les API REST.

Le fonctionnement d'OpenStack Neutron est montré dans la figure19.

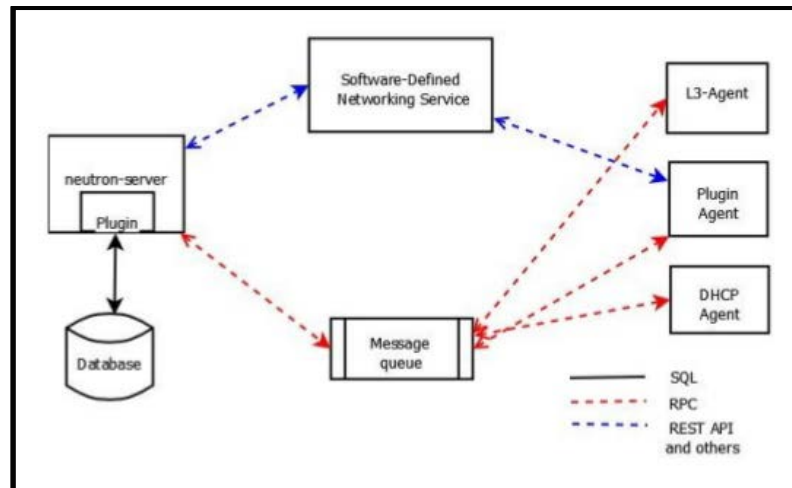


Figure 19: Architecture d'OpenStack Neutron. [45]

Neutron interagira principalement avec Nova, où il fournira des réseaux et la connectivité pour ses instances.

3.4.6 Cinder- bloc de stockage

Ce module fournit un stockage persistant en mode block. Il gère les opérations de création, d'attachement et de détachement de ces blocs sur les serveurs. En plus de stockage local sur le serveur, Cinder peut utiliser de multiples plateformes de stockage tel que Ceph, EMC (Scabilo, VMAX et VNX), GlusterFS, HitachiData Systems, IBM Storage (Storwise Family, SAN Volume Controller, XIV Storage System, et GPFS), Net App, HP (Store Virtual et 3PAR) et bien d'autres. [46]

Cinder donne la possibilité d'ajouter des volumes réseaux aux instances démarrées. Il s'agit d'un disque (Raw device) attaché à une instance et une seule à un instant donné. Il faut formater ce disque pour le monter et l'utiliser sur l'instance. Il s'agit de la même fonctionnalité que les EBS d'AWS. Ce service gère aussi la création d'instantanés (Snapshot), très utile pour sauvegarder des données contenues dans les périphériques de type bloc. [33]

OpenStack bloc Storage se décline en plusieurs sous composants :

- **Cinder-api** : accepte les requêtes et les achemine vers cinder-volume pour l'action.
- **Cinder-volume** : répond aux demandes de lecture et d'écriture à la base de données Cinder pour maintenir l'état, interagit avec d'autres processus (comme Cinder-Scheduler) via une file d'attente de messages.

- **Cinder-Scheduler** : sélectionne le nœud de stockage de bloc optimal pour créer le volume. Ce composant est similaire au Nova-Scheduler.
- **Cinder-backup daemon** : fournit des volumes de sauvegarde de tout type à un fournisseur de stockage de sauvegarde. Comme le service Cinder-Volume, il peut interagir avec une variété de fournisseurs de stockage via une architecture de pilote (driver architecture).
- **Messaging queue** : ou file d'attente de messageries, achemine les informations entre les processus de stockage de bloc.

Comme Neutron, Cinder interagira principalement avec Nova en fournissant des volumes pour ses instances. L'architecture de Cinder est montrée dans la figure 20 ci-dessous.

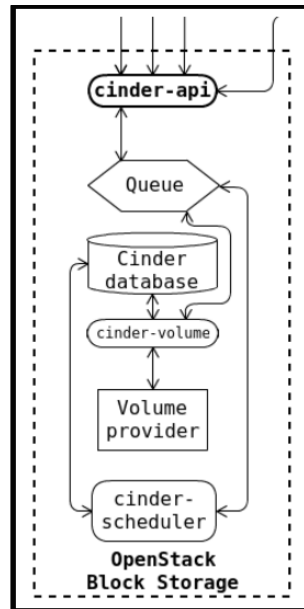


Figure 20: Architecture d'OpenStack Cinder. [47]

3.4.7 Swift- Stockage d'Objets

OpenStack Object Storage est un système de stockage d'Objet multi-locataires. Il est hautement évolutif et peut gérer de grandes quantités de données non structurées à faible coût via une API http RESTful qui peut être intégrée directement dans des applications ou utilisée pour la sauvegarde, l'archivage et la rétention de données. [48]

Ce système de fichier est conçu pour le stockage à long terme de gros volumes. Il utilise une architecture distribuée offrant plusieurs points d'accès pour éviter les SPOF (Single Point Of Failure) qui entraînent l'arrêt complet du système lors d'une panne. [33]

Swift comprend les composants suivants :

- **Swift-proxy-server** : accepte les demandes entrantes, comme les fichiers à télécharger, les modifications de métadonnées ou la création des conteneurs ; il sert également les fichiers et les listes de conteneurs aux navigateurs web. Pour améliorer les performances, le serveur proxy peut utiliser une cache facultative généralement déployée avec "Memcache".
- **Swift-account-server** : gère les comptes définis avec storage d'objet.
- **Swift-container-server** : gère le mappage des conteneurs ou des dossiers, dans storage d'objet.
- **Swift- Object-Server** : gère les objets réels, tels que les fichiers, sur les nœuds de stockage.
- **Processus périodiques divers** : effectue des tâches de maintenance sur le grand magasin de données. Les services de réplication garantissent la cohérence et la disponibilité à travers le cluster.
- **Client Swift** : permet aux utilisateurs de soumettre des commandes à l'API REST via un client de ligne de commande autorisé en tant qu'utilisateur administrateur, utilisateur revendeur ou utilisateur swift.
- **Swift-init** : script qui initialise la construction du fichier en anneau, prend les noms de démons comme paramètre et propose des commandes.
- **Swift- recon** : est un outil CLI utilisé pour récupérer diverses métriques et informations de télémétrie sur un cluster, qui ont été collectées par le middleware Swift-con.
- **Swift-ring-builder** : utilitaire de création et de rééquilibrage d'anneaux de stockage.

La figure 21 suivante montre l'architecture du composant OpenStack Swift ou on peut voir les sous composants principales.

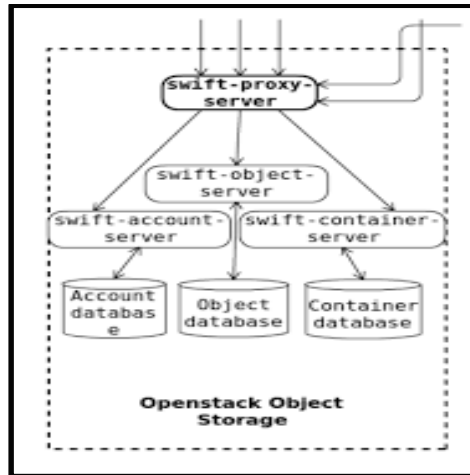


Figure 21: Architecture d'OpenStack Swift. [47]

3.5 Conclusion

Dans cette section nous avons pu définir la solution OpenStack ainsi que ses composants et le fonctionnement de ceux-ci.

Dans la prochaine Partie, nous allons entamer la phase conception, implémentation et test de la solution adapté.

PARTIE 2

**Conception, implémentation et test
de la solution adaptée.**

Chapitre 4: Conception de la solution

4.1 Introduction

La réalisation de ce projet doit être précédé d'une conception pour pouvoir formaliser les traits et les étapes de réalisation de notre système. Pour cette conception nous allons utiliser le langage UML pour spécifier le fonctionnement du système. Mais avant cela nous allons d'abord spécifier nos besoins.

4.2 Spécification des besoins

On va analyser dans cette partie les différents besoins fonctionnels et non fonctionnels de notre cloud.

4.2.1 Besoins fonctionnels

Cette partie, est pour détailler l'ensemble de fonctionnalités que le cloud offre a ces utilisateurs.

4.2.1.1 Gestion d'images

On parle d'images disques stockées par le service Glance. L'utilisateur pourrait consulter la liste des images autorisées pour les projets, les éditer. Aussi il sera possible de lancer des nouvelles instances de cette image, créer une nouvelle ou supprimer une existante.

4.2.1.2 Gestion d'instances

Une instance est une machine virtuelle en cours d'exécution ou dans un état connu comme « suspendue » qui peut être utilisé comme un serveur matériel. L'utilisateur pourrait consulter la liste d'instances des machines virtuelles actuelles plus quelques informations globales comme le projet auquel elles appartiennent, le serveur hôte, l'adresse IP, la taille, le statut et les actions en cours. Il aurait aussi les possibilités d'éditer, mettre fin, pause, redémarrer ou supprimer une instance. Aussi Il pourrait se connecter à la console VNC de l'instance ou créer une nouvelle.

4.2.1.3 Gestion des volumes

Le nuage permettrait à l'utilisateur de consulter la liste des volumes disques virtuels existants, la création d'un nouveau volume et la modification d'un ancien.

4.2.1.4 Gestion des flavors

Un flavor est une configuration de matériel disponible dans un serveur. Chaque Flavor possède une combinaison unique d'espace disque et la capacité de mémoire. L'utilisateur

pourrait consulter la liste des types d'instances disponibles, leurs spécifications en nombre de CPUs, mémoire, espace disque et créer des nouvelles définitions d'instance.

4.2.1.5 Gestion des projets

Un projet est un groupement logique des utilisateurs au sein de Nova, utilisé pour définir les limites des ressources pour ce projet et l'accès aux images des machines virtuelles. Il serait possible de consulter les projets existants et leur statut et de créer des nouveaux projets.

4.2.1.6 Gestion des utilisateurs

L'utilisateur aurait la possibilité de consulter la liste d'utilisateurs enregistrés, avec la possibilité d'ajouter ou d'éditer les détails mais pas d'ajouter l'utilisateur à plusieurs projets.

4.2.1.7 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, les groupes de règles de pare-feu et leur interface d'édition et enfin la liste des clés SSH avec l'import ou la création de certificat.

4.2.2 Besoins non fonctionnels

- Simplicité d'un service à la demande.
- Flexibilité.
- Accès léger.
- Sûreté.

4.3 Identification des acteurs

Comme on traite la mise en place d'un cloud privé interne où les services Cloud sont hébergés par l'entreprise. On présente deux types d'acteurs pour notre système : un administrateur et un utilisateur.

- **L'administrateur** : c'est toute personne physique ayant reçu les droits d'administration. Ces droits sont configurés lors d'installation du premier administrateur.
- **L'utilisateur** : c'est toute personne physique de l'entreprise ayant reçu un compte.

Chaque utilisateur possède un login et un mot de passe unique, modifiable par le concerné.

4.4 Spécification technique (Environnement matériel)

La couche IAAS du cloud Computing comprend trois parties essentielles :

- Partie stockage SAN qui comprend des baies.
- Partie réseau qui regroupe des routeurs, commutateurs et des firewalls.
- Partie compute qui est constitué des châssis regroupant des serveurs blades.

4.4.1 Partie stockage

Le SAN est une technologie de stockage en réseau qui fournit l'espace disque rapide et fiable. C'est un réseau physique en fibre optique, il connecte l'ensemble des unités de stockages et des serveurs. Dans ce réseau, les données stockées sont routées et structurées via des commutateurs FC. Cette technologie est basée sur le protocole Fibre Channel, qui autorise le transfert de données entre périphériques sans surcharger les serveurs.

4.4.1.1 Baie de stockage

Une baie de stockage est un équipement de sauvegarde de données informatiques qui comporte principalement un ensemble de disques, permettent d'emmagasiner et de gérer de grandes quantités de données généralement à travers un réseau de stockage dite SAN.

Les baies de stockage utilisent différentes techniques d'agrégat de disques, nommées RAID qui gèrent la cohérence et la répartition des données sur plusieurs disques durs. Les disques qui existent sur le marché sont : FC, SATA, SAS mais le meilleur c'est le FC.

Les baies utilisent aussi des protocoles de stockage comme iSCSI ou FC. Mais ce dernier est le plus performant et il peut aller jusqu'à 10GB/s.

4.4.2 Partie compute

4.4.2.1 Châssis

Le châssis est un équipement qui héberge un ensemble de serveurs lames et fournit une source d'alimentation électrique unique pour ces serveurs en mutualisant plusieurs unités d'alimentations électriques, assurant ainsi une redondance et permettant une tolérance aux pannes. Les connexions réseau sont incluses dans le châssis. Cela permet de connecter un serveur lame à différents supports physiques (paire torsadée ou fibre optique) et de mettre en place des configurations avancées (agrégation de ports). Chaque châssis peut contenir un certain

nombre de switch internes. Mais généralement il intègre six switches: quatre switches ETH et deux switches FC.

4.4.2.2 Serveurs blades

Un serveur lame ou blade est un serveur de la taille d'une carte d'extension PCI, intégrant processeur, mémoire vive, interface réseau et disque dur, dont la compacité simplifie la gestion de l'espace, économise la consommation d'énergie, et autorise l'installation d'un grand nombre de serveurs. Tenant sur une simple carte PCI, le serveur lame permet de ranger dans un seul châssis des dizaines de serveurs. Chaque lame est un serveur à part entière, souvent dédié à une seule application.

En effet, chaque lame à six connectiques Réseau (3 carte bi-port) :

- **Une carte pour l'administration des blades** : une path sur ETH 1 et l'autre sur ETH 2 de châssis.
- **Une carte pour le LAN** : une path sur ETH 3 et l'autre sur ETH 4.
- **Une Carte pour le stockage** : une path sur FC 1 et l'autre sur FC 2.

Nous notons que la couche de virtualisation s'installe sur les serveurs blades, tel que chaque blade héberge un certain nombre de machines virtuelles.

4.4.3 Partie réseau :

Cette partie est composée de:

- Un serveur OpenStack dans lequel est déployé OpenStack et qui a comme rôle la gestion du nuage.
- Serveurs reliés entre eux par un Switch.
- Les postes de développeurs de la Société.
- Une instance de réseau privé de la société destiné aux communications entre les VMs.

4.5 Architecture de Solution Openstack

Trois éléments interagissent avec tous les composants du système. Horizon est l'interface graphique que les administrateurs peuvent plus facilement utiliser pour gérer tous les projets. Keystone gère la gestion des utilisateurs autorisés, et Neutron (Quantum) définit les réseaux qui fournissent une connectivité entre les composants.

Nova peut sans doute être considérée comme l'OpenStack de base. Il gère l'orchestration des charges de travail. Ses instances de calcul nécessitent généralement une certaine forme de stockage persistant qui peut être soit à base de blocs (Cinder) ou orienté objet (Swift). Il nécessite également une image pour lancer une instance et pour gérer cette demande, de sorte qu'il peut éventuellement utiliser Swift que pour le stockage back-end.

L'architecture en exemple montrée dans la figure 22 nécessite au moins deux nœuds (hôtes) pour lancer une simple machine virtuelle ou instance. Les services optionnels comme le Stockage par Blocs et le Stockage Objet nécessite des nœuds additionnels.

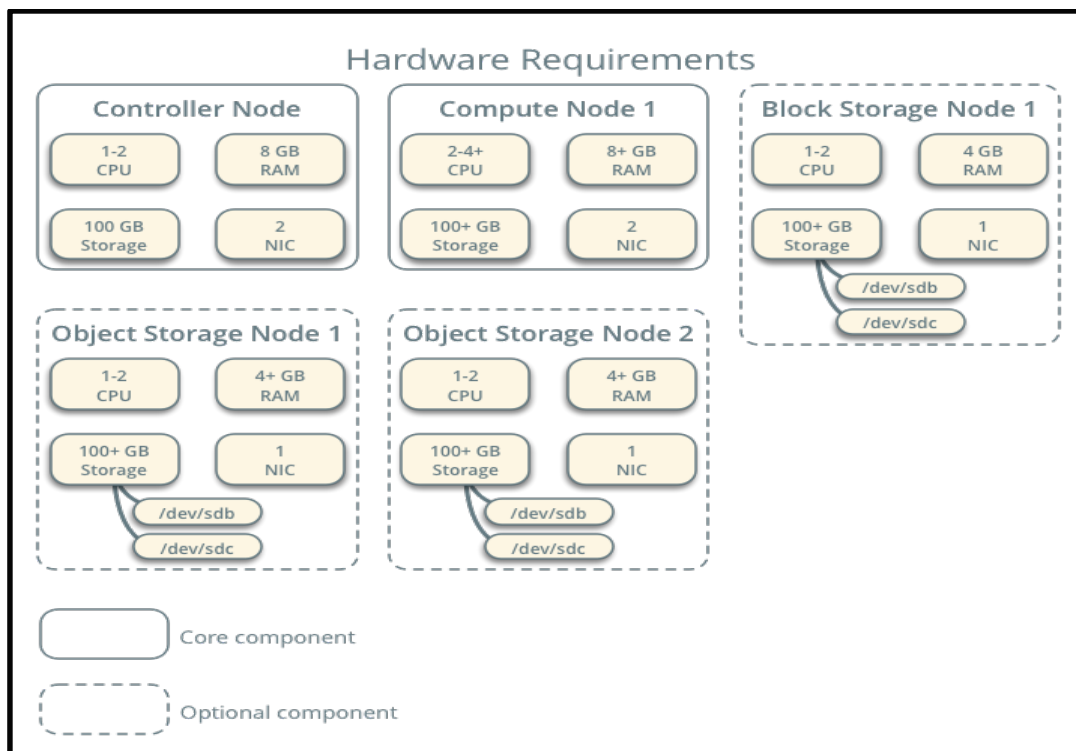


Figure 22: Architecture de solution. [49]

4.5.1 Réseau Option 1 : Réseaux fournisseurs

L'option de réseaux fournisseurs montrée dans la figure 23 ou elle déploie le service Réseau d'OpenStack de la façon la plus simple possible avec essentiellement les services de couche-2 (bridging/switching) et une segmentation des réseaux en VLAN. Principalement, il fait le lien (bridge) entre les réseaux virtuels et les réseaux physiques et s'appuie sur l'infrastructure réseau physique pour les services de couche-3 (routing). De plus, un service DHCP fournit les informations d'adresse IP aux instances. [49]

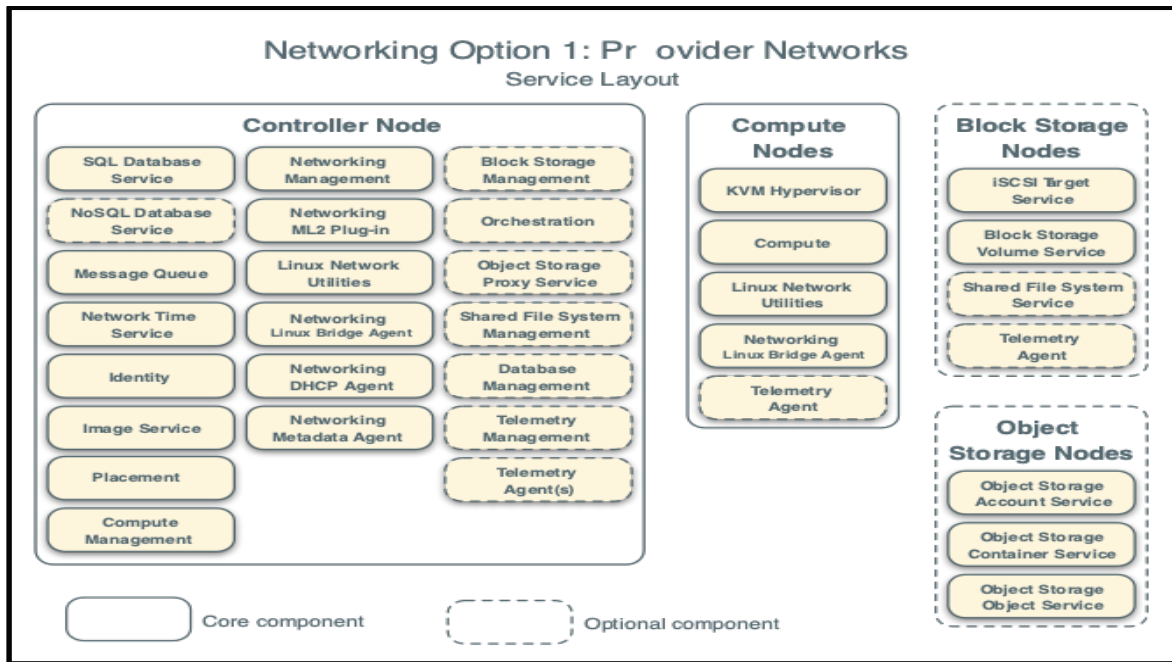


Figure 23: Option 1 de réseaux solution: Réseaux fournisseurs. [49]

4.5.2 Réseau Option 2 : Réseaux libre-service

L’option de réseaux libre-service améliore l’option de réseaux fournisseurs avec des services de couche-3 (routing) qui permettent la création de réseaux libre-service utilisant des techniques de segmentation overlay comme VXLAN montrée dans la figure 24.

Essentiellement, cela permet de router les réseaux virtuels vers les réseaux physiques via le NAT. De plus, cette option sert de base aux services avancés comme LBaaS et FWaaS.

[49]

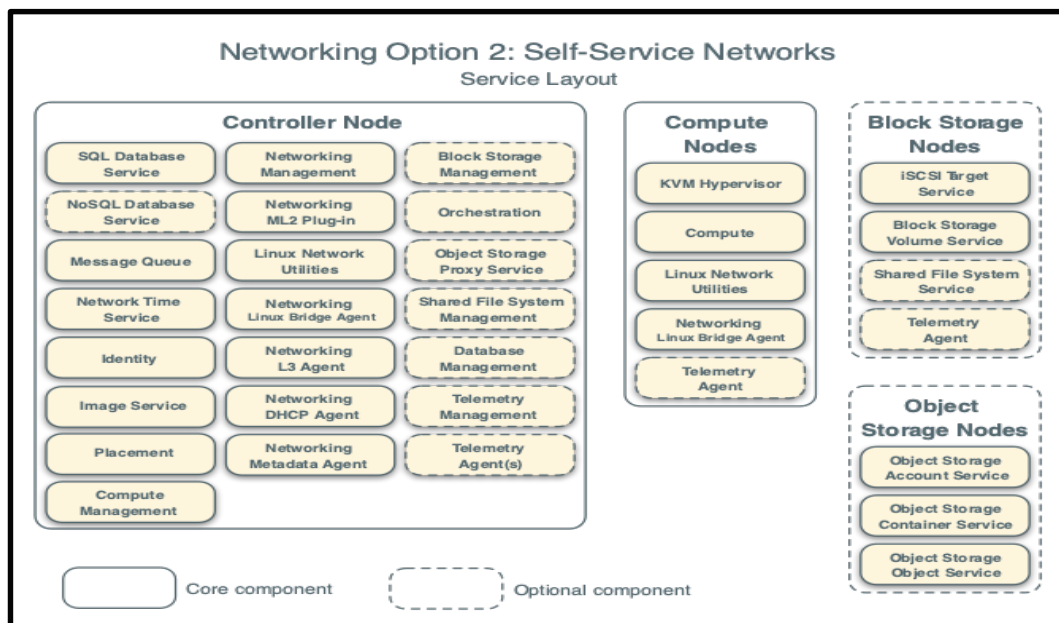


Figure 24: Option 2 de réseaux solution: Réseaux libre-service. [49]

4.6 Diagramme de cas d'utilisation global :

Un diagramme de cas d'utilisation est un diagramme UML utilisé pour décrire le comportement du système, c'est-à-dire l'interaction entre ce dernier et l'acteur qui interagit avec.

Un acteur représente le rôle d'entité externe qui interagit directement avec le système étudié. Le diagramme global montré dans la figure 25 regroupe tous les fonctionnalités et les utilisations de ce système pour nos acteurs définis précédemment.

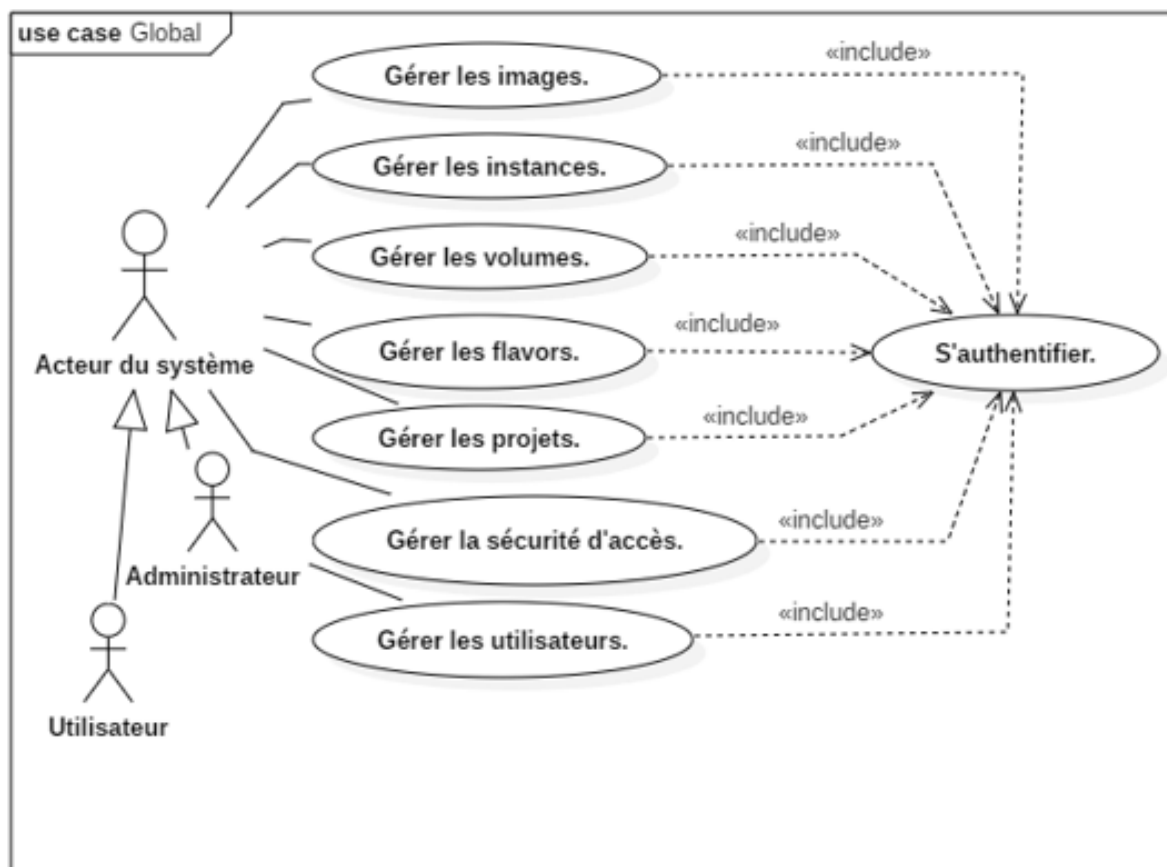


Figure 25: Diagramme de cas d'utilisations general.

Un utilisateur ne pourrait consulter et gérer que les ressources des projets auxquels il appartient, ainsi il serait possible pour lui de :

- **Consulter les états de ces projets :** Un utilisateur pourrait consulter l'usage des serveurs d'un projet sélectionné, utilisation actuelle en nombre de CPU virtuels, RAM et Disques puis compteur en CPU et espace disque (GB) par heures.

- **Gérer les instances et les volumes :** Un utilisateur pourrait consulter la liste des instances existantes et aurait la possibilité de les éditer, de créer ou de modifier des volumes disques virtuels.
- **Gérer les images et leurs instances :** Un membre aurait la possibilité de consulter la liste des images autorisées pour le projet et lancer de nouvelles instances.
- **Gérer la sécurité et l'accès :** Un membre pourrait consulter la liste des adresses IP disponibles pour connecter les instances au réseau public avec la possibilité de création des groupes de règles et de Pare-feu. Il aurait aussi la possibilité de consulter la liste des clés SSH et de créer de certificat.

4.7 Diagramme de séquence

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Dans un l'objectif de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droit du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets. Les périodes d'activité des classes sont symbolisées par des rectangles.

4.7.1 Diagramme de séquence « Créer projet »

Un projet OpenStack est peut-être l'élément le plus essentiel pour notre conception, il nécessite une gestion par l'administrateur du système ou par le client qui est l'utilisateur de ce système. Pour la gestion de projet on peut créer, mettre à jour et supprimer un projet comme on peut pour un même projet par l'administrateur l'ajout d'un nouvel utilisateur, le supprimer le mettre à jour et activer/ désactiver des permissions ou des fonctionnalités. Dans ce diagramme montré dans la figure 26 nous allons présenter la fonction principale qui est la création d'un projet.

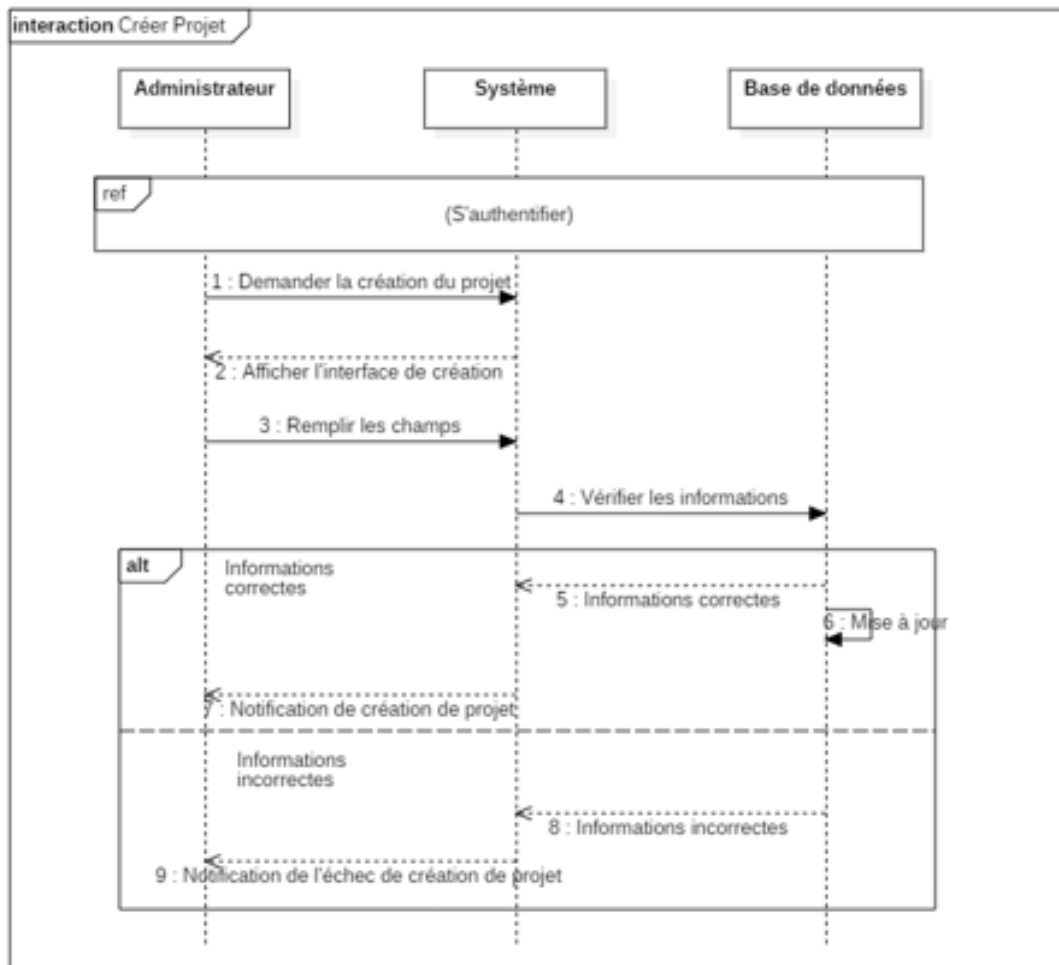


Figure 26: Diagramme de séquence Créer projet.

4.7.2 Diagramme de séquence « Déployer VM »

Après la création d'un projet, le déploiement d'une machine virtuelle est ainsi nécessaire. Une machine virtuelle aussi dite instance est déployer dans le Cloud, et peut être lancé depuis plusieurs sources d'images de système d'exploitation. Pour la création d'instance on doit spécifier (le nom, la zone de disponibilité, source de démarrage d'instance, flavor qui est la taille d'instance, le réseau à qui elle va être connecté ainsi que les ports, les groupes de sécurité, paire de clé pour le chiffrement, personnalisation du script de source, et des métadonnées s'ils sont disponibles. On peut aussi suivre l'utilisation d'une instance, créer des snapshots et gérer les instances. Dans ce diagramme montré dans la figure 27 on va présenter le déploiement d'une machine virtuelle.

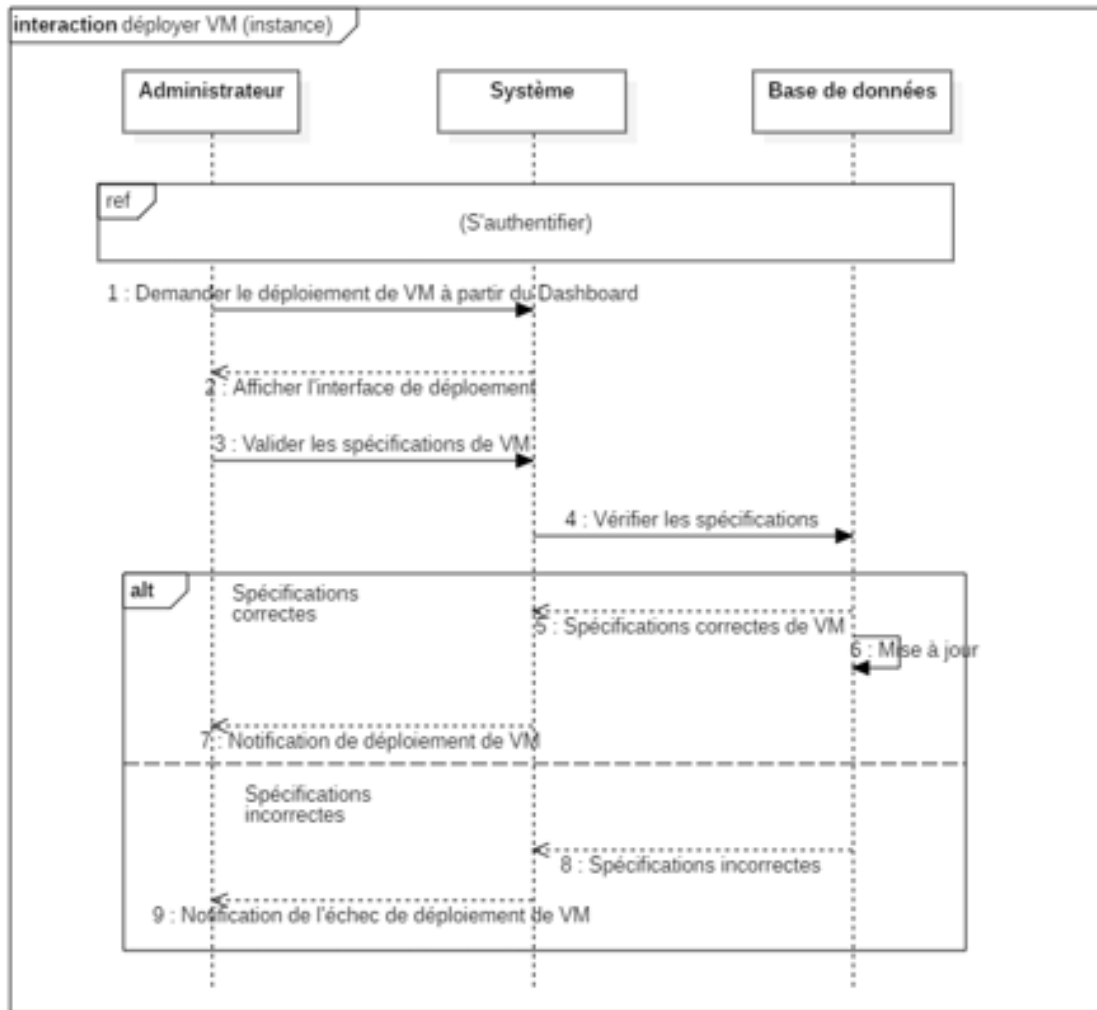


Figure 27: Diagramme de séquence Déployer une VM.

4.8 Conclusion

À la fin de ce chapitre nous avons pu voir la conception faite ainsi que l'architecture de la solution choisi pour pouvoir l'implémenter.

Dans le chapitre qui vas suivre, on va parler sur les principales étapes et les configurations faites afin de mettre en œuvre notre solution.

Chapitre 5 : Implémentation et test de la solution OpenStack

5.1 Introduction

Dans ce chapitre qui est l'implémentation et test de la solution OpenStack, nous nous intéressons à la présentation des grandes étapes faites pour l'acquisition de notre solution et ensuite la testé pour montrer que c'est le meilleur choix.

5.2 Installation de l'hyperviseur ESXi

VMware ESXi est un hyperviseur de type 1 indépendant des systèmes d'exploitation. Il repose lui-même sur le système d'exploitation VMkernel qui assure l'interface avec les agents dont il soutient l'exécution. ESXi est l'hyperviseur exclusif des licences VMware vSphere 5.x.

Les étapes pour installer VMware vSphere ESXi6.7 sont les mêmes que pour les versions précédentes.

Une fois le programme d'installation extrait les fichiers on se trouve devant l'interface montré dans la figure 28, ou on vas attendre le téléchargement d'installation de l'hyperviseur ESXi.



Figure 28: Chargement d'installation d'ESXi.

Une fois le programme est chargé on reçoit ce message de bienvenue comme indiqué ci-dessous dans la figure 29, on presse sur entrée du clavier pour continuer l'installation.

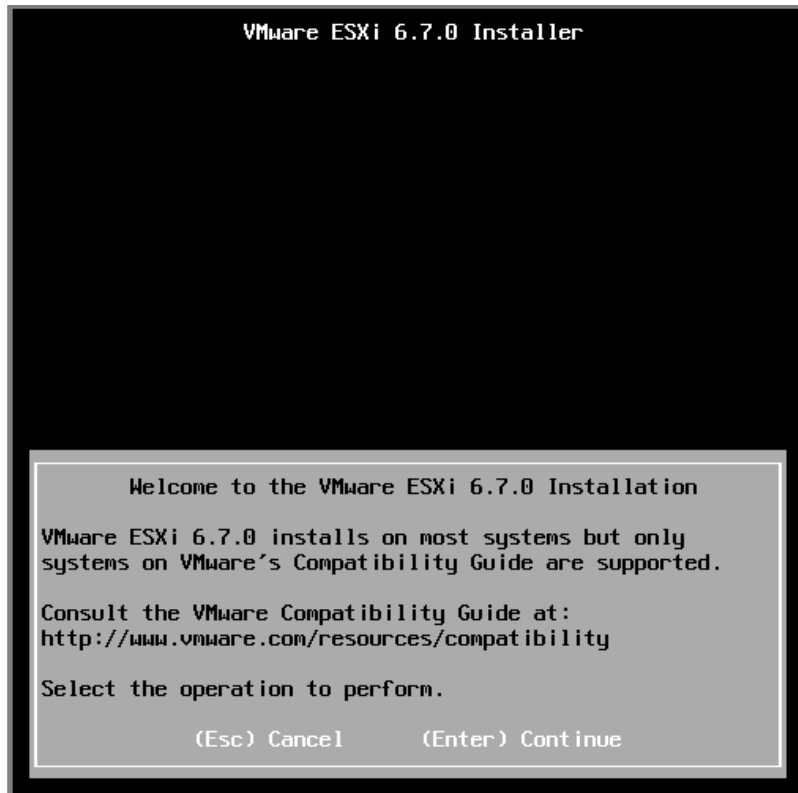


Figure 29: Démarrage d'installation d'ESXi.

Un contrat de licence s'affiche comme le montre la figure 30, on appuie sur la touche du clavier F11 pour accepter le contrat.

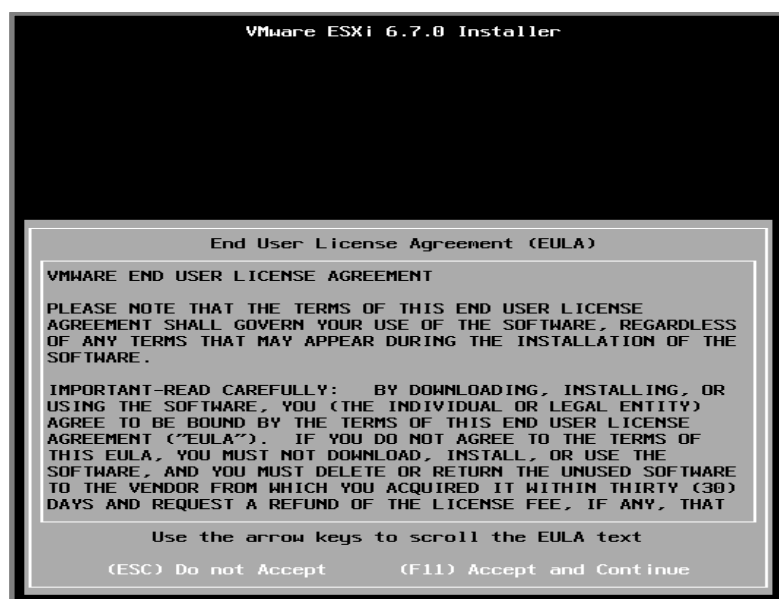


Figure 30: Contrat de licence utilisateur final.

Après la confirmation du contrat, on va se trouver devant l'interface illustré dans la figure 31 ou il montre le disque de stockage disponible ainsi que sa capacité.



Figure 31: Sélection de disque.

On sélection la langue de clavier dont on va utiliser comme le montre la figure 32 affichée ci-dessous, et on presse la touche entrer pour continuer l'installation.



Figure 32: Sélection de la disposition de clavier.

L'étape suivante consiste au saisi de mot de passe comme indiquier dans la figure 33, où on va taper un mot de passe pour l'utilisateur root ESXi.



Figure 33: Saisir le mot de passe.

Passant à la configuration de l'installation comme la figure 34 l'illustre. On appuie sur F11 pour confirmer l'installation d'ESXi.



Figure 34: Confirmation d'installation d'ESXi.

Enfin, une fois l'installation terminée on appuie sur entrée pour redémarrer le serveur comme montré dans la figure 35.

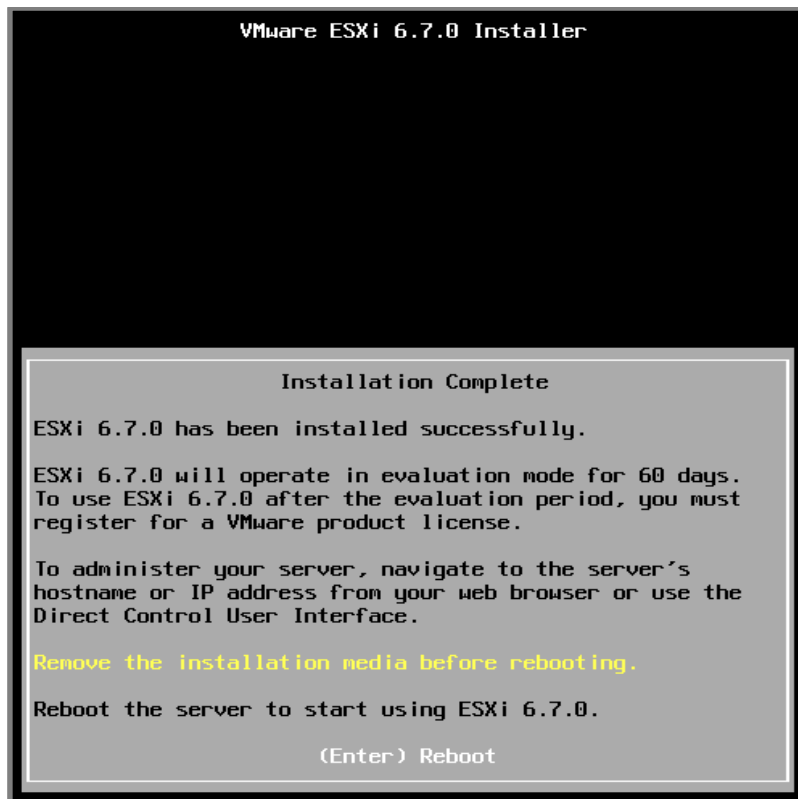


Figure 35: Fin d'installation d'ESXi.

5.3 Installation du VCenter

VMware VCenter est la gestion centralisée composant de l'environnement virtuel VMware vSphere. VCenter Server peut être utilisé pour administrer les entités du plan de contrôle de l'environnement virtuel, telles que les hôtes ESXi, les machines virtuelles, les réseaux virtuels, les banques de données, etc. à partir d'un emplacement centralisé. VCenter Server gère principalement de nombreuses fonctionnalités qui sont directement liées à l'administration de l'environnement virtuel. Il offre de nombreux avantages aux charges de travail des machines virtuelles soulignées, notamment l'optimisation de la disponibilité et de la possibilité de restauration, une administration simple et facile, l'analyse des performances et l'alerte pour avertir de tout problème croissant de l'environnement.

L'interface illustré dans la figure 36 montre le menu principal de VMware VCenter, qui offre plusieurs options ou on va choisir l'option d'installation.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

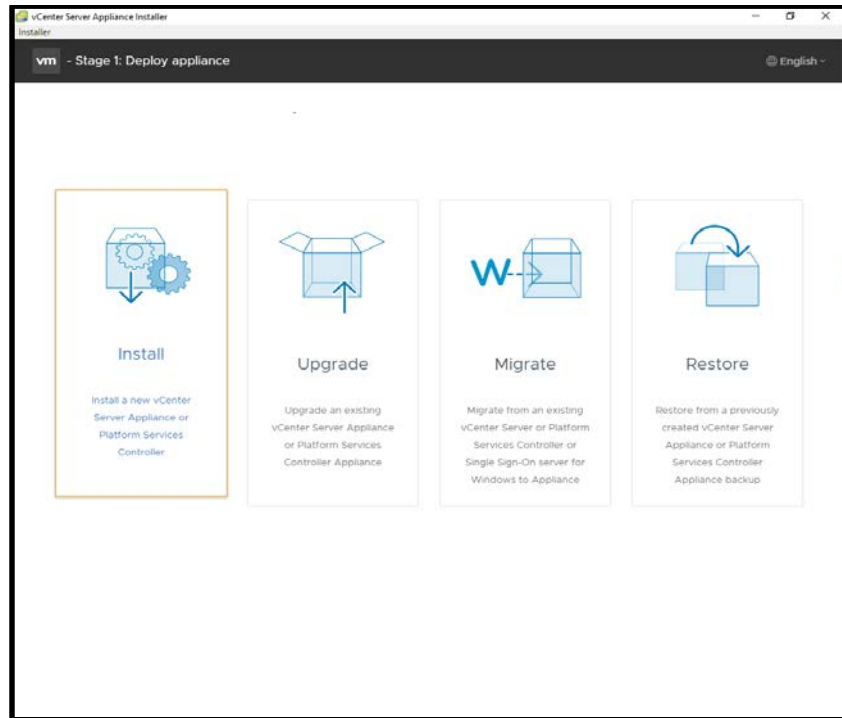


Figure 36: Menu principale de VCenter.

Comme illustré dans la figure 37 suivante, nous choisissons le serveur ESXI sur lequel sera déployé la VM VCenter, on va spécifier l'adresse, le port http le nom d'utilisateur et le mot de passe.

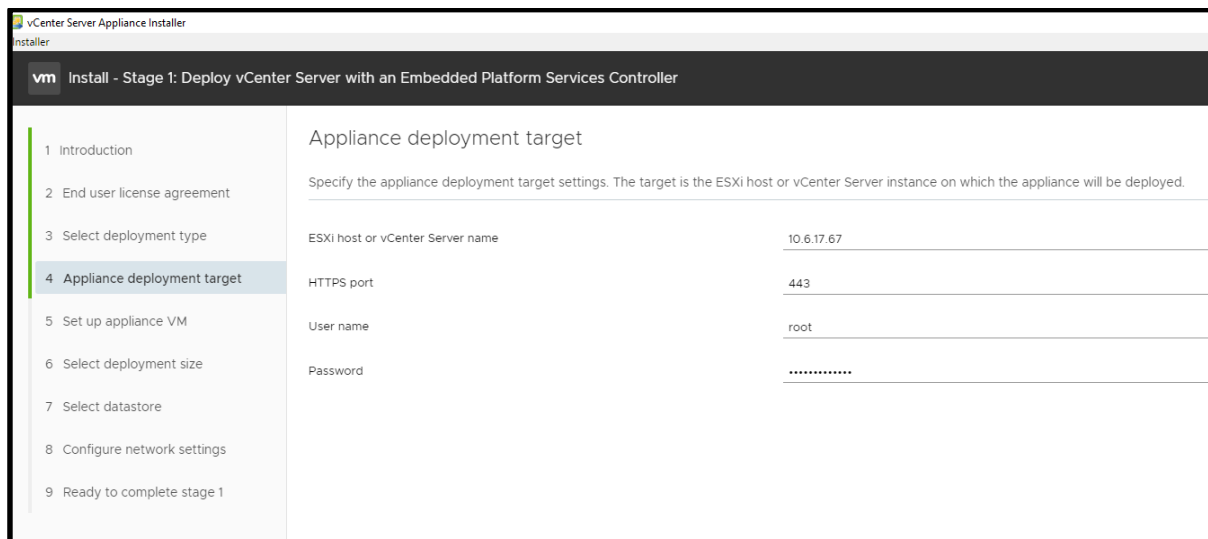


Figure 37: Choix de serveur ESXi.

La figure 38 ci-dessous illustre la sélection de la taille de notre VCenter (ici Tiny, qui peut héberger jusqu'à 100 machines virtuelles et 10 serveurs). Pour la taille de la machine administrateur VCenter on aura besoin de 2VCPUs, 10GB de RAM et 300GB de Stockage.

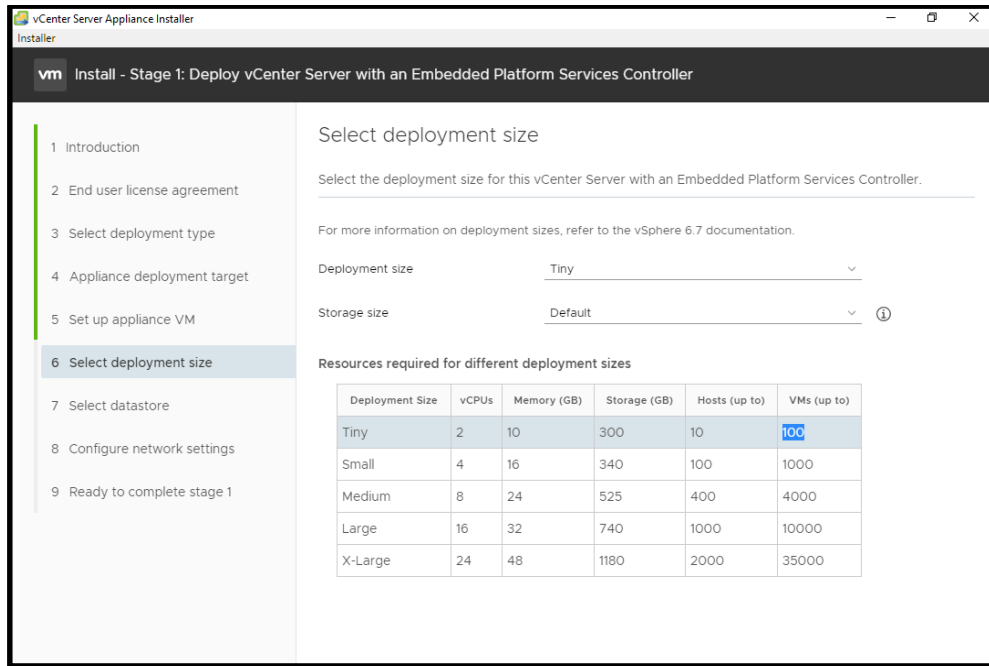


Figure 38: Sélection de la taille de VCenter.

Passant maintenant à la configuration d'adresses, et comme illustré dans la figure 39 ci-dessous, nous configurons l'adresse IP de VCenter, ainsi que ses ports d'accès http et https.

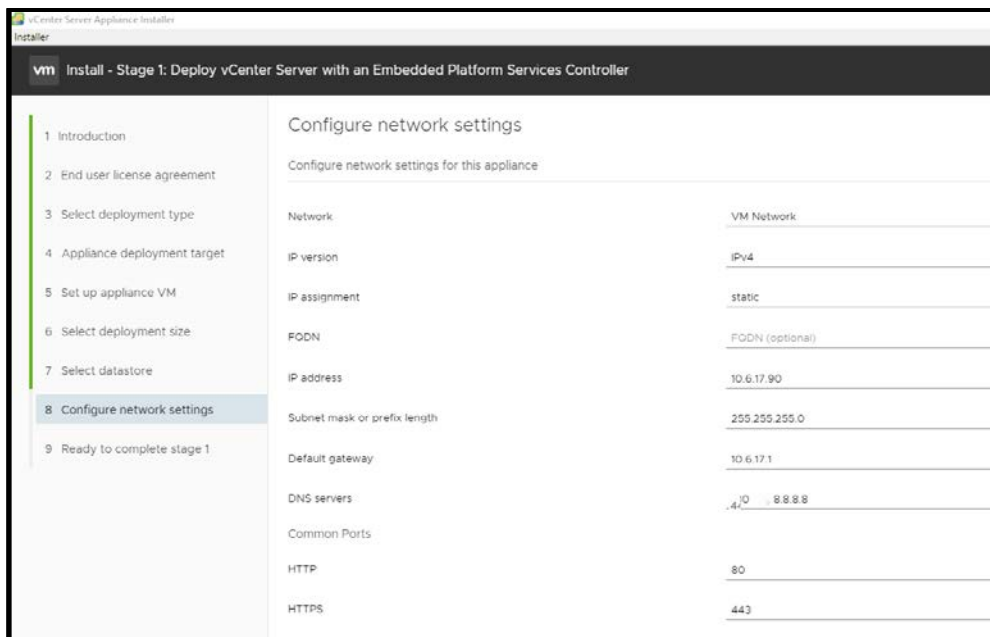


Figure 39: Configuration d'adresse.

Enfin, le visionnement des paramètres avant d'appliquer le déploiement est montré dans la figure 40, où on peut voir tous les caractéristiques de notre machine VCenter.

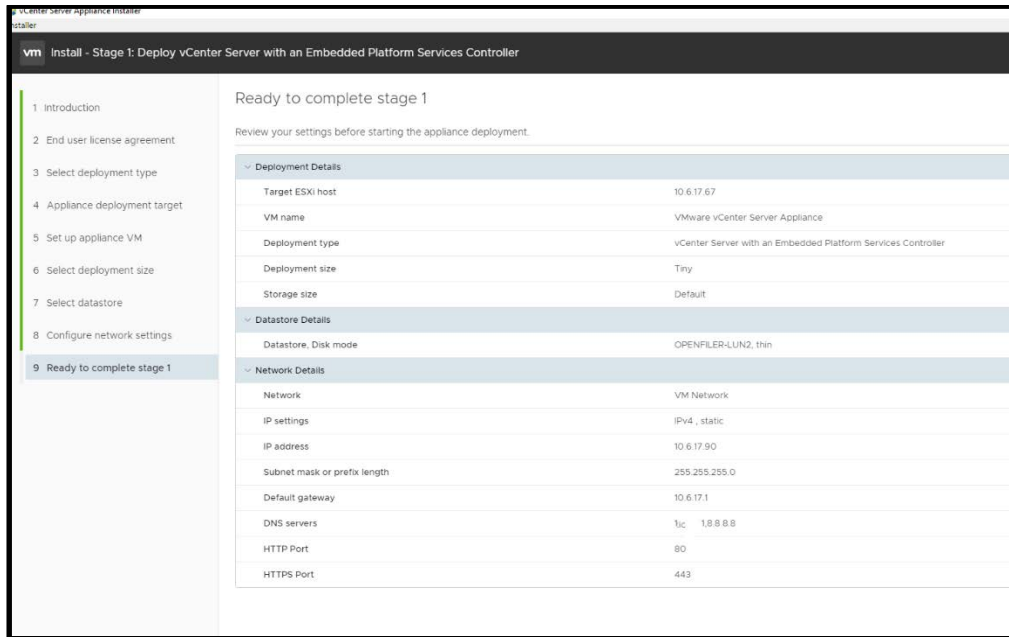


Figure 40: Paramètres finales.

5.4 Préparation des VMS dans le cluster

Notre cluster VMware nommé "OpenStack-Cluster" qui est montré dans la figure 41 ci-dessous, représentant l'ensemble de notre architecture avec les deux serveurs et les VMs qu'ils hébergent. (Fonctionnelles (en vert) et les backups qui sont éteints).

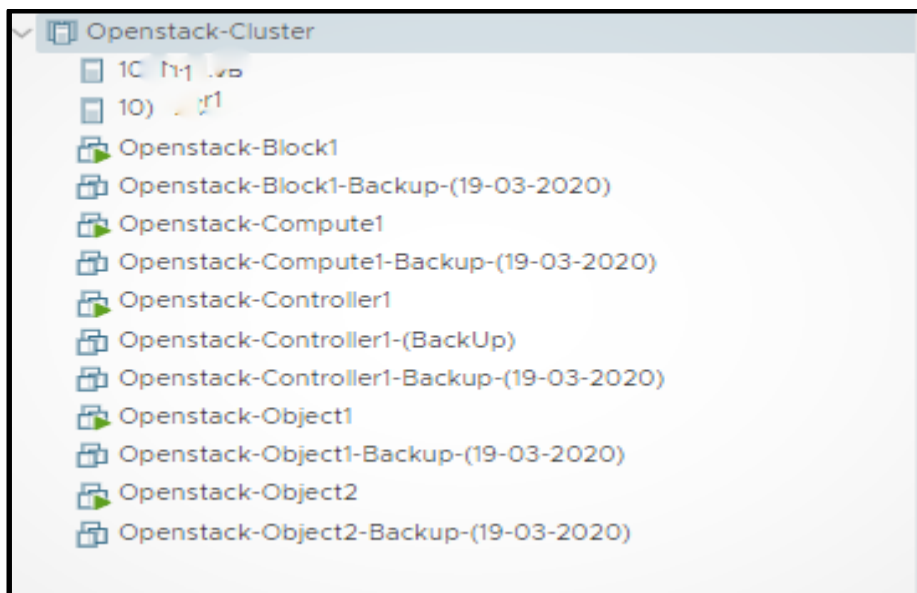


Figure 41: Vision sur le Cluster.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

On spécifie les caractéristiques des serveurs, l'un contenant actuellement les VMs backup d'OpenStack, et qui est présent pour le HA et DRS de la plateforme en cas de besoin, ou d'incidents sur l'autre serveur comme le montre la figure 42, et l'autre avec les machines allumées de la plateforme OpenStack (Montré par la "memory used" de 102.19GB) montré dans la figure 43 suivante.

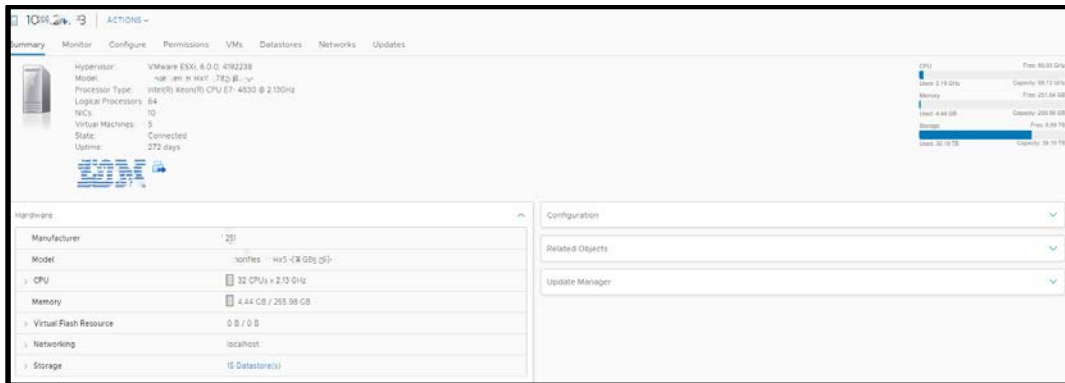


Figure 42: Le premier serveur.

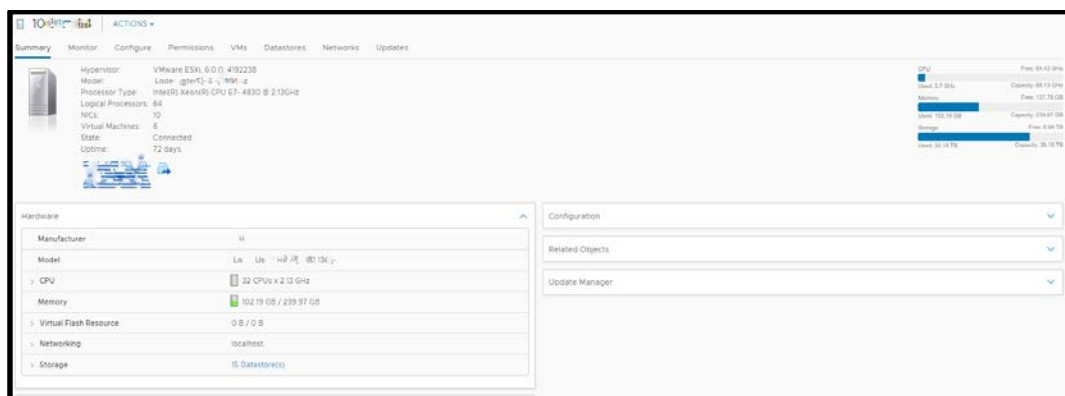


Figure 43: Le deuxième serveur.

Configuration réseau coté VMware des deux serveurs comme montrée dans les deux figures 44 et 45 : Nous avons alors créé un Switch virtuel sur chaque serveur, coté VMware connecté à deux cartes réseaux physiques (sur chaque serveur) de sortie (Physical Adapter "vmnic" sur l'image), nous plaçons nos machines dans un groupe sur ces switch avec un tag VLAN correspondant à l'architecture réseau existant ("VLAN ID: ").

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

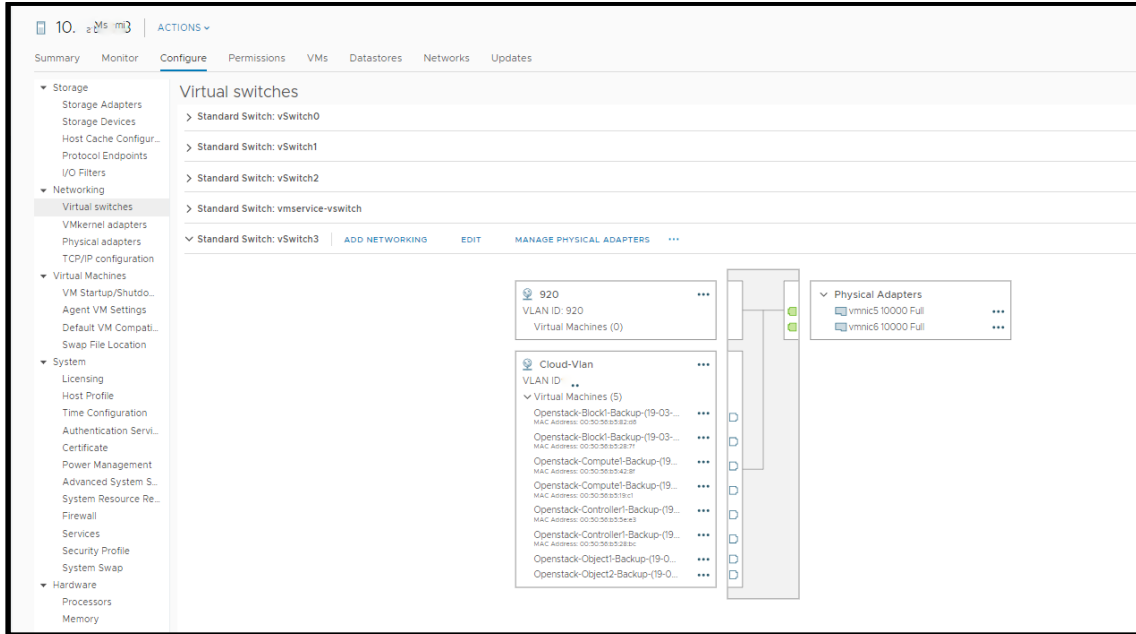


Figure 44: Configuration du réseau du premier serveur.

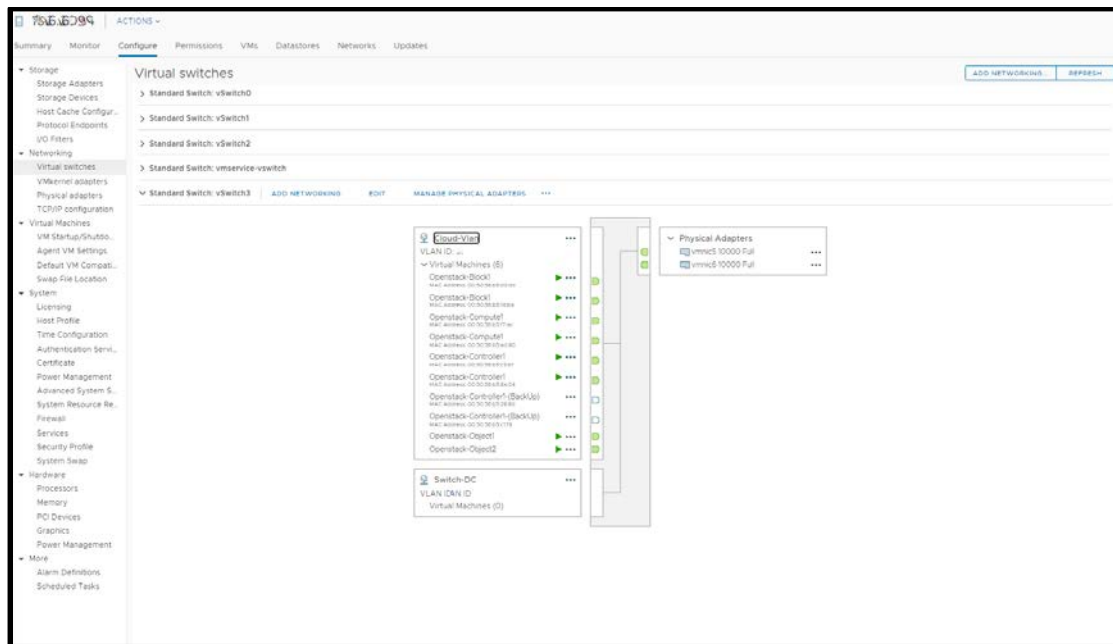


Figure 45: Configuration du réseau du deuxième serveur.

L'ensemble de nos machines virtuelles VMWare, avec leur caractéristiques (CPU, Ram "memory", Disque Dur, et cartes réseau "network adapters" connectées au groupe "Cloud-Vlan" montré dans les figures 46 - 50).

Openstack-Controller1 | [ACTIONS](#)

Summary | Monitor | Configure | Permissions | Datastores | Networks | Updates

Powered On

Guest OS: Linux 3.10.0-1062.12.1.el7.x86_64 CentOS Linux release 7.7.1908 (Core)
Compatibility: ESXi 6.0 and later (VM version 11)
VMware Tools: Running, version:2147483647 (Guest Managed)
[More info](#)

DNS Name: controller1
IP Addresses: 192.168.122.1
[View all 6 IP addresses](#)

Host: [3 ds](#)

[Launch Web Console](#)
[Launch Remote Console](#)

VM Hardware

> CPU	6 CPU(s)
> Memory	30 GB, 0.9 GB memory active
> Hard disk 1	400 GB
∨ Network adapter 1	
Adapter Type	VMXNET 3
MAC Address	00:50:56:5a:5b:5c f
DirectPath I/O	Inactive
Network	Cloud-Vlan (connected)
∨ Network adapter 2	
Adapter Type	VMXNET 3
MAC Address	00:50:56:10:5a:5b a
DirectPath I/O	Inactive
Network	Cloud-Vlan (connected)
CD/DVD drive 1	Disconnected v
> Video card	128 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
> Other	Additional Hardware
Compatibility	ESXi 6.0 and later (VM version 11)

[Edit Settings...](#)

Figure 46: Caractéristiques VM OpenStack Controlleur1

The screenshot shows the OpenStack Compute1 VM details page. The VM is named 'compute1' and is powered on. The hardware specifications are as follows:

VM Hardware	
CPU	24 CPU(s)
Memory	60 GB, 1.2 GB memory active
Hard disk 1	400 GB
Network adapter 1	
Adapter Type	VMXNET 3
MAC Address	00:50:56:1b:17:73
DirectPath I/O	Inactive
Network	Cloud-Vlan (connected)
Network adapter 2	
Adapter Type	VMXNET 3
MAC Address	00:50:56:1b:17:73
DirectPath I/O	Inactive
Network	Cloud-Vlan (connected)
CD/DVD drive 1	Disconnected
Video card	128 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
Other	Additional Hardware
Compatibility	ESXi 6.0 and later (VM version 11)

Figure 47: Caractéristiques VM OpenStack Compute1.

The screenshot shows the OpenStack Block1 VM details page. The VM is named 'block1' and is powered on. The hardware specifications are as follows:

VM Hardware	
CPU	4 CPU(s)
Memory	16 GB, 0.32 GB memory active
Hard disk 1	400 GB
Hard disk 2	800 GB
Network adapter 1	
Adapter Type	VMXNET 3
MAC Address	00:50:56:1b:17:73
DirectPath I/O	Inactive
Network	Cloud-Vlan (connected)
Network adapter 2	
Adapter Type	VMXNET 3
MAC Address	00:50:56:1b:17:73
DirectPath I/O	Inactive
Network	Cloud-Vlan (connected)
CD/DVD drive 1	Disconnected
Video card	128 MB
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
Other	Additional Hardware
Compatibility	ESXi 6.0 and later (VM version 11)

Figure 48: Caractéristiques VM OpenStack block1.

The screenshot shows the configuration page for VM 'Openstack-Object1'. The interface includes a navigation bar with tabs for Summary, Monitor, Configure, Permissions, Datastores, Networks, and Updates. The main content area displays the following information:

- Summary:** Guest OS: Linux 3.10.0-1062.12.1.el7.x86_64 CentOS Linux release 7.7.1908 (Core); Compatibility: ESXi 6.0 and later (VM version 11); VMware Tools: Running, version:2147483647 (Guest Managed); DNS Name: object1; IP Addresses: 192.168.122.1; Host: lre Tc11.
- VM Hardware:**
 - CPU: 4 CPU(s)
 - Memory: 8 GB, 0.4 GB memory active
 - Hard disk 1: 50 GB
 - Total hard disks: 3 hard disks
 - Network adapter 1:
 - Adapter Type: VMXNET 3
 - MAC Address: 00:5C:9E:00:00:00 mcbd l66
 - DirectPath I/O: Inactive
 - Network: Cloud-Vlan (connected)
 - CD/DVD drive 1: Disconnected
 - Video card: 128 MB
 - VMCI device: Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
 - Other: Additional Hardware
 - Compatibility: ESXi 6.0 and later (VM version 11)

Figure 49: Caractéristiques VM OpenStack Object1.

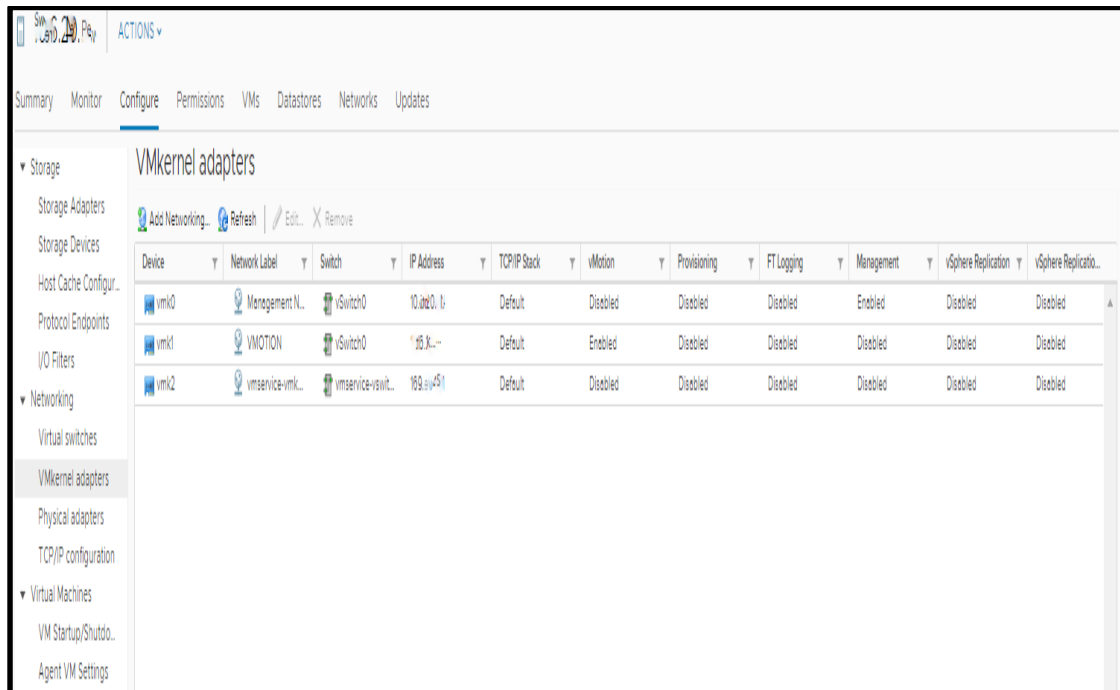
The screenshot shows the configuration page for VM 'Openstack-Object2'. The interface includes a navigation bar with tabs for Summary, Monitor, Configure, Permissions, Datastores, Networks, and Updates. The main content area displays the following information:

- Summary:** Guest OS: Linux 3.10.0-1062.12.1.el7.x86_64 CentOS Linux release 7.7.1908 (Core); Compatibility: ESXi 6.0 and later (VM version 11); VMware Tools: Running, version:2147483647 (Guest Managed); DNS Name: object2; IP Addresses: 192.168.122.1; Host: 10 eiea dt1.
- VM Hardware:**
 - CPU: 4 CPU(s)
 - Memory: 8 GB, 0.48 GB memory active
 - Hard disk 1: 50 GB
 - Total hard disks: 3 hard disks
 - Network adapter 1:
 - Adapter Type: VMXNET 3
 - MAC Address: 00:50:56:9A:00:00
 - DirectPath I/O: Inactive
 - Network: Cloud-Vlan (connected)
 - CD/DVD drive 1: Disconnected
 - Video card: 128 MB
 - VMCI device: Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
 - Other: Additional Hardware
 - Compatibility: ESXi 6.0 and later (VM version 11)

Figure 50: Caractéristiques VM OpenStack Object2.

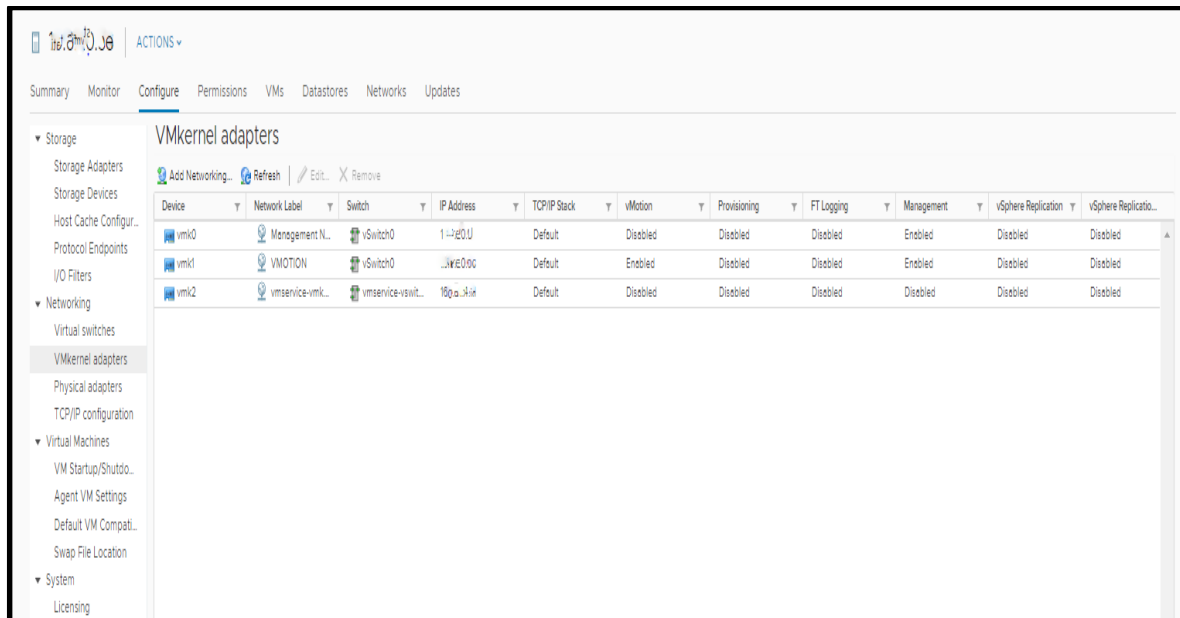
CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Les ports configurés pour les services de VMware sur chaque serveur, permettant une gestion centralisée des serveurs (vmk0), ainsi que les déplacements des machines entre chaque serveur (VMotion, vmk1). Les deux figures suivantes 51 et 52 présentent la configuration des ports sur le premier et le deuxième serveur.



Device	Network Label	Switch	IP Address	TCP/IP Stack	vMotion	Provisioning	FT Logging	Management	vSphere Replication	vSphere Replicatio...
vmk0	Management N...	vSwitch0	10.102.0.1	Default	Disabled	Disabled	Disabled	Enabled	Disabled	Disabled
vmk1	VMOTION	vSwitch0	10.102.0.2	Default	Enabled	Disabled	Disabled	Disabled	Disabled	Disabled
vmk2	vmervice-vmk...	vmervice-vswit...	10.102.0.3	Default	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled

Figure 51: Configuration des ports pour le premier serveur.



Device	Network Label	Switch	IP Address	TCP/IP Stack	vMotion	Provisioning	FT Logging	Management	vSphere Replication	vSphere Replicatio...
vmk0	Management N...	vSwitch0	10.102.0.1	Default	Disabled	Disabled	Disabled	Enabled	Disabled	Disabled
vmk1	VMOTION	vSwitch0	10.102.0.2	Default	Enabled	Disabled	Disabled	Enabled	Disabled	Disabled
vmk2	vmervice-vmk...	vmervice-vswit...	10.102.0.3	Default	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled

Figure 52: Configuration des ports pour le deuxième serveur.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

La figure 53 illustré ci-dessous montre la configuration DRS de notre cluster VMware, configuré pour le déplacement des machines entre nos deux serveurs en cas de surcharge sur l'un. Nous avons choisi "fully automated" pour permettre le déplacement automatique de nos VMs en cas de besoin sans une intervention manuelle nécessaire.

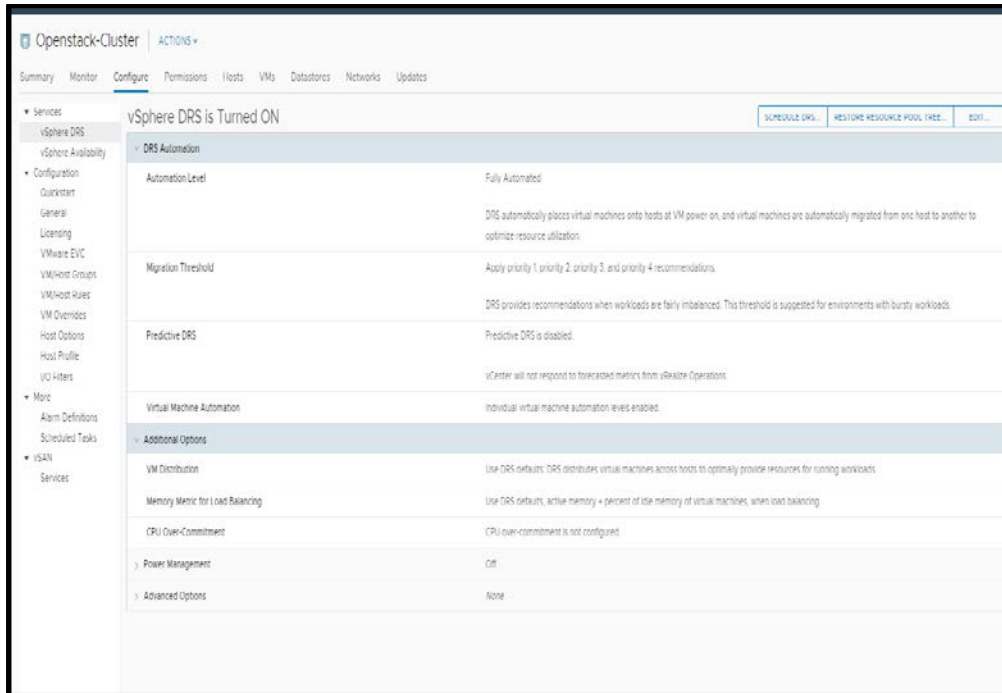


Figure 53: Configuration DRS du Cluster.

La figure 54 suivante illustre la configuration HA (High Availability) afin de redémarrer notre environnement OpenStack en cas d'incident ou perte de réponse par l'un des serveurs, sur l'autre "Host Failure : restart VMs" et "Host Isolation : Power Off and restart VMs" avec une priorité élevée "Default VM restart priority : High" comme le montre la figure 55.

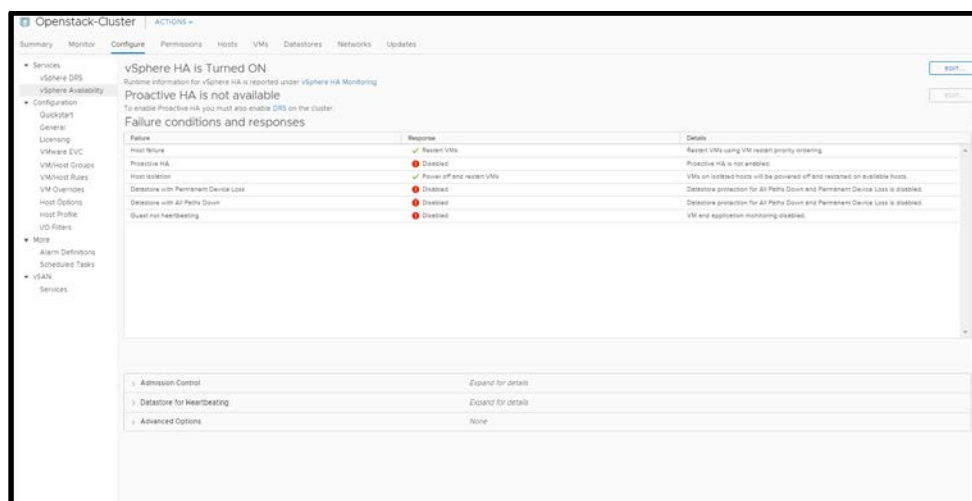


Figure 54: Configuration HA.

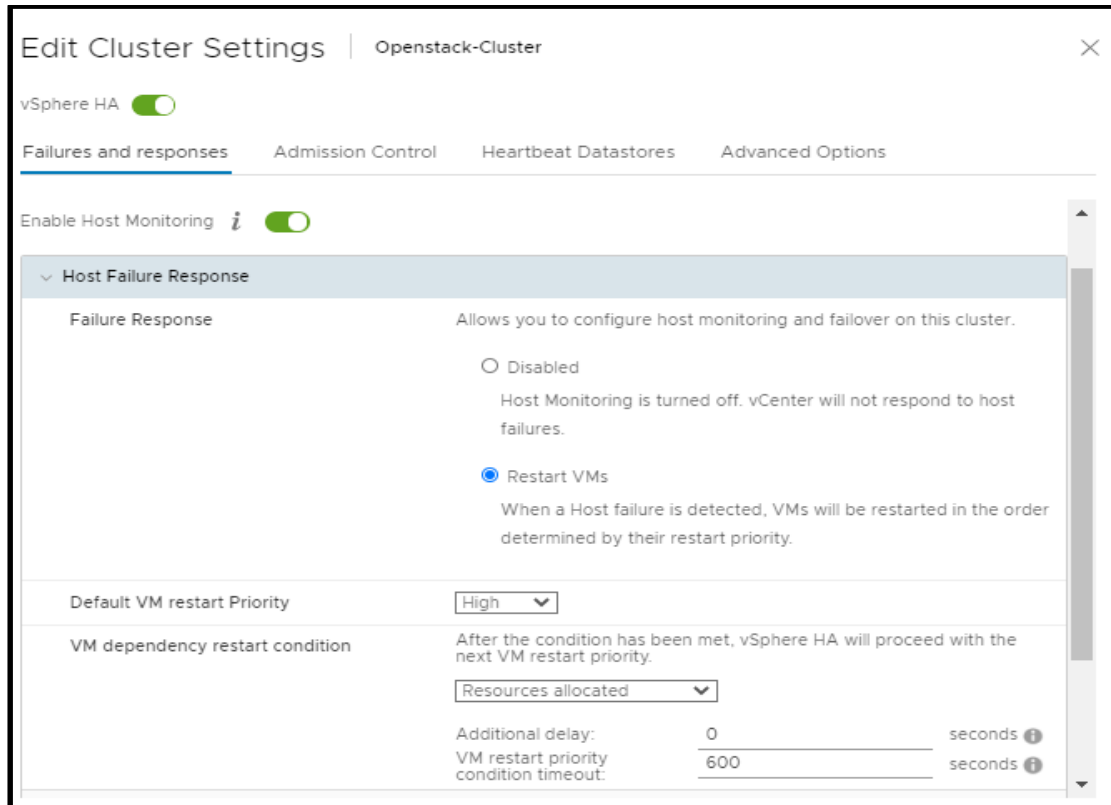


Figure 55: Éditer les paramètres du cluster.

5.5 Préparation des VMs avant les services OpenStack

Interfaces (cartes) réseaux des Vms créées coté VMware "ens160 et ens192". Les deux figures suivantes 56 et 57 montrent la création d'interface réseau sur le nœud compute1 et la création d'interface réseau pour le nœud controller1 respectivement.

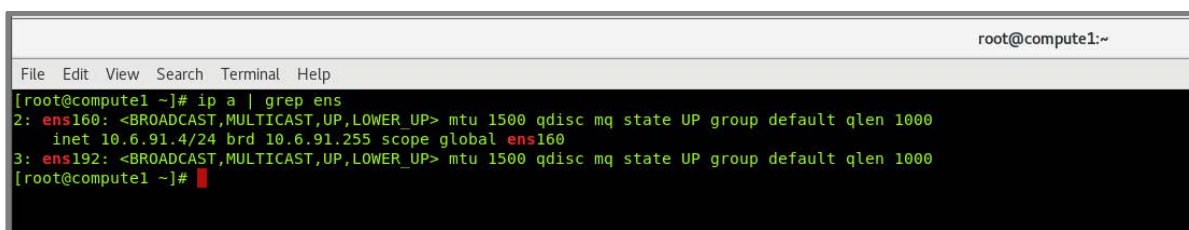


Figure 56: Création d'interface réseau sur le nœud compute1.

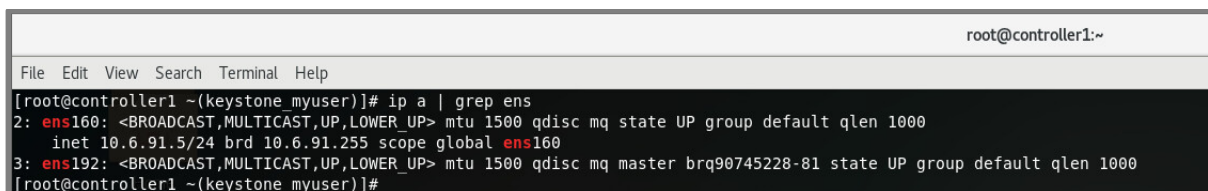
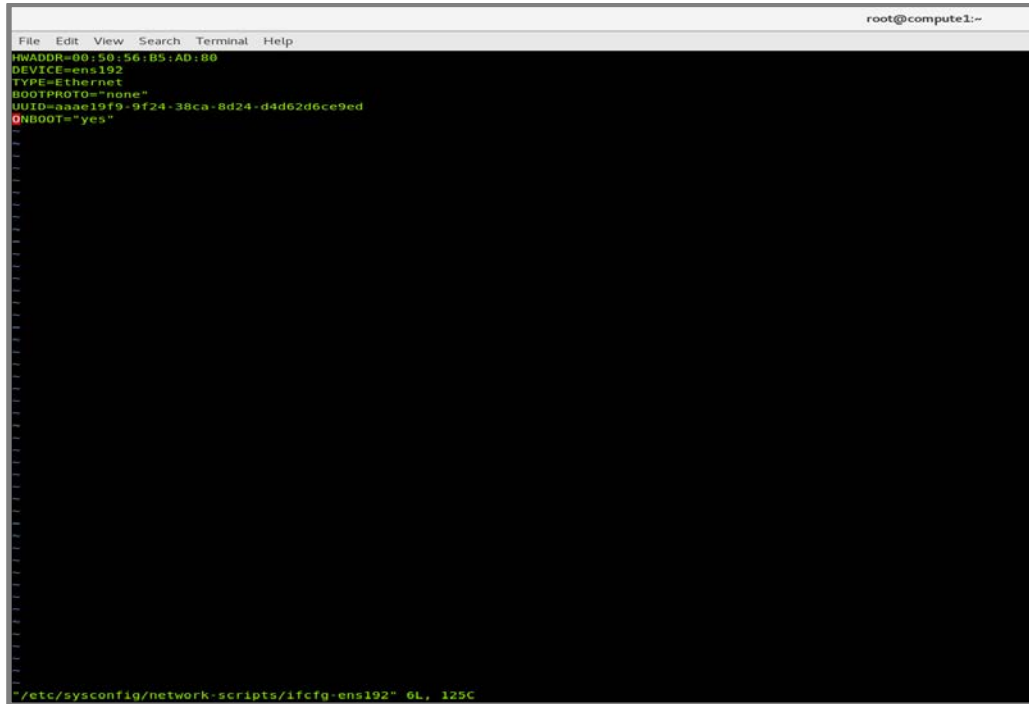


Figure 57: Création d'interface réseau pour le nœud controller1.

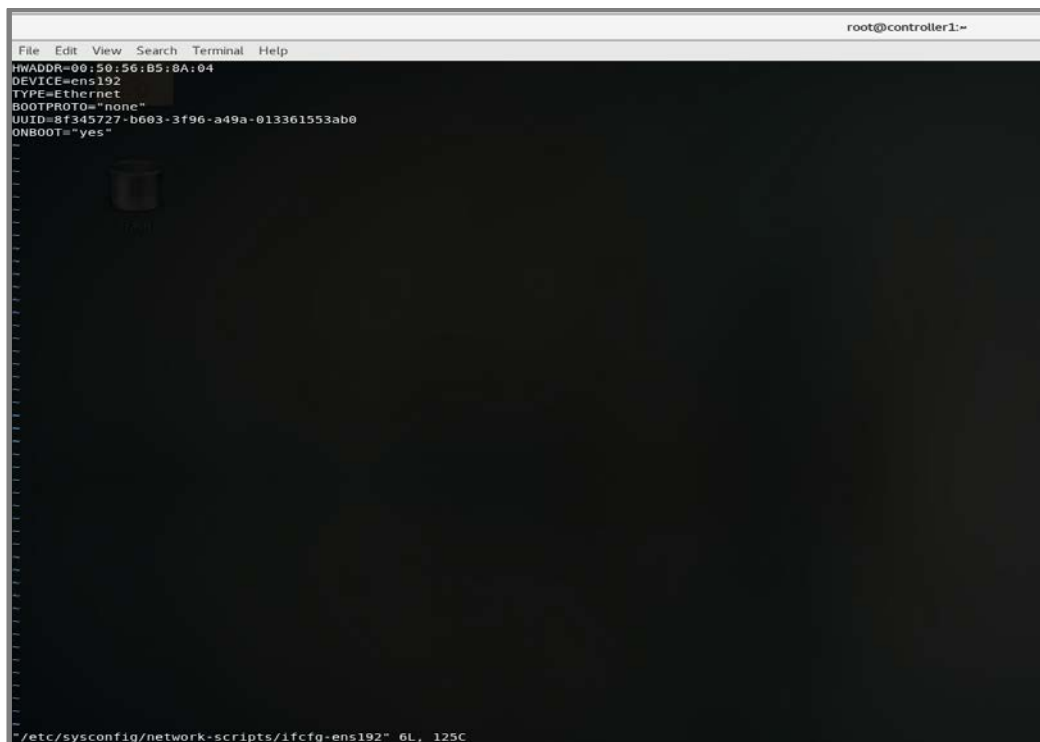
CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Configurations des cartes réseaux "ens160" sur les machines Compute1 et Contrôleur1 présentées sur les figures 58 et 59 respectivement. On configure Ens160 proprement avec une adresse IP, étant donné que ce sera la carte assurant la connectivité de ces Vms.



```
root@compute1:~#  
HWADDR=00:50:56:B5:AD:80  
DEVICE=ens192  
TYPE=Ethernet  
BOOTPROTO="none"  
UUID=aaae19f9-9f24-38ca-8d24-d4d62d6ce9ed  
ONBOOT="yes"  
  
"/etc/sysconfig/network-scripts/ifcfg-ens192" 6L, 125C
```

Figure 58: Configuration de la carte réseau "ens160" du nœud compute1.

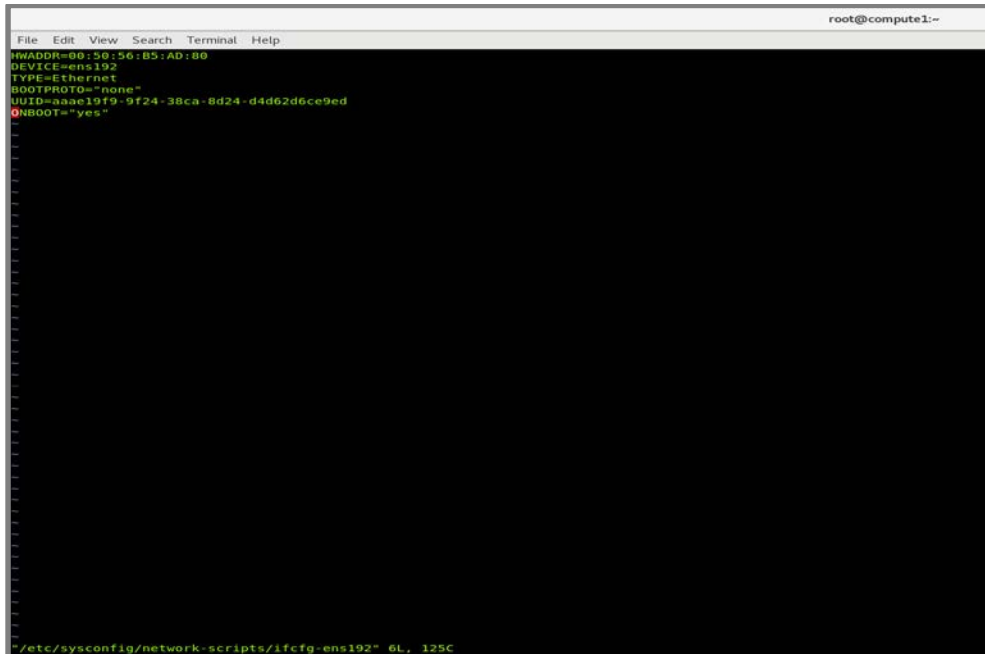


```
root@controller1:~#  
HWADDR=00:50:56:B5:8A:04  
DEVICE=ens192  
TYPE=Ethernet  
BOOTPROTO="none"  
UUID=8f345727-d603-3f96-a49a-013361553ab0  
ONBOOT="yes"  
  
"/etc/sysconfig/network-scripts/ifcfg-ens192" 6L, 125C
```

Figure 59: Configuration de la carte réseau "ens160" du nœud controller1.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

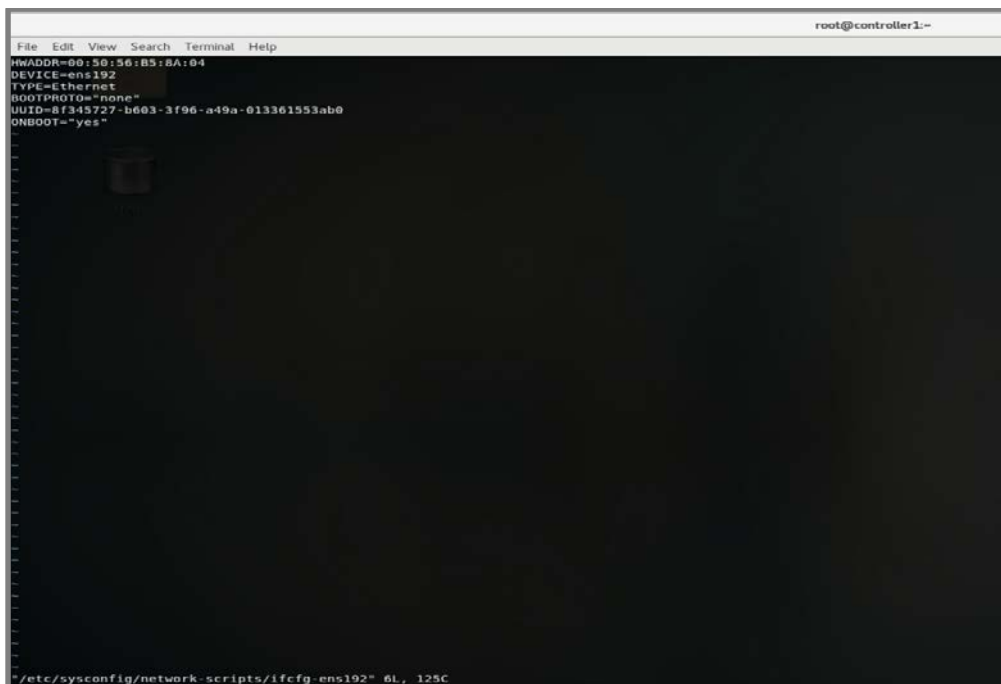
Configurations des deuxièmes cartes réseaux "ens192" Compute1 et Contrôleur1 présentées sur les figures 60 et 61 respectivement. Celles-ci n'auront pas d'adresse IP ou de protocole (ni DHCP ni IP statique). Ces cartes réseau seront utilisées comme Pont (Bridge) de sortie des instances créées sur OpenStack au réseau externe.



```
File Edit View Search Terminal Help
root@compute1:~
HWADDR=00:50:56:B5:AD:80
DEVICE=ens192
TYPE=Ethernet
BOOTPROTO="none"
UUID=aaac19f9-9f24-38ca-8d24-d4d62d6ce9ed
ONBOOT="yes"

~/etc/sysconfig/network-scripts/ifcfg-ens192" 6L, 125C
```

Figure 60: Configuration de la carte réseau "ens192" du nœud compute1.



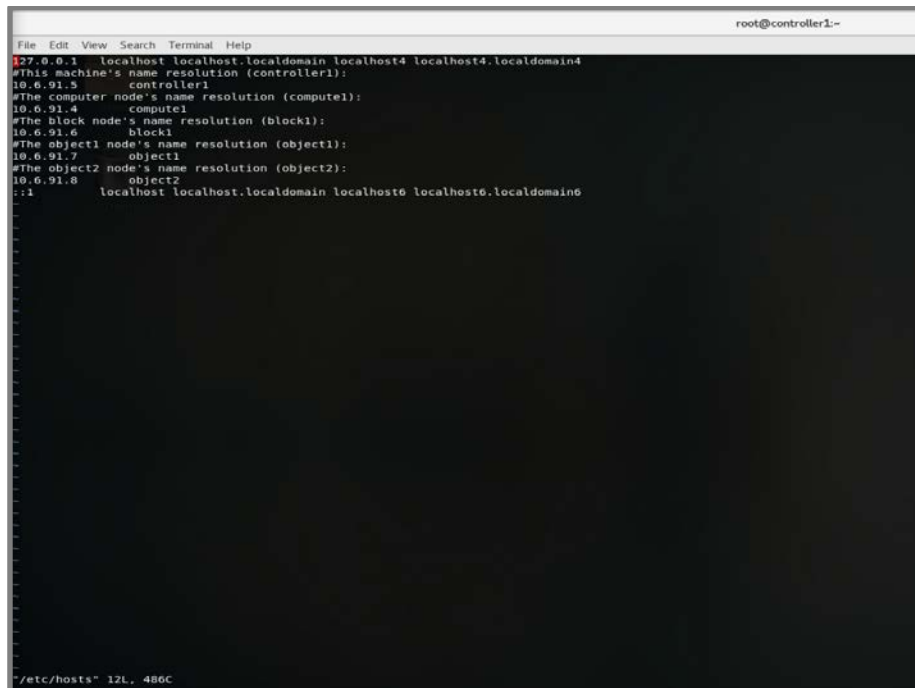
```
File Edit View Search Terminal Help
root@controller1:~
HWADDR=00:50:56:B5:8A:04
DEVICE=ens192
TYPE=Ethernet
BOOTPROTO="none"
UUID=8f345727-b603-3f96-a49a-013361553ab0
ONBOOT="yes"

~/etc/sysconfig/network-scripts/ifcfg-ens192" 6L, 125C
```

Figure 61: Configuration de la carte réseau "ens192" du nœud controller1.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

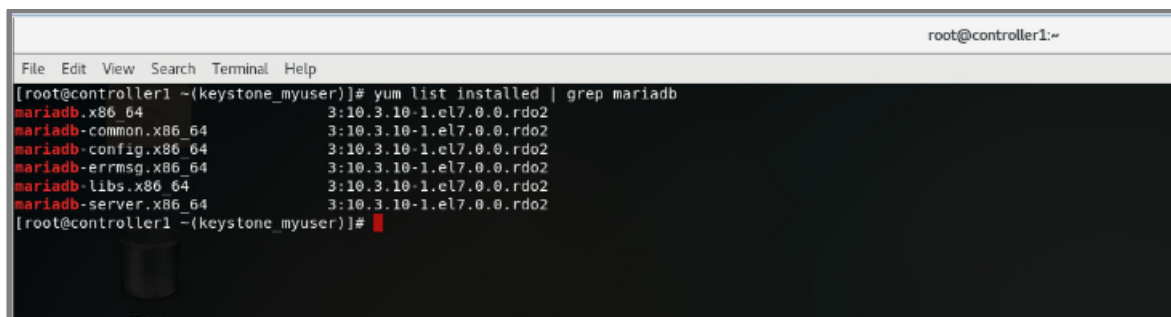
La figure 62 illustre la configuration de la résolution de noms sur chaque machine OpenStack, afin de permettre une interprétation des Hostnames des machines, à travers leurs adresses IP (controller1 = 10.6.91.5, compute1 = 10.6.91.4 ...etc. Ces machines pourront donc interpréter "Block1" comme 10.6.91.6 par exemple).



```
root@controller1:~  
File Edit View Search Terminal Help  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
#This machine's name resolution (controller1):  
10.6.91.5 controller1  
#The computer node's name resolution (compute1):  
10.6.91.4 compute1  
#The block node's name resolution (block1):  
10.6.91.6 block1  
#The object1 node's name resolution (object1):  
10.6.91.7 object1  
#The object2 node's name resolution (object2):  
10.6.91.8 object2  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
/etc/hosts" 12L, 486C
```

Figure 62: Configuration de la résolution de noms.

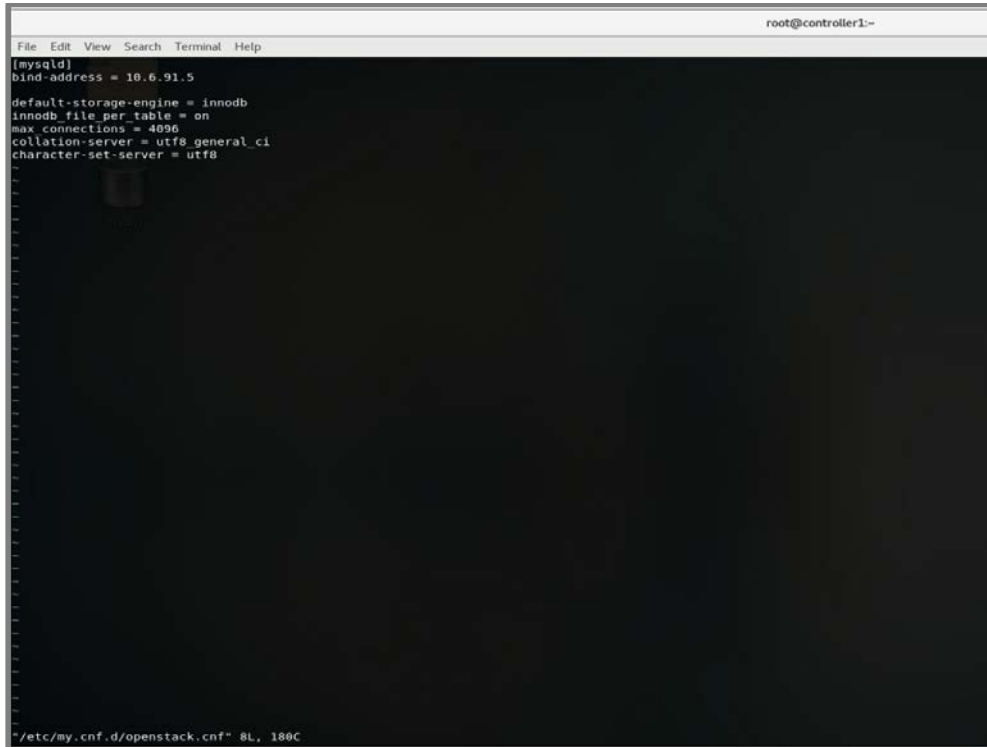
Maria dB (qui est une base de données MySQL) est installée et hébergée sur la VM Contrôleur. Elle sera utilisée par tous les services OpenStack pour la sauvegarde des données comme le montre la figure 63.



```
root@controller1:~  
File Edit View Search Terminal Help  
[root@controller1 ~(keystone_myuser)]# yum list installed | grep mariadb  
mariadb.x86_64 3:10.3.10-1.el7.0.0.rdo2  
mariadb-common.x86_64 3:10.3.10-1.el7.0.0.rdo2  
mariadb-config.x86_64 3:10.3.10-1.el7.0.0.rdo2  
mariadb-errmsg.x86_64 3:10.3.10-1.el7.0.0.rdo2  
mariadb-libs.x86_64 3:10.3.10-1.el7.0.0.rdo2  
mariadb-server.x86_64 3:10.3.10-1.el7.0.0.rdo2  
[root@controller1 ~(keystone_myuser)]#
```

Figure 63: Sauvegarde des données en utilisant Maria dB.

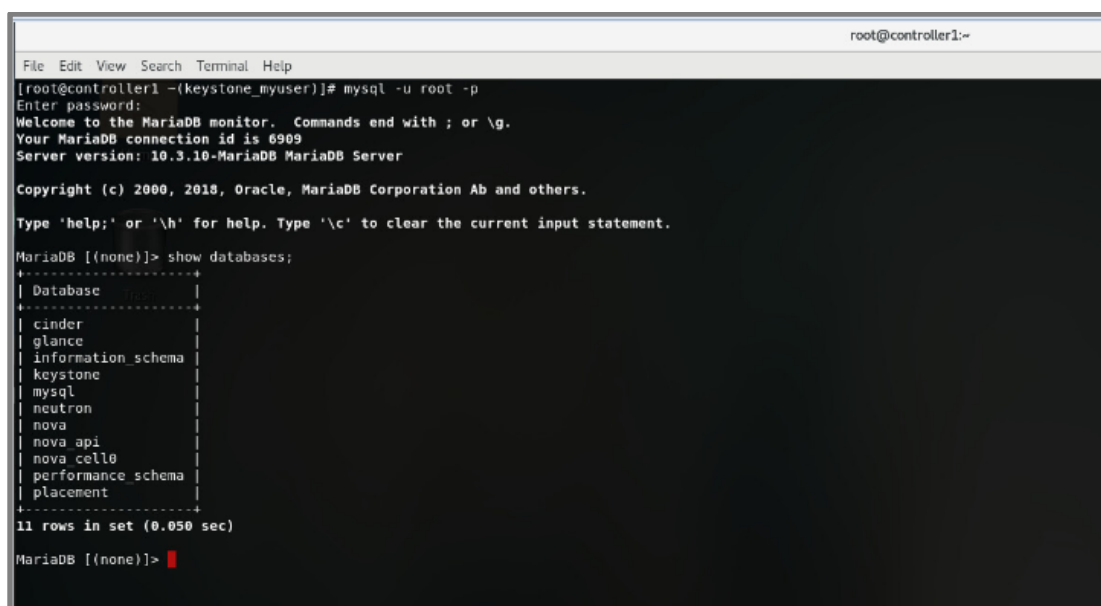
Configuration de MySQL présentée dans la figure 64 afin qu'il utilise la carte réseau ens160 sur le contrôleur comme interface de connexion pour les autres services (bind-address = 10.6.91.5).



```
File Edit View Search Terminal Help
root@controller1:~
[mysqld]
bind-address = 10.6.91.5
default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
...
/etc/my.cnf.d/openstack.cnf" 8L, 180C
```

Figure 64: Configuration de MySQL.

Les bases de données qu'on crée sur Maria dB pour chaque service sont illustrées dans la figure 65 suivante.



```
File Edit View Search Terminal Help
root@controller1:~
[root@controller1 ~(keystone_myuser)]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6909
Server version: 10.3.10-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

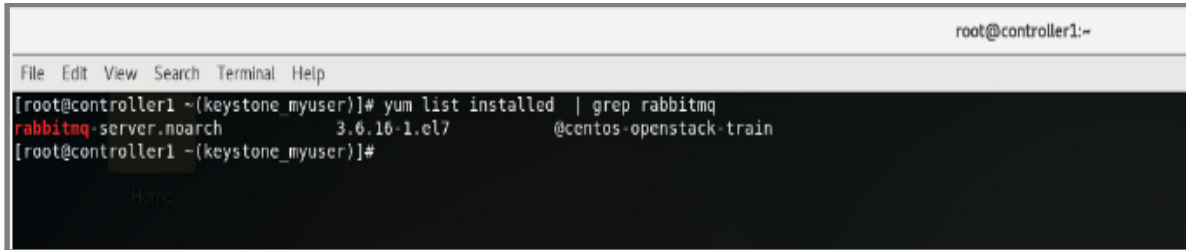
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| cinder   |
| glance  |
| information schema |
| keystone |
| mysql   |
| neutron |
| nova    |
| nova api|
| nova cell0 |
| performance schema |
| placement |
+-----+
11 rows in set (0.050 sec)

MariaDB [(none)]> █
```

Figure 65: Les bases de données de chaque service.

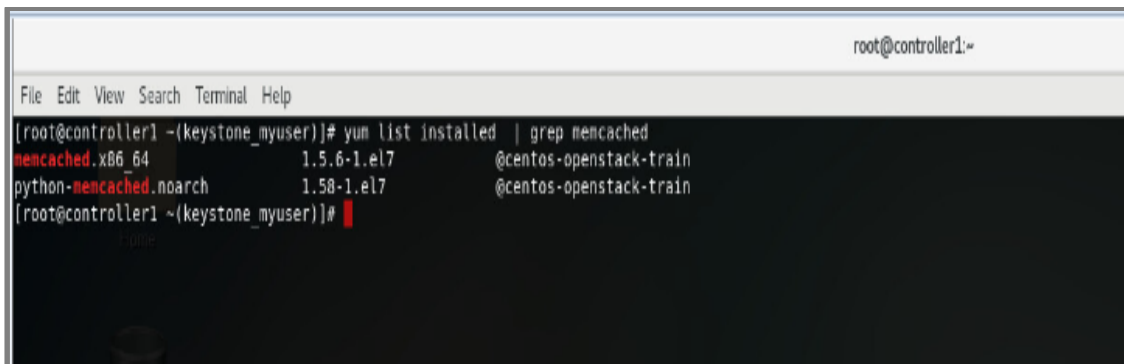
Outil de file d'attente (RabbitMQ-server) pour les requêtes des services d'OpenStack installé sur le Contrôleur. Afin d'organiser les requêtes des services en ordre. La figure 66 montre l'installation de RabbitMQ.



```
root@controller1:~  
File Edit View Search Terminal Help  
[root@controller1 ~-(keystone_myuser)]# yum list installed | grep rabbitmq  
rabbitmq-server.noarch      3.6.16-1.el7      @centos-openstack-train  
[root@controller1 ~-(keystone_myuser)]#
```

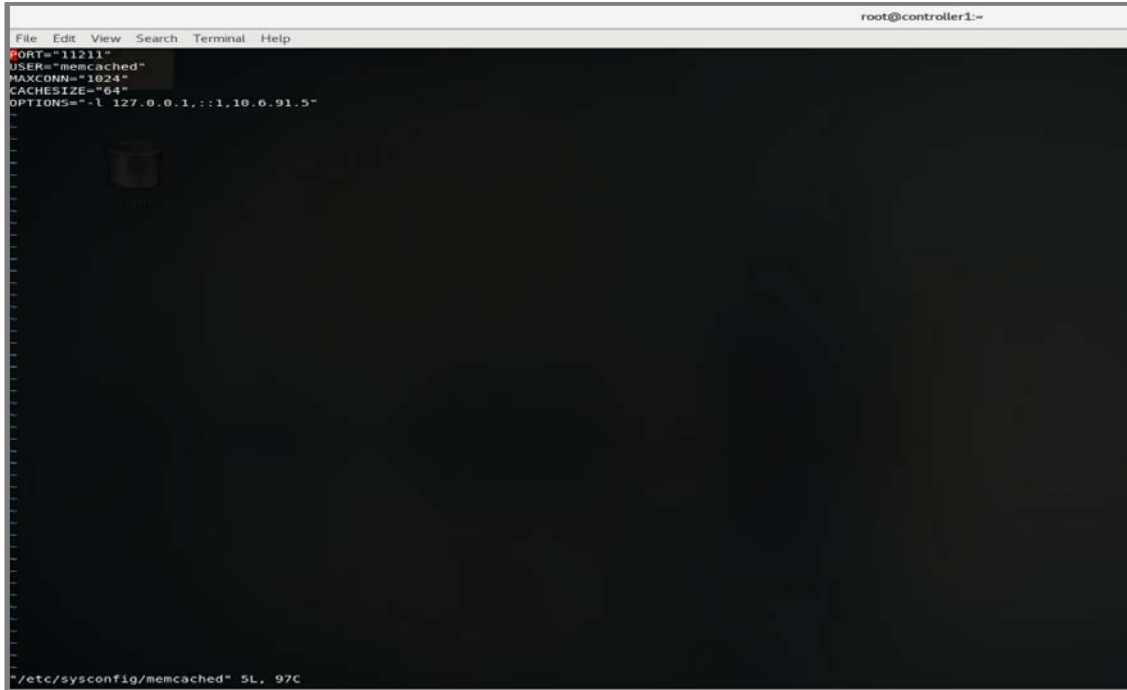
Figure 66: Installation de RabbitMQ.

Outil d'authentification (Memcached) des services d'OpenStack installé sur le Contrôleur comme illustré dans la figure 67. Cet outil assure la crédibilité de tous les services OpenStack, et vérifie leur authenticité avant de leur permettre d'accéder aux configurations. Chaque service sera configuré pour passer à travers Memcached sur le Contrôleur. Il est configuré sur le port 11211 du Contrôleur (10.6.91.5) comme montré dans la figure 68.



```
root@controller1:~  
File Edit View Search Terminal Help  
[root@controller1 ~-(keystone_myuser)]# yum list installed | grep memcached  
memcached.x86_64           1.5.6-1.el7      @centos-openstack-train  
python-memcached.noarch    1.58-1.el7       @centos-openstack-train  
[root@controller1 ~-(keystone_myuser)]#
```

Figure 67: Installation de l'outil Memcached.

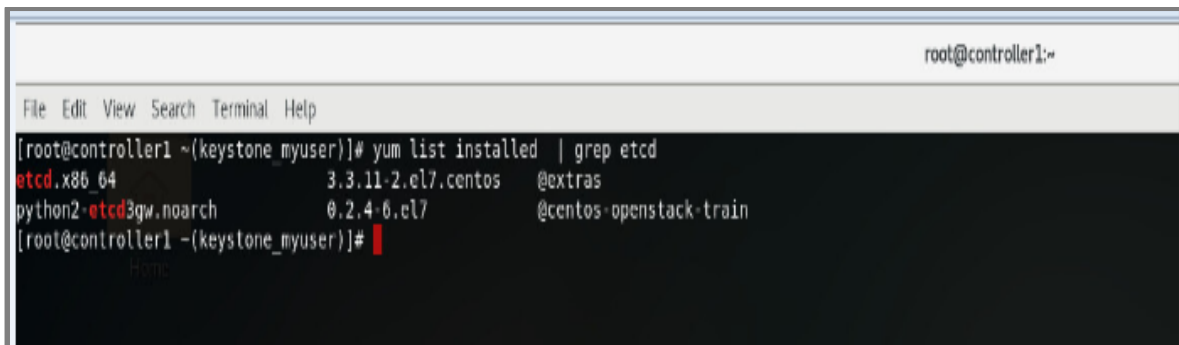


```
File Edit View Search Terminal Help
root@controller1:~#
PORT="11211"
USER="memcached"
MAXCONN="1024"
CACHE_SIZE="64"
OPTIONS="-l 127.0.0.1,::1,10.6.91.5"

~/etc/sysconfig/memcached 5L, 97C
```

Figure 68: Configuration du Memcached.

ETCD, est un outil installé sur le Contrôleur comme illustré dans la figure 69, qui aura pour rôle principal de rester en écoute des autres services OpenStack, afin que le contrôleur soit au courant de l'état des autres services.

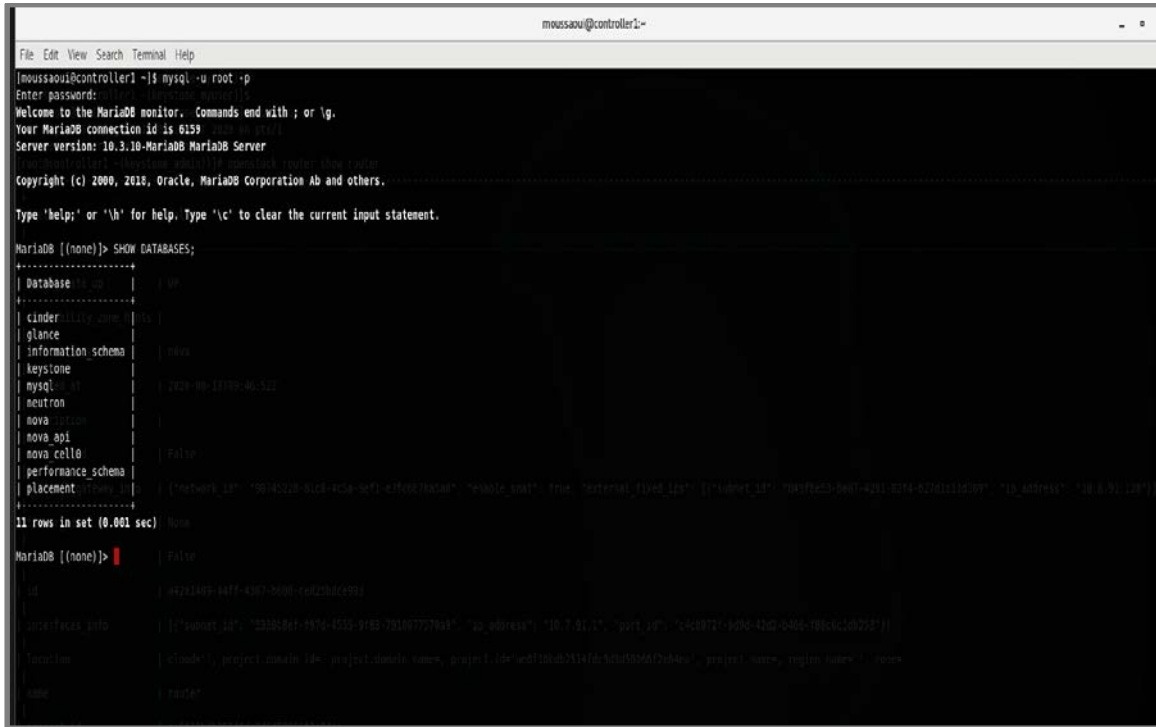


```
File Edit View Search Terminal Help
root@controller1:~#
[root@controller1 ~](keystone_myuser)# yum list installed | grep etcd
etcd.x86_64 3.3.11-2.el7.centos @extras
python2-etcd3gw.noarch 0.2.4-0.el7 @centos-openstack-train
[root@controller1 ~](keystone_myuser)#
```

Figure 69: Installation d'ETCD.

5.6 Déploiement des services

Lors du déploiement des services, nous configurons pour chacun, une base de données sur MySQL qui est déployée au niveau du contrôleur comme l'illustre la figure 70.



```
moussau@controller1-  
File Edit View Search Terminal Help  
[moussau@controller1 ~]$ mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 6193  
Server version: 10.3.10-MariaDB MariaDB Server  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> SHOW DATABASES;  
+-----+-----+  
| Database | Collation | |  
+-----+-----+  
| cinder | | |  
| glance | | |  
| information_schema | | |  
| keystone | | |  
| mysql | | |  
| neutron | | |  
| nova | | |  
| nova_api | | |  
| nova_cell0 | | |  
| performance_schema | | |  
| placement | | |  
+-----+-----+  
11 rows in set (0.001 sec)  
  
MariaDB [(none)]> USE mysql;  
mysql>
```

Figure 70: Configuration de base de données pour chaque service.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Nous attribuons les droits sur les bases de données des services à chaque utilisateur réciproque. Cela est présenté dans la figure 71.

```
| Grants for keystone@%
+-----+
| GRANT USAGE ON *.* TO 'keystone'@'%' IDENTIFIED BY PASSWORD '*6D25B1DB9A2E2AD2040F82DB425C2495E52E7698'
| GRANT ALL PRIVILEGES ON 'keystone'.* TO 'keystone'@'%'
+-----+
2 rows in set (0.000 sec)

MariaDB [keystone]> SHOW GRANTS FOR 'nova';
+-----+
| Grants for nova@%
+-----+
| GRANT USAGE ON *.* TO 'nova'@'%' IDENTIFIED BY PASSWORD '*6D25B1DB9A2E2AD2040F82DB425C2495E52E7698'
| GRANT ALL PRIVILEGES ON 'nova'.* TO 'nova'@'%'
| GRANT ALL PRIVILEGES ON 'nova_api'.* TO 'nova'@'%'
| GRANT ALL PRIVILEGES ON 'nova_cell0'.* TO 'nova'@'%'
+-----+
4 rows in set (0.000 sec)

MariaDB [keystone]> SHOW GRANTS FOR 'neutron';
+-----+
| Grants for neutron@%
+-----+
| GRANT USAGE ON *.* TO 'neutron'@'%' IDENTIFIED BY PASSWORD '*6D25B1DB9A2E2AD2040F82DB425C2495E52E7698'
| GRANT ALL PRIVILEGES ON 'neutron'.* TO 'neutron'@'%'
+-----+
2 rows in set (0.000 sec)

MariaDB [keystone]> SHOW GRANTS FOR 'cinder';
+-----+
| Grants for cinder@%
+-----+
| GRANT USAGE ON *.* TO 'cinder'@'%' IDENTIFIED BY PASSWORD '*6D25B1DB9A2E2AD2040F82DB425C2495E52E7698'
| GRANT ALL PRIVILEGES ON 'cinder'.* TO 'cinder'@'%'
+-----+
2 rows in set (0.000 sec)

MariaDB [keystone]> SHOW GRANTS FOR 'glance';
+-----+
| Grants for glance@%
+-----+
| GRANT USAGE ON *.* TO 'glance'@'%' IDENTIFIED BY PASSWORD '*6D25B1DB9A2E2AD2040F82DB425C2495E52E7698'
| GRANT ALL PRIVILEGES ON 'glance'.* TO 'glance'@'%'
+-----+
2 rows in set (0.000 sec)
```

Figure 71: Attribution des droits sur les bases de données.

La figure 72 suivante présente la configuration d'accès à la base de données où chaque service aura une configuration d'accès à sa base de données que nous devons définir "utilisateur: motdepasse@Controlleur/[Base de données du service]"

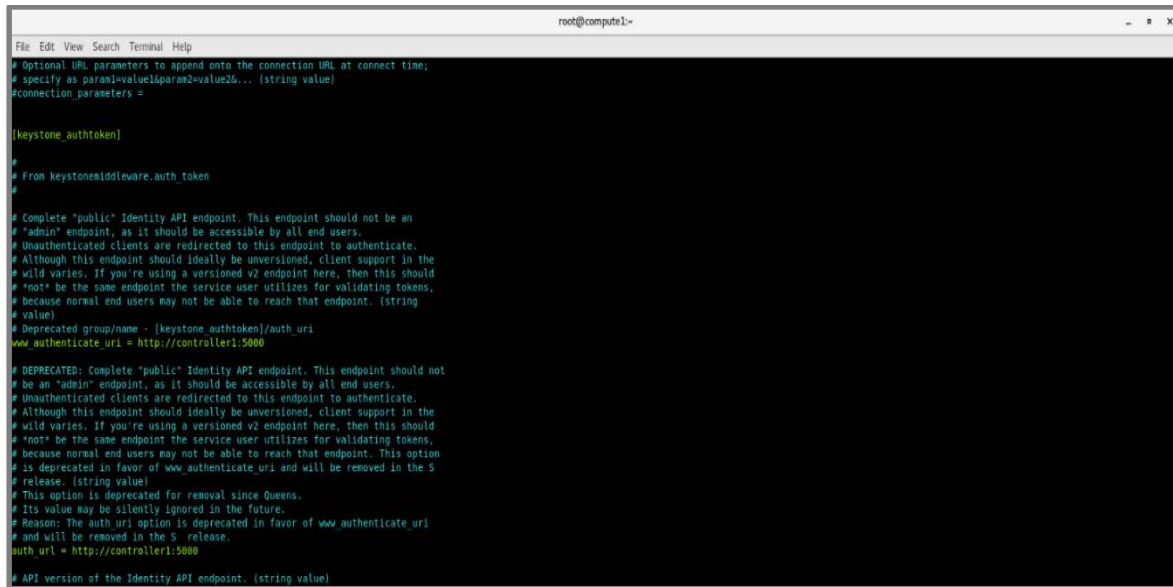
```
root@controller1:~# su - keystone
[database] Sun Aug  9 13:05:12 CET 2020 on pts/1
root@controller1:~# . admin-login
# root@controller1:~(keystone admin)]# openstack router show router
# From oslo.db
#
+-----+
| Field | Value
+-----+
# If True, SQLite uses synchronous mode. (boolean value)
#sqlite_synchronous = true
+-----+
# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy
+-----+
# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://glance:se@10.0.0.1@controller1/glance
distributed | False
+-----+
# The SQLAlchemy connection string to use to connect to the slave database.
# (string value)
connection = mysql+pymysql://glance:se@10.0.0.1@controller1/slave-glance
distributed | False
```

Figure 72: Configuration d'accès à la base de données.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Les services auront aussi notre configuration pour s'authentifier à Keystone sur le contrôleur à travers le port 5000.

Les figures 73 et 74 suivantes présentent respectivement la configuration du Compute1 et la configuration du Controller.



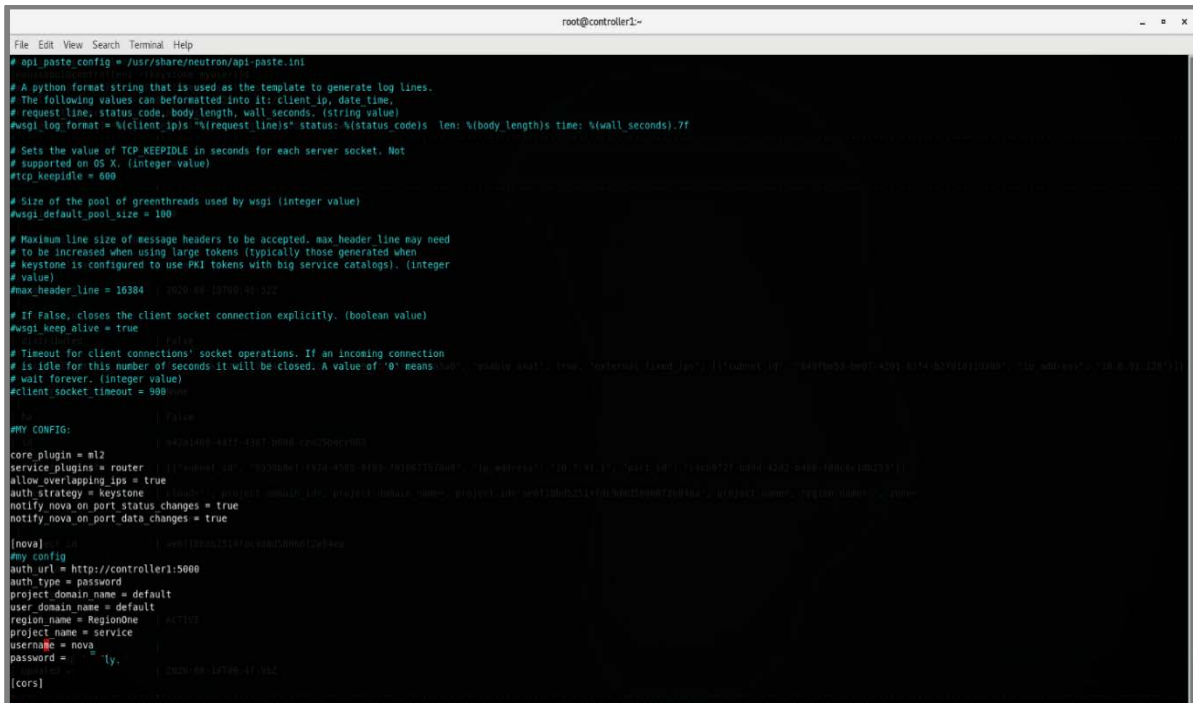
```
root@compute1~# cat /etc/nova/nova.conf | grep -A 20 -B 20 keystone
# Optional URL parameters to append onto the connection URL at connect time;
# specify as param1=value1param2=value2... (string value)
#connection_parameters =

[keystone_auth(token)]
#
# From keystone.middleware.auth token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users.
# Unauthenticated clients are redirected to this endpoint to authenticate.
# Although this endpoint should ideally be unversioned, client support in the
# wild varies. If you're using a versioned v2 endpoint here, then this should
# *not* be the same endpoint the service user utilizes for validating tokens,
# because normal end users may not be able to reach that endpoint. (string
# value)
# Deprecated group/name = [keystone_auth(token)/auth_uri
www_authenticate_uri = http://controller1:5000

# DEPRECATED: Complete "public" Identity API endpoint. This endpoint should not
# be an "admin" endpoint, as it should be accessible by all end users.
# Unauthenticated clients are redirected to this endpoint to authenticate.
# Although this endpoint should ideally be unversioned, client support in the
# wild varies. If you're using a versioned v2 endpoint here, then this should
# *not* be the same endpoint the service user utilizes for validating tokens,
# because normal end users may not be able to reach that endpoint. This option
# is deprecated in favor of www_authenticate_uri and will be removed in the 5
# release. (string value)
# This option is deprecated for removal since Queens.
# Its value may be silently ignored in the future.
# Reason: The auth_uri option is deprecated in favor of www_authenticate_uri
# and will be removed in the 5 - release.
auth_uri = http://controller1:5000

# API version of the Identity API endpoint. (string value)
```

Figure 73: Configuration du Compute1 pour s'authentifier à Keystone.



```
root@controller1~# cat /etc/nova/nova.conf | grep -A 20 -B 20 keystone
# api_paste_config = /usr/share/neutron/api.paste.ini
# A python format string that is used as the template to generate log lines.
# The following values can be formatted into its client_log_date_time,
# request_line, status_code, body length, wall seconds. (string value)
#wsgi_log_format = %(client_ip)s "%(request_line)s" status: %(status_code)s len: %(body_length)s time: %(wall_seconds)s.%f

# Sets the value of TCP_KEEPIDLE in seconds for each server socket. Not
# supported on OS X. (integer value)
#tcp_keepidle = 600

# Size of the pool of greenthreads used by wsgi. (integer value)
#wsgi_default_pool_size = 100

# Maximum line size of message headers to be accepted. max_header_line may need
# to be increased when using large tokens (typically those generated when
# keystone is configured to use PKI tokens with big service catalogs). (integer
# value)
#max_header_line = 16384

# If False, closes the client socket connection explicitly. (boolean value)
#wsgi_keep_alive = true

# Timeout for client connections' socket operations. If an incoming connection
# is idle for this number of seconds it will be closed. A value of '0' means
# wait forever. (integer value)
#client_socket_timeout = 900

[MY CONFIG]
core_plugin = ml2
service_plugins = router
allow_overlapping_ips = true
auth_strategy = keystone
notify_nova_on_port_status_changes = true
notify_nova_on_port_data_changes = true

[nova]
my_config
auth_url = http://controller1:5000
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = nova
password = ly

[cors]
```

Figure 74: Configuration du Controller1 pour s'authentifier à Keystone.

La figure 75 suivante présente la création de projet service où nous créons un projet "Service" qui est utilisé par les services pour l'administration et l'authentification à OpenStack. Nous définirons et créerons sur ce projet chaque service et son utilisateur.

```
[root@controller1 ~(keystone_admin)]# openstack project list
+-----+-----+
| ID:saoui@controller1 ~(keystone_admin) | Name |
+-----+-----+
| 14c438a6fb6f4d5ea48f623e4052572c | service |
| 1a4d9088818d49049836d5bc6f387e1c | admin |
| ae6f10bdb2514fdc9d8d58066f2e84ea | myproject |
+-----+-----+

[root@controller1 ~(keystone_admin)]# openstack project show service
+-----+-----+
| Field | Value |
+-----+-----+
| description | Service Project |
| domain_id | default |
| enabled | True |
| id | 14c438a6fb6f4d5ea48f623e4052572c |
| is_domain | False |
| name | service |
| options | {} |
| parent_id | default |
| tags | [] |
+-----+-----+

[root@controller1 ~(keystone_admin)]# openstack user list --project service
+-----+-----+
| ID: | Name: |
+-----+-----+
| 9cc8e5aa511d494ebd21060a250daf7c | swift |
| 74e4ff7f69814415a9b16d1f19fd95bc | nova |
| 012f47710f5b47108226d740f29b3f85 | neutron |
| dacd024847c64aa6a2597c9ab8649328 | placement |
| 0c086b2cac584123875ea6cbcc9ae073 | glance |
| df62daaa48a14447838f1bd48b70aafb | cinder |
+-----+-----+

[root@controller1 ~(keystone_admin)]#
location | cloud=, project.domain_id=, project.domain_name=, project.id='ae6f10b
name | router
project_id | ae6f10bdb2514fdc9d8d58066f2e84ea
```

Figure 75: Création du projet service.

Nous déclarons après cela le service et ses points d'accès au contrôleur qui sont définis par des ports, au niveau d'OpenStack. Chaque service a 3 points d'accès (Admin / Internal et External) pour nous permettre de sécuriser les communications des services en séparant la partie administration "Admin" de la partie utilisation interne, ou accès Externes. (Pour notre part nous n'avons pas encore d'utilisateurs externes, nous gardons alors le même accès pour les 3)

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Les figures 76 et 77 suivantes présentent respectivement la liste des services OpenStack et la liste des utilisateurs OpenStack.

```
[root@controller1 ~](keystone_admin)]# openstack service list
+-----+-----+-----+-----+
| ID | Name | Type |
+-----+-----+-----+
| 29c2c1ba83ba4d12930f1f97f6517c21 | keystone | identity |
| 3fef479815d44e569a6c1be0d0ac836e | cinderv2 | volumev2 |
| 51d425b2b7b34f649df06a2e81a988cf | cinderv3 | volumev3 |
| 5ac3163fdf5f4721928696bb7362739c | neutron | network |
| 71491a992767420ebb3b48e071d1ec9e | placement | placement |
| a59685a1a568477ca0e304ec17e47ccf | glance | image |
| ac1df36ab06a4ef2845405045a3195ab | nova | compute |
| dd53b4a47a824a3988a381df82875bc6 | swift | object-store |
+-----+-----+-----+-----+
[root@controller1 ~](keystone_admin)]# openstack endpoint list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Region | Service Name | Service Type | Enabled | Interface | URL |
+-----+-----+-----+-----+-----+-----+-----+
| 08052ba9bfe449eaadc9fcec392ceb8 | RegionOne | swift | object-store | True | admin | http://controller1:8080/v1 |
| 0c3107412f5141c3af12bf8ff97ea7 | RegionOne | cinderv3 | volumev3 | True | internal | http://controller1:8776/v3/%(project_id)s |
| 1d248e49148049938e5eb47d0eabf2a | RegionOne | keystone | identity | True | internal | http://controller1:5000/v3/ |
| 21c6790c374c463d8770ce0e80be1c6f | RegionOne | placement | placement | True | admin | http://controller1:8778 |
| 30b320a60f954403a44c2c7b51edd3db | RegionOne | keystone | identity | True | admin | http://controller1:5000/v3/ |
| 30a7f2703f24c0b341022f50b7e85 | RegionOne | cinderv2 | volumev2 | True | public | http://controller1:8776/v2/%(project_id)s |
| 46020683e5c44b31a1d4d76de67db429 | RegionOne | swift | object-store | True | public | http://controller1:8080/v1/AUTH_%(project_id)s |
| 4803cdf27de943959180882e42184478 | RegionOne | cinderv2 | volumev2 | True | admin | http://controller1:8776/v2/%(project_id)s |
| 4e90ff2345634f198959c0cbffed8a21 | RegionOne | nova | compute | True | internal | http://controller1:8774/v2.1 |
| 6764e6e53984890988a0e07ba5f4132 | RegionOne | cinderv3 | volumev3 | True | admin | http://controller1:8776/v3/%(project_id)s |
| 6ce12f2d724d417495a0021e2a126621 | RegionOne | keystone | identity | True | public | http://controller1:5000/v3/ |
| 7286551625e74336a7ca739fd122a280 | RegionOne | neutron | network | True | internal | http://controller1:9696 |
| 73b5a930efa4495e80805c356b58a00e | RegionOne | cinderv3 | volumev3 | True | public | http://controller1:8776/v3/%(project_id)s |
| 75c5a29a353442ac4804db0249851 | RegionOne | nova | compute | True | public | http://controller1:8774/v2.1 |
| 89a54d2093bb42ac9a87bbfdd944e9e | RegionOne | placement | placement | True | public | http://controller1:8778 |
| 92e01b5f6c264a3089c4c180160a5cb | RegionOne | glance | image | True | public | http://controller1:9292 |
| 95ea73959604ce8f573c3d212263e | RegionOne | neutron | network | True | public | http://controller1:9696 |
| 9c0191718e0d2494fb7dcf5b0606090b | RegionOne | cinderv2 | volumev2 | True | internal | http://controller1:8776/v2/%(project_id)s |
| c0c78a13ad0d4e8fbb5b9436798de643 | RegionOne | glance | image | True | admin | http://controller1:9292 |
| c444bb2f76fd45f08a378260c4f70441 | RegionOne | neutron | network | True | admin | http://controller1:9696 |
| d5bb0506334c6987d9144cf58e780 | RegionOne | nova | compute | True | admin | http://controller1:8774/v2.1 |
| e4a0c958c1447f6ad67931c58ac4e64 | RegionOne | swift | object-store | True | internal | http://controller1:8080/v1/AUTH_%(project_id)s |
| f1ad81b91144808060931b3991b22d | RegionOne | glance | image | True | internal | http://controller1:9292 |
| fdbc893af60f4f548ca6478599b74da7 | RegionOne | placement | placement | True | internal | http://controller1:8778 |
+-----+-----+-----+-----+-----+-----+-----+
[root@controller1 ~](keystone_admin)]#
```

Figure 76: La liste des services OpenStack.

```
[root@controller1 ~](keystone_admin)]# openstack endpoint list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Region | Service Name | Service Type | Enabled | Interface | URL |
+-----+-----+-----+-----+-----+-----+-----+
| 08052ba9bfe449eaadc9fcec392ceb8 | RegionOne | swift | object-store | True | admin | http://controller1:8080/v1 |
| 0c3107412f5141c3af12bf8ff97ea7 | RegionOne | cinderv3 | volumev3 | True | internal | http://controller1:8776/v3/%(project_id)s |
| 1d248e49148049938e5eb47d0eabf2a | RegionOne | keystone | identity | True | internal | http://controller1:5000/v3/ |
| 21c6790c374c463d8770ce0e80be1c6f | RegionOne | placement | placement | True | admin | http://controller1:8778 |
| 30b320a60f954403a44c2c7b51edd3db | RegionOne | keystone | identity | True | admin | http://controller1:5000/v3/ |
| 30a7f2703f24c0b341022f50b7e85 | RegionOne | cinderv2 | volumev2 | True | public | http://controller1:8776/v2/%(project_id)s |
| 46020683e5c44b31a1d4d76de67db429 | RegionOne | swift | object-store | True | public | http://controller1:8080/v1/AUTH_%(project_id)s |
| 4803cdf27de943959180882e42184478 | RegionOne | cinderv2 | volumev2 | True | admin | http://controller1:8776/v2/%(project_id)s |
| 4e90ff2345634f198959c0cbffed8a21 | RegionOne | nova | compute | True | internal | http://controller1:8774/v2.1 |
| 6764e6e53984890988a0e07ba5f4132 | RegionOne | cinderv3 | volumev3 | True | admin | http://controller1:8776/v3/%(project_id)s |
| 6ce12f2d724d417495a0021e2a126621 | RegionOne | keystone | identity | True | public | http://controller1:5000/v3/ |
| 7286551625e74336a7ca739fd122a280 | RegionOne | neutron | network | True | internal | http://controller1:9696 |
| 73b5a930efa4495e80805c356b58a00e | RegionOne | cinderv3 | volumev3 | True | public | http://controller1:8776/v3/%(project_id)s |
| 75c5a29a353442ac4804db0249851 | RegionOne | nova | compute | True | public | http://controller1:8774/v2.1 |
| 89a54d2093bb42ac9a87bbfdd944e9e | RegionOne | placement | placement | True | public | http://controller1:8778 |
| 92e01b5f6c264a3089c4c180160a5cb | RegionOne | glance | image | True | public | http://controller1:9292 |
| 95ea73959604ce8f573c3d212263e | RegionOne | neutron | network | True | public | http://controller1:9696 |
| 9c0191718e0d2494fb7dcf5b0606090b | RegionOne | cinderv2 | volumev2 | True | internal | http://controller1:8776/v2/%(project_id)s |
| c0c78a13ad0d4e8fbb5b9436798de643 | RegionOne | glance | image | True | admin | http://controller1:9292 |
| c444bb2f76fd45f08a378260c4f70441 | RegionOne | neutron | network | True | admin | http://controller1:9696 |
| d5bb0506334c6987d9144cf58e780 | RegionOne | nova | compute | True | admin | http://controller1:8774/v2.1 |
| e4a0c958c1447f6ad67931c58ac4e64 | RegionOne | swift | object-store | True | internal | http://controller1:8080/v1/AUTH_%(project_id)s |
| f1ad81b91144808060931b3991b22d | RegionOne | glance | image | True | internal | http://controller1:9292 |
| fdbc893af60f4f548ca6478599b74da7 | RegionOne | placement | placement | True | internal | http://controller1:8778 |
+-----+-----+-----+-----+-----+-----+-----+
[root@controller1 ~](keystone_admin)]#
```

Figure 77: Liste des utilisateurs OpenStack.

5.7 Installation OpenStack

Après l'installation de tous les composants d'OpenStack, Nous allons accéder au tableau de bord OpenStack illustré dans la figure 78 par le compte utilisateur créé dès que l'installation et la configuration se termine convenablement.



Figure 78: OpenStack Dashboard.

On attend quelques instants, et le Dashboard s'affiche et présente une vue d'ensemble avec tous les éléments comme le montre la figure 79.

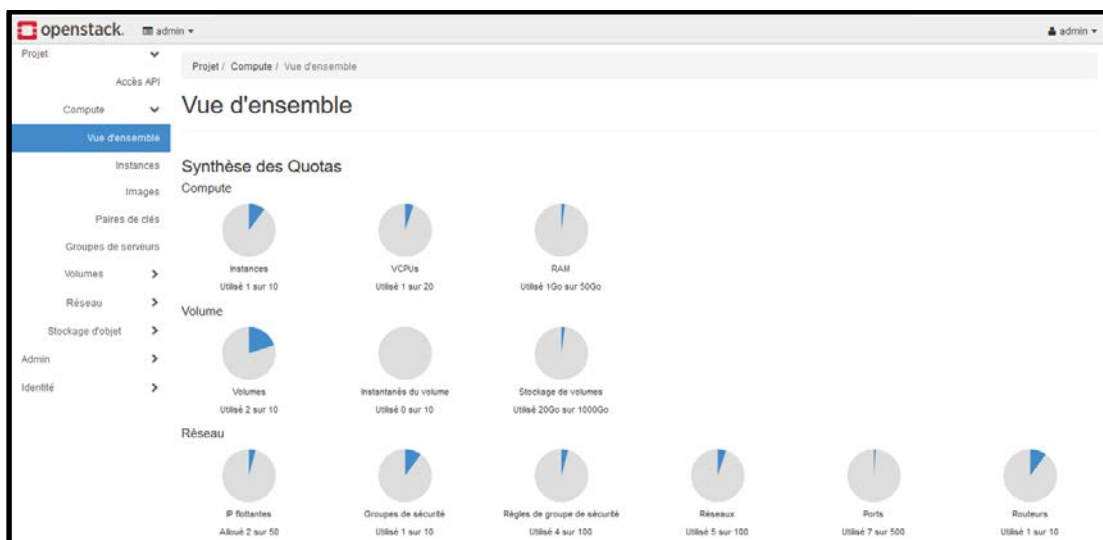


Figure 79: Vue d'ensemble du Dashboard.

Afin de créer une instance on clique dessous, une fenêtre s'affiche ou on doit spécifier le nom de l'instance, la zone de disponibilité est par défaut Nova.

La figure 80 suivante illustre la première étape de création d'instances.

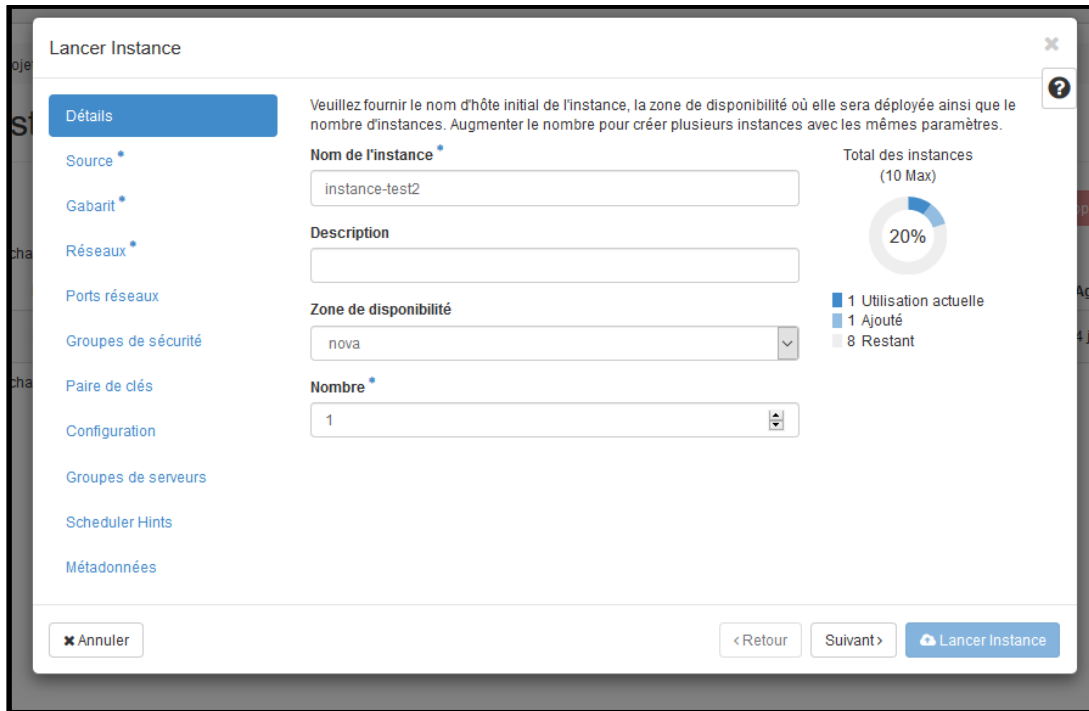


Figure 80: Etape1 de création d'instances.

La figure 81 présente la deuxième étape de création d'instances où à ce niveau on doit spécifier l'image d'os, et le volume pour notre instance.

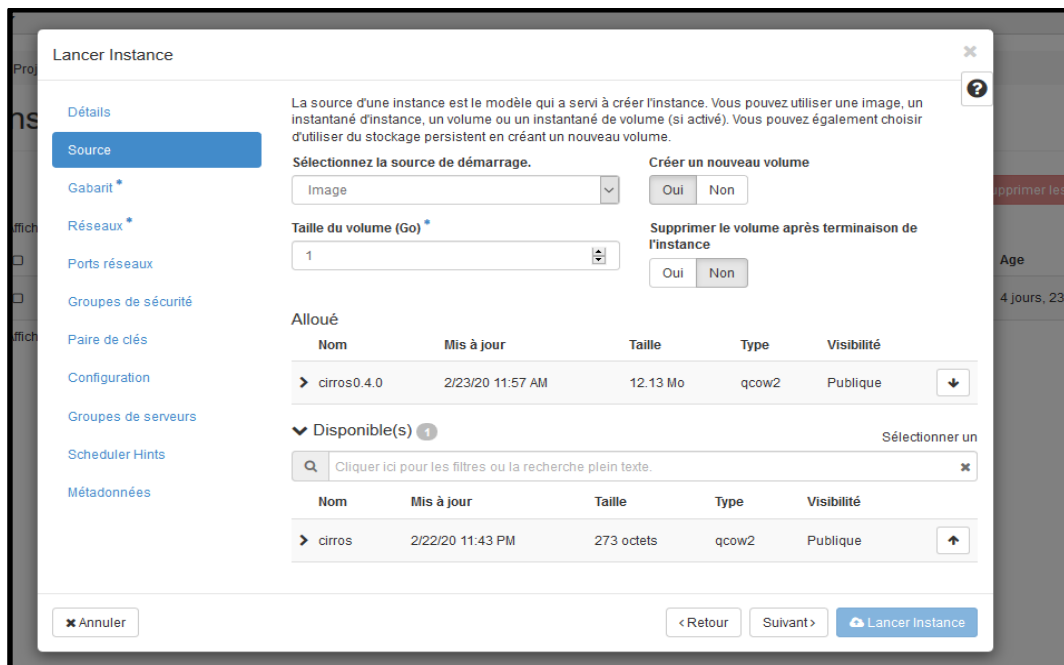


Figure 81: Etape2 de création d'instances.

Dans la figure 82 où la troisième étape est illustrée, on va choisir les gabarits, pour gérer la taille de la capacité de stockage, de mémoire et de calcul d'instance.

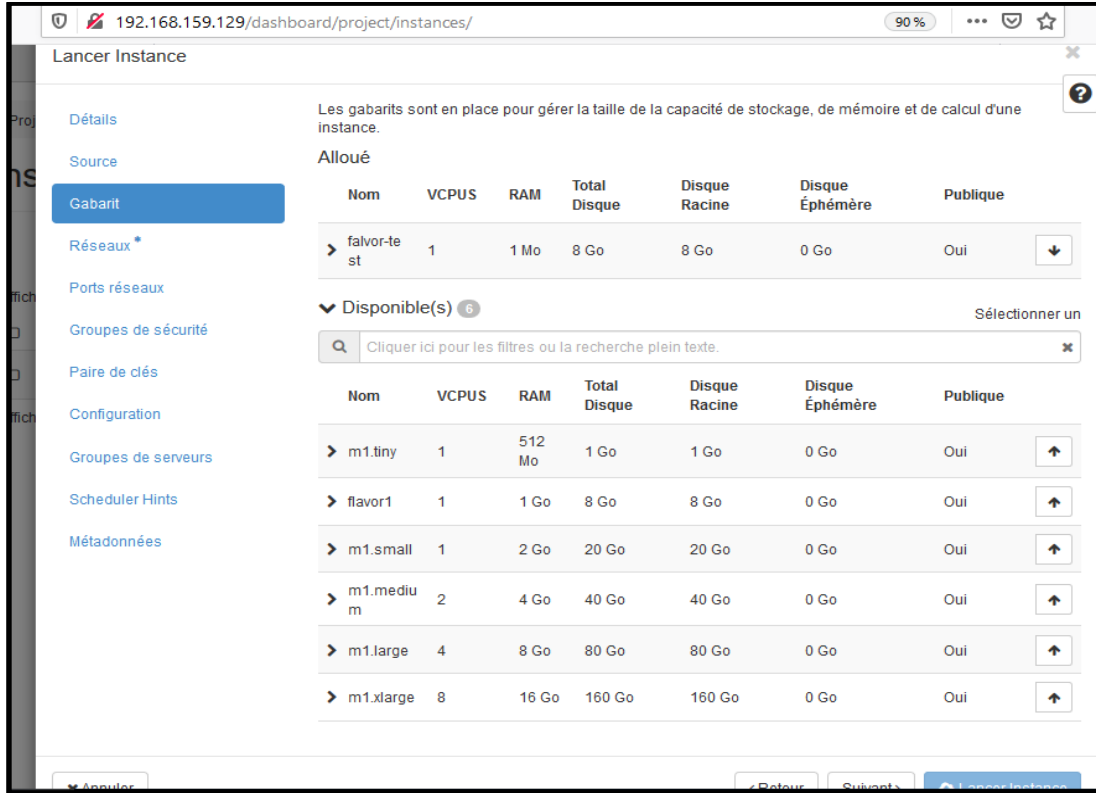


Figure 82: Etape3 de création d'instances.

Comme présenté dans la figure 83, la quatrième étape de création on va choisir le réseau pour notre instance afin de communiquer dans le Cloud.

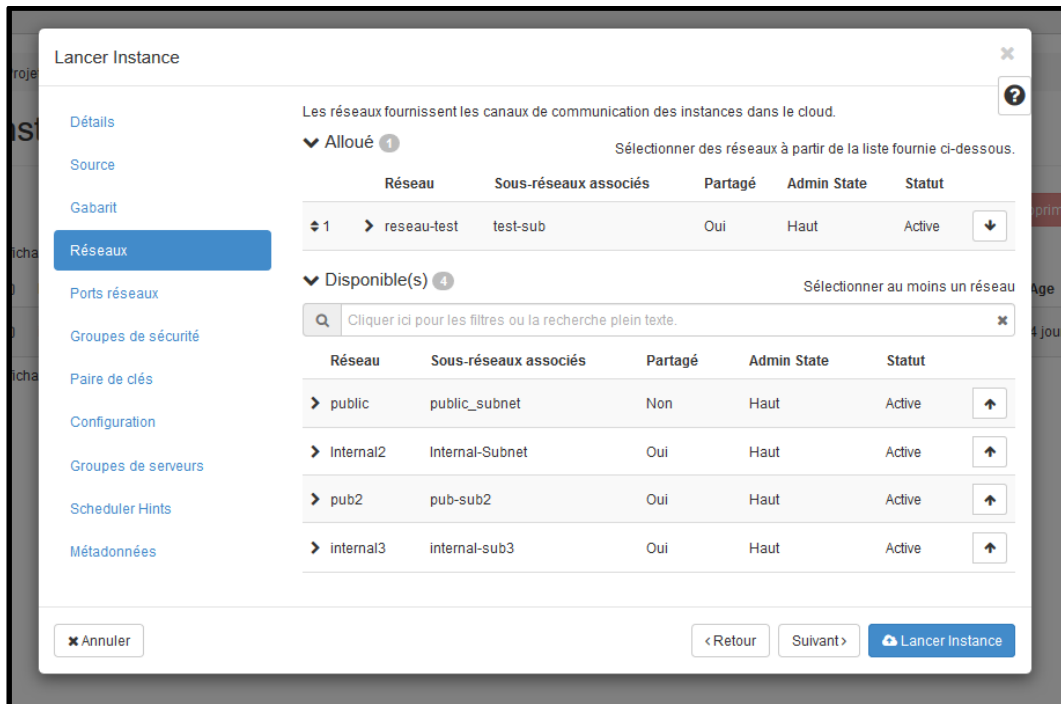


Figure 83: Etape4 de création d'instances.

Ensuite, on lance l'instance, une barre d'état s'affiche comme le montre la figure 84.

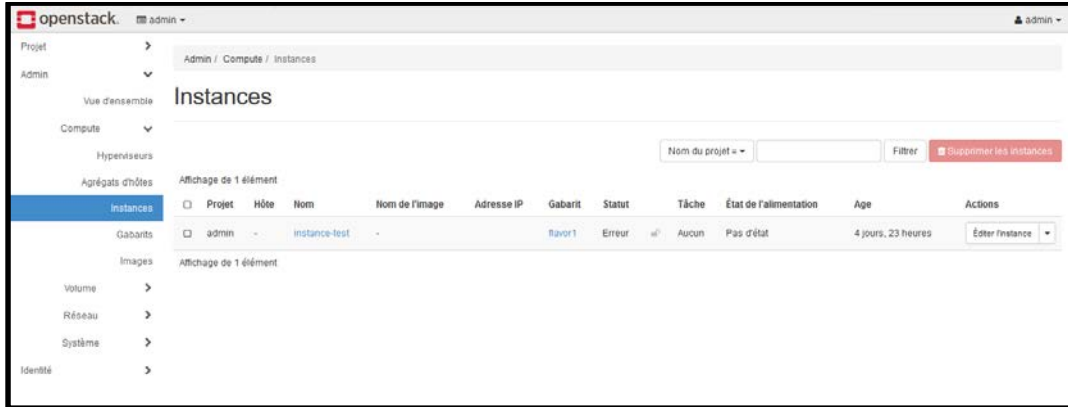


Figure 84: État d'instance.

5.8 Déploiement de l'interface graphique web d'OpenStack

Comme une dernière étape, on va présenter la partie déploiement de l'interface graphique web d'OpenStack, ainsi que son utilisation pour le déploiement d'un site web.

Pour le déploiement du service interface graphique OpenStack nous installons et configurons le service serveur web sur notre contrôleur (service httpd).

La figure 85 montre l'installation et configuration du service de serveur web.

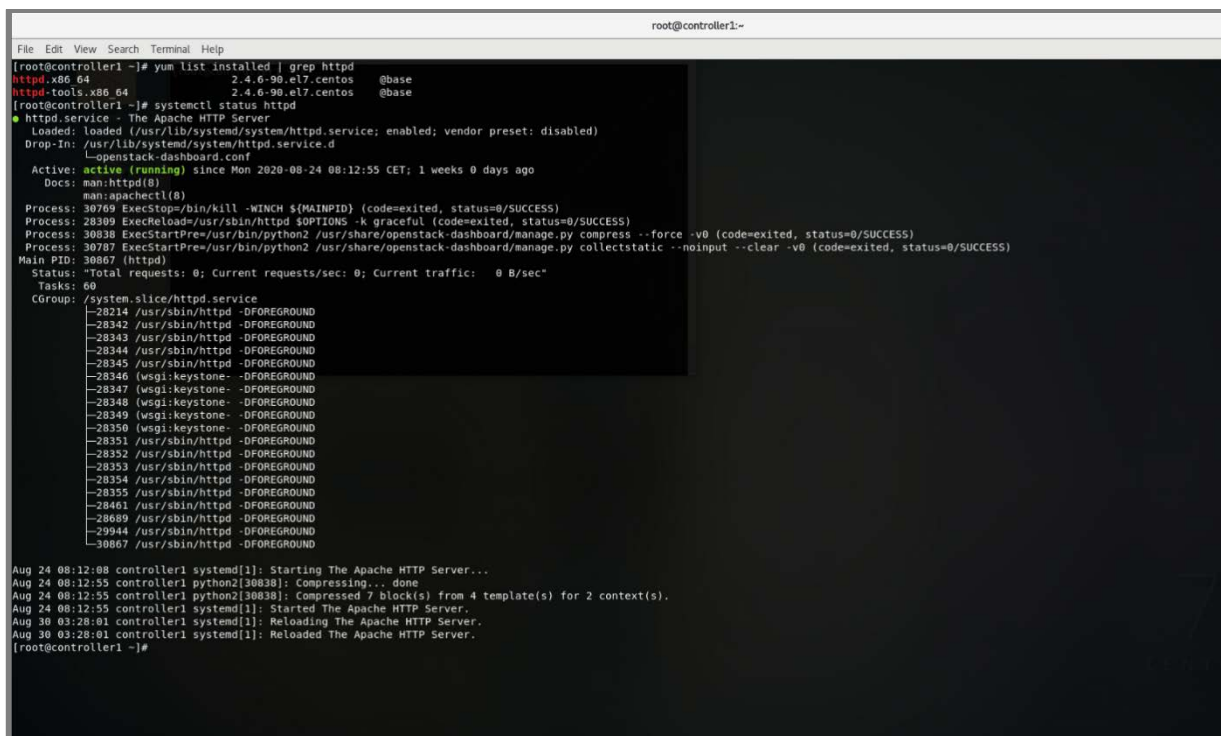


Figure 85: Installation et configuration du service de serveur web.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Quelques configurations du service web OpenStack Horizon comme le montre la figure 86, où nous définissons l'url d'authentification à Keystone pour la page web "http://%s:5000/v3" ("%s = controller1), le contrôleur "OpenStack_Host =", la versions des services Keystone (Identity : 3), Glance (image : 2) et Cinder (volume : 3). Ainsi que les valeurs par défaut pour certaines tâches qui se font par l'interface web, telles que la création d'un utilisateur qui aura par défaut le rôle "user", ou les projets qui auront "Défaut" comme domaine par défaut...

```
# Configure these for your outgoing email host
#EMAIL_HOST = 'smtp.my-company.com'
#EMAIL_PORT = 25
#EMAIL_HOST_USER = 'django@mail'
#EMAIL_HOST_PASSWORD = 'top-secret!'

OPENSTACK_HOST = "controller1"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST

#ADDED MYSELF :
#ENABLE_MULTIPLE_DOMAINS_SUPPORT:
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True

WEBROOT = '/dashboard/'

#API_VERSIONS:
OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 3,
}

OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

# The OPENSTACK_NEUTRON_NETWORK settings can be used to enable optional
# services provided by neutron. Options currently available are load
```

Figure 86: Configurations de service web.

Après l'installation et la configuration du service web nous pourrons y accéder grâce à l'url définie "http://adresseip/dashboard" ou bien "http://nomdedomaine/dashboard". L'interface graphique de service web est montrée dans la figure 87.

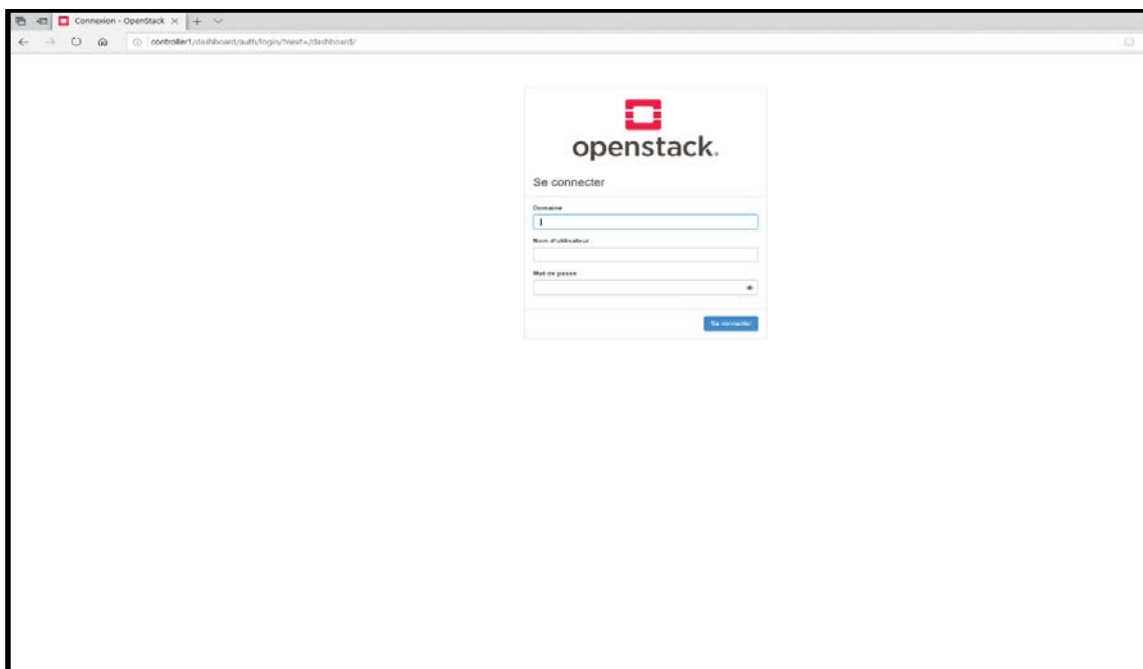


Figure 87: Interface graphique de service web.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Nous accédons alors au projet Administratif d'OpenStack que nous avons créé lors de l'installation en lignes de commande. L'interface graphique de projet administratif d'OpenStack est montrée dans la figure 88.

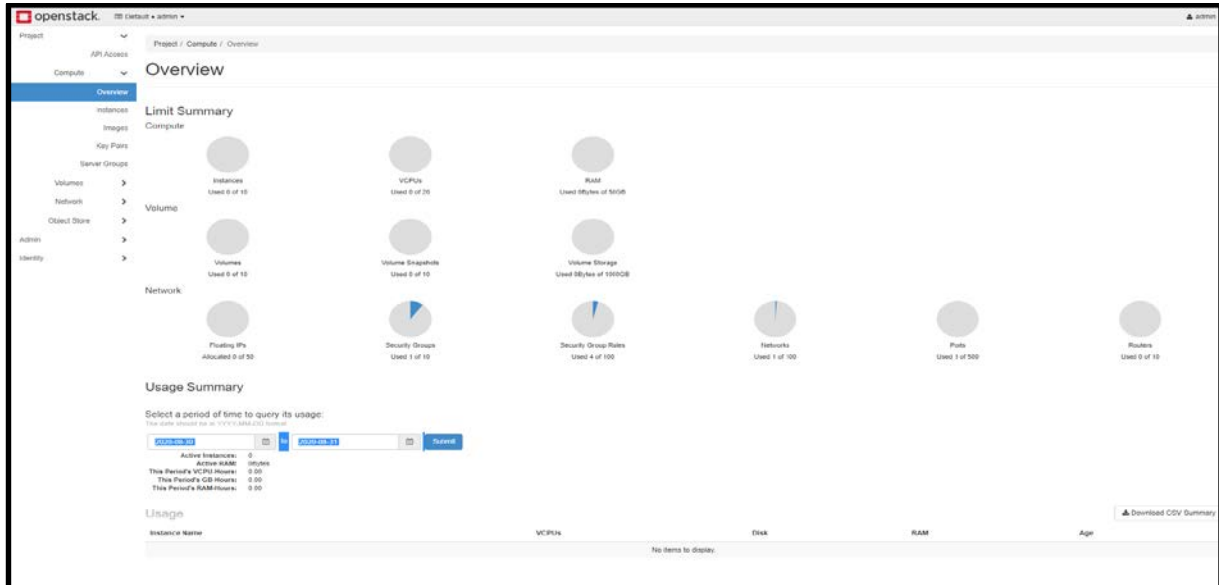


Figure 88: Interface projet administratif d'OpenStack.

La figure 89 montre des informations sur le projet où nous pourrons alors à travers ce projet administratif créer un projet utilisateur pour l'exploitation. Ici "myproject", ensuite un utilisateur que nous affectons à ce projet, ici "myuser" qui a pour rôle "myrole" (rôle que nous avons créé comme rôle pour utilisateurs simples) sur ce projet. Cela est montré dans la figure 90.

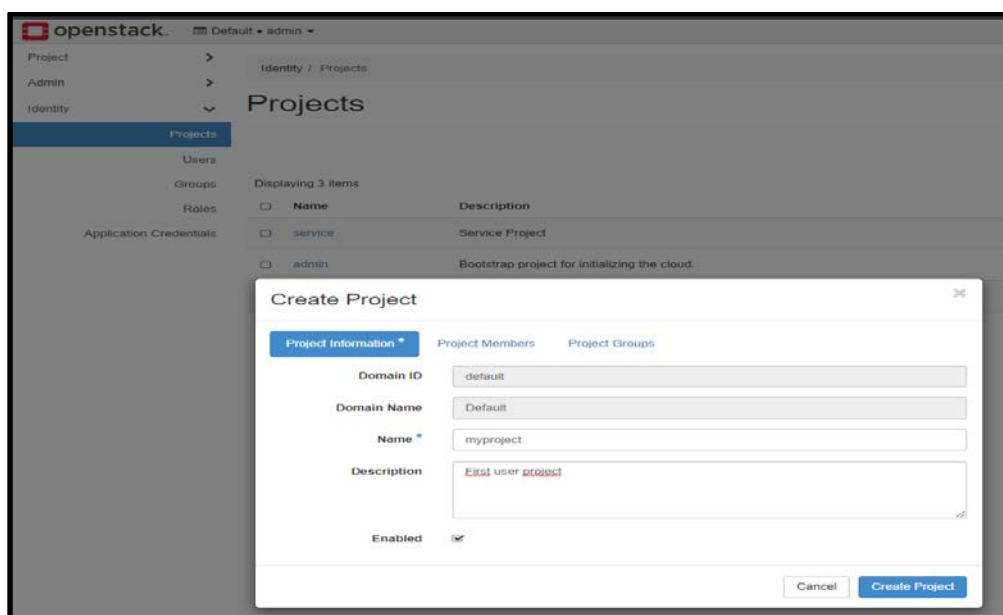


Figure 89: Informations du projet.

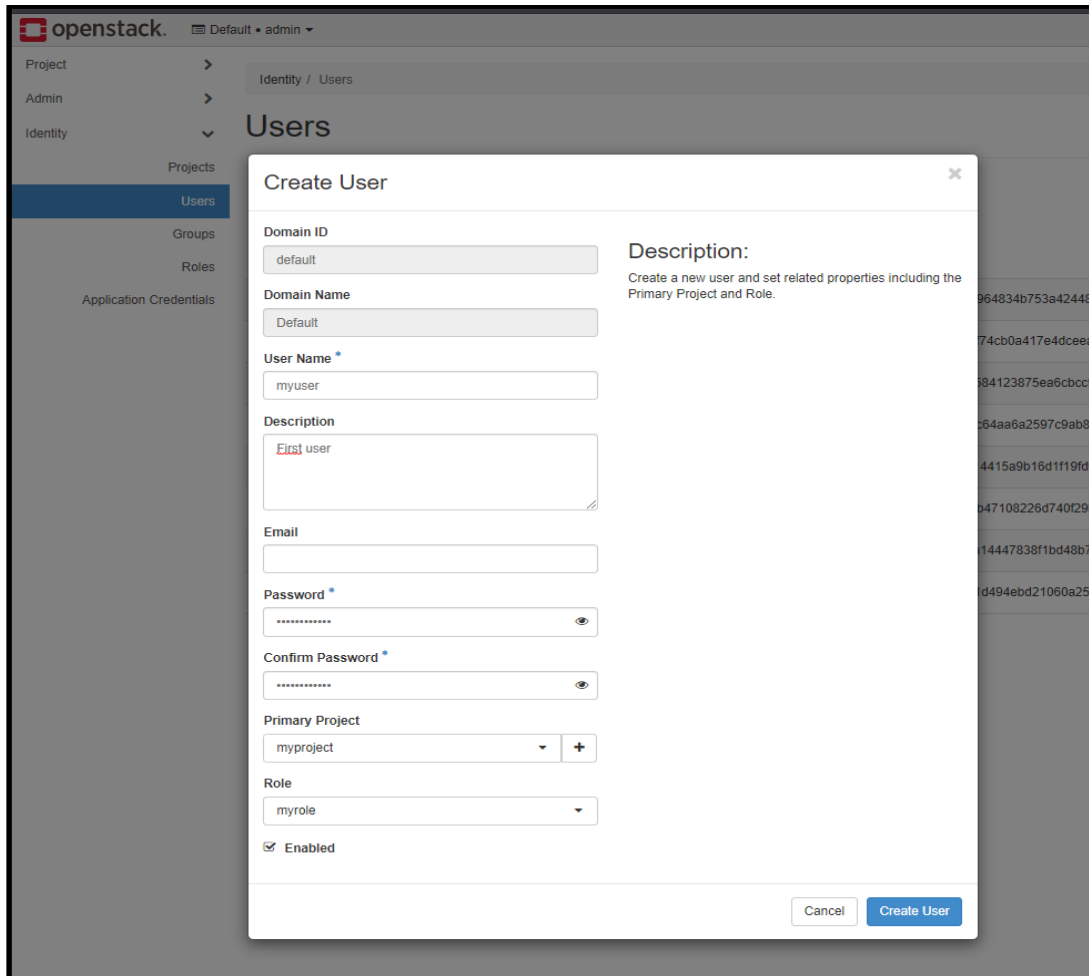


Figure 90: Informations d'utilisateur.

On accède au projet que nous venons de créer à travers l'utilisateur "myuser", (vous pouvez voir en haut à gauche de la page "Default. Myproject" et à droite "myuser"). L'interface graphique de projet est montrée dans la figure 91.

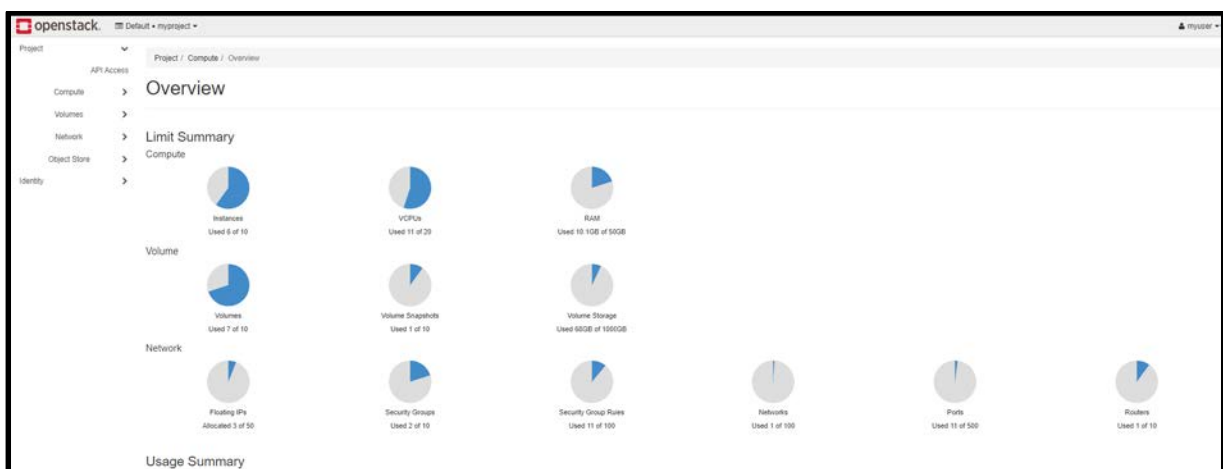


Figure 91: Interface projet.

CHAPITRE 5 : IMPLÉMENTATION ET TEST DE LA SOLUTION OPENSTACK

Ici nous avons créé une instance Linux sur notre Cloud OpenStack dans laquelle on a importé une copie de la configuration du site web officiel de Sonelgaz (Https://www.sonelgaz.dz/) et qu'on a reconfiguré à une adresse IP locale temporaire, ici "10.6.91.91". Tout ça est montré dans la figure 92 qui illustre la reconfiguration du site web officiel de Sonelgaz.

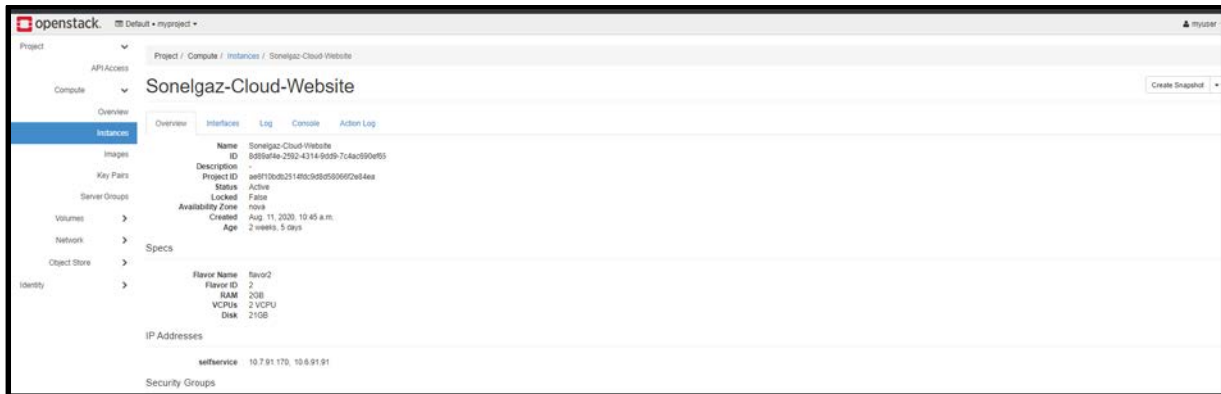


Figure 92: Reconfiguration du site web officiel de Sonelgaz.

Comme nous pouvons le voir, notre copie du site web de Sonelgaz est bien accessible et fonctionnelle de l'extérieur de notre environnement à travers l'adresse IP que nous lui avons affectée <http://10.6.91.91/>.

L'interface graphique du site web déployé est montrée dans la figure 93.

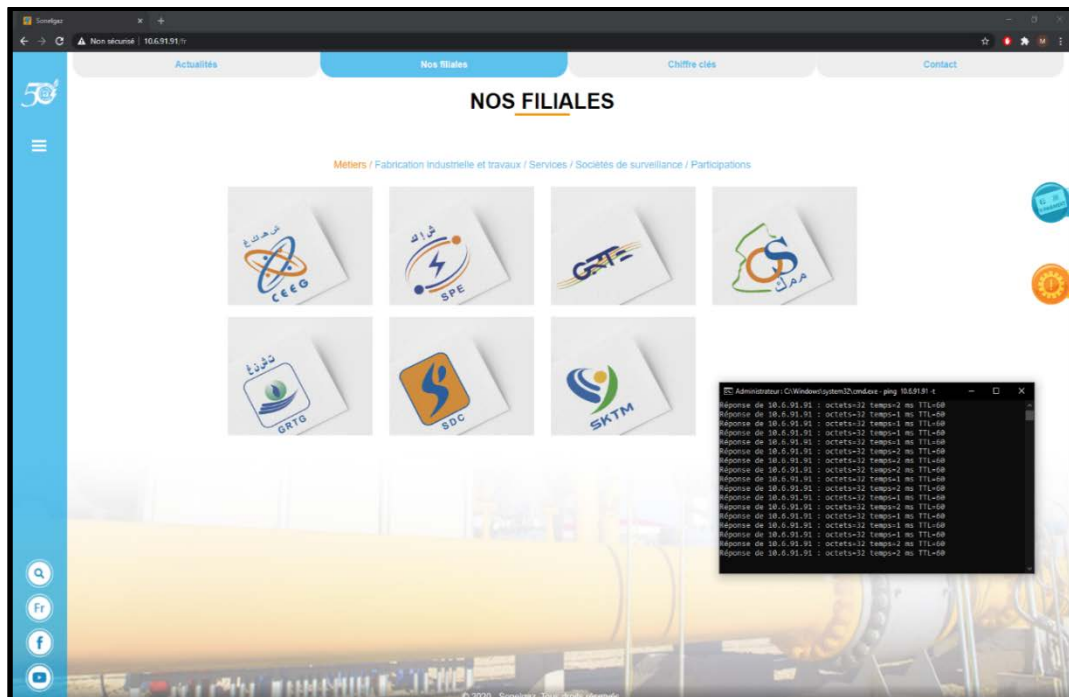


Figure 93: Site web déployé.

5.9 Conclusion

Dans ce chapitre nous avons présenté D'abord l'installation de l'hyperviseur ESXi, ensuite l'installation de VCenter ; on a montré par la suite les étapes suivies pour la préparation des machines virtuelles dans le cluster ainsi qu'avant le service OpenStack, on a cité ensuite les étapes de déploiement des services pour arriver à l'installation d'OpenStack et pour une dernière étape et pour finaliser le travail on a montré le déploiement de l'interface web sur OpenStack.

Conclusion Générale

Le Cloud Computing est un domaine innovant qui répond à la majorité des besoins des entreprises, l'investissement dans ce domaine attire de plus en plus les ingénieurs réseau qui ne cessent de proposer des solutions de Cloud variés en qualité de service fiable et rentable.

Dans le cadre de stage de fin d'étude, nous avons mis en place l'infrastructure Cloud Computing basé sur OpenStack dans des Data Center pour optimiser les services de ce Cloud en virtualisation imbriquée avec son environnement final où on a pu tester la fonctionnalité de cette solution.

Le travail sur ce projet ouvre des perspectives multiples, la majoritaire est d'assurer l'interopérabilité entre les services de cette solution.

L'expérience de ce projet a approfondi nos connaissances dans le vaste domaine de la virtualisation et Cloud Computing.

Bibliographie

- [1] S. R. a. A. Jairath, «“Cloud computing: The fifth generation of computing,”», *Int. Conf. Commun. Syst. Netw. Technol. CSNT 2011*, p. 665–667, 2011.
- [2] H. M. K. Abdeslam, «“ Rapport Cloud Computing, université de henri poincaré ”»,» 2011.
- [3] E. Simmon, «Evaluation of Cloud Computing Services Based on NIST SP 800-145,» *NIST Special Publication*, vol. 500, p. 27, 2018.
- [4] L. M. L. C. J. A. L. M. VAQUERO, «A break in the clouds: towards a cloud definition.,» *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 50-55, JANVIER 2009.
- [5] Le big Data, «IaaS Définition : Qu’est-ce que c’est ? Quels avantages ?», [En ligne]. Available: <https://www.lebigdata.fr/definition-iaas>. [Accès le 31 janvier 2020].
- [6] Le Big Data, «PaaS Définition : Qu’est-ce que c’est ? Quels avantages ?», [En ligne]. Available: <https://www.lebigdata.fr/definition-paas>. [Accès le 31 janvier 2020].
- [7] Wygwam Bureau d'expertise technologie de Aurélien Prunier, « Le Cloud Computing: Réelle révolution ou simple révolution?», 2012. [En ligne]. Available: <https://www.youscribe.com/BookReader/Index/213197/?documentId=182078>. [Accès le 30 janvier 2020].
- [8] «ETUDE ET MISE EN PLACE D'UNE PLATE-FORME WEB 2.0 SUR UNE SOLUTION DE CLOUD COMPUTING -mémoire Master,» [En ligne]. Available: <https://www.memoireonline.com/11/19/11287/tude-et-mise-en-place-d-une-plate-forme-web-20-sur-une-solution-de-cloud-computing.html>. [Accès le 28 JANvier 2020].
- [9] On Click, «Everything as a Service,» 12 novembre 2019. [En ligne]. Available: <https://oneclick-cloud.com/en/blog/trends-en/everything-as-a-service/>. [Accès le 30 janvier 2020].
- [10] Redhat, «Un cloud public, qu’est-ce que c’est ?», [En ligne]. Available: <https://www.redhat.com/fr/topics/cloud-computing/what-is-public-cloud>. [Accès le 31 janvier 2020].
- [11] Redhat, «Qu'est-ce qu'un cloud privé ?», [En ligne]. Available: Qu'est-ce qu'un cloud privé ?. [Accès le 31 janvier 2020].
- [12] Redhat, «Qu'est-ce qu'un cloud hybride ?», [En ligne]. Available: <https://www.redhat.com/fr/topics/cloud-computing/what-is-hybrid-cloud>. [Accès le 31 janvier 2020].

- [13] L. Naugès, «Cloud commutataire : la troisième voie,» 03 avril 2011. [En ligne]. Available: https://nauges.typepad.com/my_weblog/2011/04/cloud-communautaire-la-troisi%C3%A8me-voie-.html. [Accès le 31 janvier 2020].
- [14] «La sécurité et le Cloud Computing,» [En ligne]. Available: <https://fr.slideshare.net/TactikaComInc/la-scurit-et-le-cloud-computing>. [Accès le 12 Septembre 2020].
- [15] «Data center,» 18 JANVIER 2020. [En ligne]. Available: <https://leshorizons.net/datacenter/>. [Accès le 12 SEPTEMBRE 2020].
- [16] «Baies informatiques : des solutions pour répondre à la complexité croissante des besoins et des installations,» 03 DECEMBRE 2018. [En ligne]. Available: <https://www.filiere-3e.fr/2018/12/03/baies-informatiques-des-solutions-pour-repondre-a-la-complexite-croissante-des-besoins-et-des-installations/>. [Accès le 13 SEPTEMBRE 2020].
- [17] Smile, «virtualisation & cloud principes mise en œuvre et outils opensource,» Smile, décembre 2012. [En ligne]. Available: <http://www.asprom.com/technologie/Virtualisation.pdf>. [Accès le 31 01 2020].
- [18] «Quels sont vos hyperviseurs préférés pour faire de la virtualisation des serveurs?,» 30 juin 2016. [En ligne]. Available: <https://www.developpez.com/actu/100679/Quels-sont-vos-hyperviseurs-preferes-pour-faire-de-la-virtualisation-de-serveurs-Et-pourquoi/>. [Accès le 29 janvier 2020].
- [19] «Hyperviseur,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/Hyperviseur>. [Accès le 31 janvier 2020].
- [20] J. Montéremal, «Quel type de virtualisation pour optimiser vos ressources informatiques?,» 05 décembre 2019. [En ligne]. Available: <https://www.appvizer.fr/magazine/services-informatiques/virtualisation/type-virtualisation>. [Accès le 29 janvier 2020].
- [21] «virtualisation des serveurs,» [En ligne]. Available: <http://monge.univ-mlv.fr/~dr/XPOSE2008/virtualisation/techniques.html>. [Accès le 31 janvier 2020].
- [22] «Introduction aux technologies de Virtualisation,» vpourchet, 25 janvier 2011. [En ligne]. Available: <https://vpourchet.com/2011/01/25/introduction-aux-technologies-de-virtualisation-2/>. [Accès le 31 janvier 2020].
- [23] VMware Global Inc, «VMWARE vCLOUD SUITE,» Paris, 2017.
- [24] Amazon Web Services, Inc., «Amazon EC2,» [En ligne]. Available: <https://aws.amazon.com/fr/ec2/>. [Accès le 31 janvier 2020].
- [25] «Chapitre 3 : Les solutions du cloud computing,» 18 juin 2013. [En ligne]. Available: <https://www.institut-numerique.org/chapitre-3-les-solutions-du-cloud-computing-51c0279cafce2>. [Accès le 30 janvier 2020].

- [26] OpenNebula Systems (OpenNebula.io), «Bringing Real Freedom to Your Enterprise Cloud,» [En ligne]. Available: <http://opennebula.org/>. [Accès le 30 janvier 2020].
- [27] The Apache Software Foundation, «Apache CloudStack™,» [En ligne]. Available: <https://cloudstack.apache.org/>. [Accès le 31 janvier 2020].
- [28] «The Most Widely Deployed Open Source Cloud Software in the World,» [En ligne]. Available: <https://www.openstack.org/>. [Accès le 31 janvier 2020].
- [29] R. A. S. K. J. S.Jain, «A Comparative Study for Cloud Computing Platform on OpenSource Software,» *ABHIYANTRIKI: An International Journal of Engineering & Technology*, vol. 1, n° %12, pp. 28-35, décembre 2014.
- [30] Q. MACHU, «Un datacenter dans le cloud,» TOURS, FRANCE, 2015.
- [31] «Zenoss: open cloud computing is three horse race,» 05 OCTOBRE 2012. [En ligne]. Available: <https://www.computerweekly.com/blog/Open-Source-Insider/Zenoss-open-cloud-computing-is-three-horse-race>. [Accès le 15 SPTEMBRE 2020].
- [32] «COMPARATIF DES OUTILS DE VIRTUALISATION,» 20 avril 2011. [En ligne]. Available: <https://www.flecheinthepeche.fr/blog/?p=3959>. [Accès le 30 janvier 2020].
- [33] F.-E. Goffinet, Administration Openstack, 2019.
- [34] «OpenStack Releases,» [En ligne]. Available: <http://releases.openstack.org/>. [Accès le février 2020].
- [35] «Conceptual architecture,» [En ligne]. Available: <https://docs.openstack.org/install-guide/get-started-conceptual-architecture.html>. [Accès le 2020].
- [36] «Logical architecture,» [En ligne]. Available: <https://docs.openstack.org/install-guide/get-started-logical-architecture.html>. [Accès le 2020].
- [37] «Horizon: The OpenStack Dashboard Project,» [En ligne]. Available: <https://docs.openstack.org/horizon/latest>. [Accès le 2020].
- [38] «Keystone, the OpenStack Identity Service,» [En ligne]. Available: <https://docs.openstack.org/keystone/latest/>. [Accès le 2020].
- [39] D. S. e. al., «Self-adaptive authorisation in OpenStackCloud platform,» *Journal of Internet Services and Applications*, p. 17, 2018.
- [40] «OpenStack Compute (nova),» [En ligne]. Available: <https://docs.openstack.org/nova/latest/>. [Accès le 2020].
- [41] «Chapter 4. Understanding Nova,» O'Reilly home, [En ligne]. Available: <https://www.oreilly.com/library/view/deploying-openstack/9781449311223/ch04.html>. [Accès le 29 janvier 2020].

- [42] «Welcome to Glance's documentation!», [En ligne]. Available: <https://docs.openstack.org/glance/latest/>. [Accès le 2020].
- [43] «Chapter 3. Understanding Glance», O'Reilly home, [En ligne]. Available: <https://www.oreilly.com/library/view/deploying-openstack/9781449311223/ch03.html>. [Accès le 29 JANVIER 2020].
- [44] «Welcome to Neutron's documentation!», [En ligne]. Available: <https://docs.openstack.org/neutron/latest/>. [Accès le 2020].
- [45] «Networking architecture», [En ligne]. Available: <https://docs.openstack.org/security-guide/networking/architecture.html>. [Accès le 2 Février 2020].
- [46] «OpenStack Block Storage (Cinder) documentation», [En ligne]. Available: <https://docs.openstack.org/cinder/latest/>. [Accès le 2020].
- [47] «Cinder vs Swift storage in OpenStack - Basic Difference and Comparison», [En ligne]. Available: <https://www.golinuxcloud.com/comparison-difference-between-cinder-vs-swift/>. [Accès le février 2020].
- [48] «Welcome to Swift's documentation!», [En ligne]. Available: <https://docs.openstack.org/swift/latest/>. [Accès le 2020].
- [49] «Vue d'ensemble», [En ligne]. Available: <https://docs.openstack.org/liberty/fr/install-guide-rdo/overview.html>. [Accès le 18 SEPTEMBRE 2020].
- [50] D. Sheppard, «Cloud interoperability and portability – necessary or nice to have?», Insightaas, 29 novembre 2017. [En ligne]. Available: <https://insightaas.com/cloud-interoperability-and-portability-necessary-or-nice-to-have/>. [Accès le 31 janvier 2020].
- [51] B. R. A. B. B.Kezia RAni, «Cloud Computing and Inter-Clouds-Types, Topologies and Research Issues», *Procedia computer science*, vol. 50, pp. 24-29, 2015.
- [52] H. M. K. Abdeslam, «“ Rapport Cloud Computing, université de henri poincaré ”», 2011.