

# TABLE DES MATIERES

Dédicace .....	i
Remerciements .....	ii
Résumé .....	iii
Abstract .....	iii
TABLE DES MATIERES .....	iv
Liste des Figures .....	vi
Liste des Tableaux .....	ix
Liste des Acronymes .....	x
<b>CHAPITRE I : PRESENTATION ET CONTEXTE.....</b>	<b>11</b>
I.1. Introduction.....	12
I.2. Contexte.....	13
I.3. Problématique .....	14
I.4. Objectifs.....	14
<b>CHAPITRE II : ETAT DE L'ART .....</b>	<b>15</b>
<b>Partie 1 : Etat de l'art sur le Cloud Computing .....</b>	<b>16</b>
1.1. Définition du cloud computing .....	16
1.1.1. Modèles de déploiement du Cloud Computing .....	17
1.1.2. Types de service du Cloud Computing .....	18
1.1.3. Les principaux fournisseurs de services Cloud .....	18
1.2. Le Cloud Computing Mobile (MCC).....	20
1.2.1. Qu'est-ce que le Cloud Computing Mobile .....	20
1.2.2. Architecture du Cloud Computing Mobile .....	21
1.2.3. Les applications mobiles du MCC.....	21
1.3. La virtualisation .....	22
1.3.1. Types de virtualisations.....	23
1.3.2. Les Techniques de Virtualisation.....	24
1.3.3. Quelques solutions de virtualisation.....	25
1.4. Cloud Computing et système d'information .....	28
<b>Partie 2 : Etat de l'art sur les plateformes de formation à distance.....</b>	<b>30</b>
2.1. Introduction.....	30
2.2. Quelques plateformes de formation à distance .....	30
<b>Partie3 : Choix technologique .....</b>	<b>32</b>

<b>CHAPITRE III : PROPOSITION D'UNE ARCHITECTURE HAUTEMENT DISPONIBLE POUR UN DISPOSITIF D'E-LEARNING .....</b>	<b>33</b>
III.1. Rappels sur la gestion de la haute disponibilité des systèmes dans le Cloud Computing.....	34
III.1.1. La tolérance aux pannes.....	34
III.1.2. La réplication des données.....	35
III.1.3. L'équilibrage de charge (Load Balancing) .....	35
III.2. Gestion de la haute disponibilité des dispositifs .....	37
III.2.1. Gestion de la haute disponibilité de Moodle.....	38
III.2.2. Gestion de la haute disponibilité de BigBlueButton .....	38
III.3. Architecture proposée .....	39
III.3.1. Système Moodle .....	40
III.3.2. Système BigBlueButton.....	40
III.4. Principe et fonctionnement .....	41
III.5. Diagramme des flux .....	43
III.6. Maquette de test .....	44
III.7. Implémentation .....	45
III.7.1. Mise en place de Proxmox Virtual Environnement.....	45
III.7.2. Implémentation du système Moodle.....	46
III.7.3. Implémentation du système BigBlueButton .....	68
III.7.4. Intégration du système Moodle et Scalelite .....	84
III.7.5. Tests et Validation.....	90
<b>CONCLUSION GENERALE ET PERSPECTIVES .....</b>	<b>97</b>
<b>BIBLIOGRAPHIE.....</b>	<b>98</b>
<b>WEBOGRAPHIE .....</b>	<b>99</b>

# Liste des Figures

Figure 1: Architecture du système existant .....	13
Figure 2: Cloud Computing .....	16
Figure 3: Modèles de déploiement du Cloud Computing .....	17
Figure 4: Types de service du Cloud Computing .....	18
Figure 5: Principaux fournisseurs services Cloud .....	19
Figure 6: Architecture du Cloud Computing Mobile [2] .....	21
Figure 7: Serveur virtualisé .....	23
Figure 8: Isolation [13] .....	24
Figure 9: La paravirtualisation [13] .....	25
Figure 10 : La virtualisation complète [14] .....	25
Figure 11: Système d'Information interne .....	28
Figure 12: Système d'Information externalisé .....	28
Figure 13: L'algorithme Round Robin .....	36
Figure 14: L'algorithme Weighted Round Robin .....	36
Figure 15: L'algorithme Least Connection .....	37
Figure 16: Architecture proposée .....	39
Figure 17: Diagramme fonctionnel .....	41
Figure 18: Diagramme des flux .....	43
Figure 19: Prototype de test .....	44
Figure 20: Installation Proxmox VE .....	45
Figure 21: Interface Proxmox .....	46
Figure 22: Configuration des noms d'hôtes .....	47
Figure 23: Vérification de la version MariaDB .....	48
Figure 24: Configuration nœud 1 .....	49
Figure 25: Configuration nœud 2 .....	50
Figure 26: Configuration nœud 3 .....	51
Figure 27: Taille du cluster .....	52
Figure 28: Test de la réplication entre les nœuds .....	52
Figure 29: Informations d'authentification .....	53
Figure 30: Etat de la table mysql_servers .....	55
Figure 31: Connection aux nœuds depuis le ProxySQL .....	56

<b>Figure 32: Statut du service HAProxy</b> .....	57
<b>Figure 33: Statut de SELinux</b> .....	58
<b>Figure 34: Configuration du serveur syslog</b> .....	58
<b>Figure 35: Configuration du HAProxy</b> .....	59
<b>Figure 36: Connection au serveur web_1 depuis le HAProxy</b> .....	60
<b>Figure 37: Connection au serveur web_2 depuis le HAProxy</b> .....	60
<b>Figure 38: Utilisateur de la base de données du serveur Moodle</b> .....	63
<b>Figure 39: Choix de la langue de configuration</b> .....	64
<b>Figure 40: Configuration du dossier Moodledata</b> .....	65
<b>Figure 41: Choix du type de base de données</b> .....	65
<b>Figure 42: Enregistrement de la base de données de Moodle</b> .....	66
<b>Figure 43: Vérification des modules d'installation de Moodle</b> .....	66
<b>Figure 44: Création d'un administrateur du site Moodle</b> .....	67
<b>Figure 45: Interface de connexion du site</b> .....	67
<b>Figure 46: Interface de connexion du site (2)</b> .....	68
<b>Figure 47: Paramètres régionaux du serveur</b> .....	69
<b>Figure 48: Vérification de la mémoire disponible</b> .....	69
<b>Figure 49: Vérification de la version d'Ubuntu</b> .....	70
<b>Figure 50: Vérification du nombre cœur du serveur</b> .....	70
<b>Figure 51: Information du serveur BigBlueButton</b> .....	71
<b>Figure 52: Vérification des dépôts</b> .....	71
<b>Figure 53: Redémarrage du serveur</b> .....	73
<b>Figure 54: Statut du serveur BigBlueButton</b> .....	73
<b>Figure 55: Information du serveur BigBlueButton</b> .....	74
<b>Figure 56: Interface de connexion au serveur BigBlueButton</b> .....	74
<b>Figure 57: Connexion à un meeting</b> .....	75
<b>Figure 58: Vérification de l'OS du serveur</b> .....	76
<b>Figure 59: Ajout de la clé CPG de docker</b> .....	77
<b>Figure 60: Vérification de la version du docker</b> .....	78
<b>Figure 61: Téléchargement de Scalelite</b> .....	79
<b>Figure 62: Obtention de clé secret</b> .....	79
<b>Figure 63: Obtention de la valeur du Loadbalancer</b> .....	80
<b>Figure 64: Contenu du fichier .env</b> .....	80

<b>Figure 65: Liste des serveurs BigBlueButton .....</b>	<b>81</b>
<b>Figure 66: Ajout du premier serveur BigBlueButton.....</b>	<b>82</b>
<b>Figure 67: Statut des serveurs BigBlueButton ajoutés.....</b>	<b>83</b>
<b>Figure 68: Informations de Scalelite .....</b>	<b>84</b>
<b>Figure 69: Connection dans notre plateforme Moodle.....</b>	<b>84</b>
<b>Figure 70: Etape d'installation Plugin BigBlueButton.....</b>	<b>85</b>
<b>Figure 71: Installation du plugin depuis le compte Moodle.....</b>	<b>86</b>
<b>Figure 72: Téléchargement du plugin BigBlueButton.....</b>	<b>86</b>
<b>Figure 73: Choix du site d'installation .....</b>	<b>87</b>
<b>Figure 74: Installation à partir du répertoire des plugins .....</b>	<b>87</b>
<b>Figure 75: Mise à jour de la base de données .....</b>	<b>88</b>
<b>Figure 76: Module ajouté .....</b>	<b>88</b>
<b>Figure 77: Liste des plugins .....</b>	<b>89</b>
<b>Figure 78: Intégration Moodle et Scalelite .....</b>	<b>89</b>
<b>Figure 79: Première connexion.....</b>	<b>90</b>
<b>Figure 80: Deuxième connexion.....</b>	<b>91</b>
<b>Figure 81: Etat initial de Scalelite .....</b>	<b>92</b>
<b>Figure 82: Première classe virtuelle .....</b>	<b>92</b>
<b>Figure 83: Première classe virtuelle (1) .....</b>	<b>93</b>
<b>Figure 84: Etat Scalelite (1).....</b>	<b>93</b>
<b>Figure 85: Deuxième classe virtuelle .....</b>	<b>94</b>
<b>Figure 86: Etat Scalelite (2).....</b>	<b>94</b>
<b>Figure 87: Troisième classe virtuelle.....</b>	<b>95</b>
<b>Figure 88: Etat Scalelite (3).....</b>	<b>95</b>
<b>Figure 89: Quatrième classe virtuelle .....</b>	<b>96</b>
<b>Figure 90: Etat Scalelite (4).....</b>	<b>96</b>

# Liste des Tableaux

<b>Tableau 1: Etude comparative des solutions de virtualisations.....</b>	<b>27</b>
<b>Tableau 2: Etude comparative de ces plateformes d'e-learning .....</b>	<b>32</b>
<b>Tableau 3 : Listes des participants aux réunions.....</b>	<b>91</b>

# Liste des Acronymes

<b>API</b>	<b>Application Programming Interface</b>
<b>BBB</b>	<b>BigBlueButton</b>
<b>CC</b>	<b>Cloud Computing</b>
<b>CPU</b>	<b>Central Processing Unit</b>
<b>FAD</b>	<b>Formation A Distance</b>
<b>FOAD</b>	<b>Formation Ouverte A Distance</b>
<b>HA</b>	<b>High Availability</b>
<b>HTTP</b>	<b>HyperText Transfer Protocol</b>
<b>IaaS</b>	<b>Infrastructure as a Service</b>
<b>IP</b>	<b>Internet Protocol</b>
<b>IT</b>	<b>Information Technology</b>
<b>KVM</b>	<b>Kernel-Based Virtual Machine</b>
<b>LMS</b>	<b>Learning Management System</b>
<b>LXC</b>	<b>Linux Containers</b>
<b>MCC</b>	<b>Mobile Cloud Computing</b>
<b>M-Commerce</b>	<b>Mobile-Commerce</b>
<b>M-Learning</b>	<b>Mobile -Learning</b>
<b>M-Santé</b>	<b>Mobile -Santé</b>
<b>NIST</b>	<b>National Institut of Standards and Technology</b>
<b>OS</b>	<b>Operating System</b>
<b>PaaS</b>	<b>Platform as a Service</b>
<b>RAM</b>	<b>Random Access Memory</b>
<b>SaaS</b>	<b>Software as a Service</b>
<b>SI</b>	<b>Système d'Information</b>
<b>SIP</b>	<b>Session Initiation Protocol</b>
<b>SQL</b>	<b>Structured Query Language</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>TIC</b>	<b>Techniques de l'information et de la communication</b>

# **CHAPITRE I : PRESENTATION ET CONTEXTE**

## I.1. Introduction

Avec le développement rapide des technologies de traitement, de stockage, et avec le succès de l'internet, des ressources informatiques sont devenues moins chères, plus puissantes et plus disponibles que jamais. Cette tendance technologique a permis la réalisation d'un nouveau modèle informatique appelé Cloud Computing.

Le cloud Computing est un concept qui ne cesse d'évoluer et d'attirer un nombre important de clients en raison de ses avantages y compris la facilité de déploiement, la disponibilité, l'efficacité, la flexibilité, etc.

Les fournisseurs Cloud utilisent la technologie de virtualisation qui leur permet de virtualiser les serveurs, le stockage ou d'autres ressources matérielles physiques ou de centre de données, ce qui peut ensuite leur permettre d'offrir à la demande des services sans procéder à de lourds investissements dans de nouvelles infrastructures et en externalisant une partie des données et des logiciels sur des serveurs distants.

La plupart des entreprises évitent les équipements informatiques internes au profit de la location d'infrastructures, de plateformes et de logiciels informatiques dans le Cloud, car ce dernier offre un moyen plus flexible et plus efficace de gérer les dépenses informatiques de l'entreprise.

Les institutions scolaires telles que les établissements d'enseignement supérieur et universités font recours au Cloud qui leur permet de profiter de nombreuses ressources telles que : stockage, de calcul, de traitement, etc..., car les systèmes d'apprentissage en ligne qu'ils utilisent nécessitent des ressources logicielles importantes et de nombreux établissements ne peuvent pas se permettre de tels investissements.

En effet, l'apprentissage en ligne (e-learning) est un type de formation à distance qui utilise Internet et les nouvelles technologies digitales. Les plateformes d'e-learning sont de plus en plus utilisées surtout en ces périodes de crise du covid-19 qui rend leur utilisation encore plus d'actualité, car beaucoup d'employés et étudiants sont obligés de travailler à distance.

Cependant, si par exemple un établissement comme l'Université Cheikh Anta Diop de Dakar (UCAD) décide de mettre en place ces dispositifs d'enseignement à distance dans le Cloud, il sera obligé de gérer son système éducatif afin qu'elle soit hautement disponible. Le fournisseur de Cloud ne sera pas responsable des problèmes d'indisponibilité des services du système de l'UCAD mais plutôt responsable de l'infrastructure allouée.

Notre mémoire porte sur l'étude et le déploiement d'une infrastructure IT Cloud hautement disponible appliquée à l'e-learning. Après une étude détaillée sur le système Cloud, sur les plateformes d'e-learning et sur les services de gestion de haute disponibilité, nous proposons ensuite une architecture pour un dispositif d'enseignement en ligne que nous allons tester dans la solution de virtualisation choisie.

Ce mémoire s'organise en trois parties :

- La première partie porte sur la présentation générale du mémoire qui est composé d'une introduction générale, du contexte, la problématique, des objectifs, et de la méthodologie suivie pour l'élaboration du mémoire.

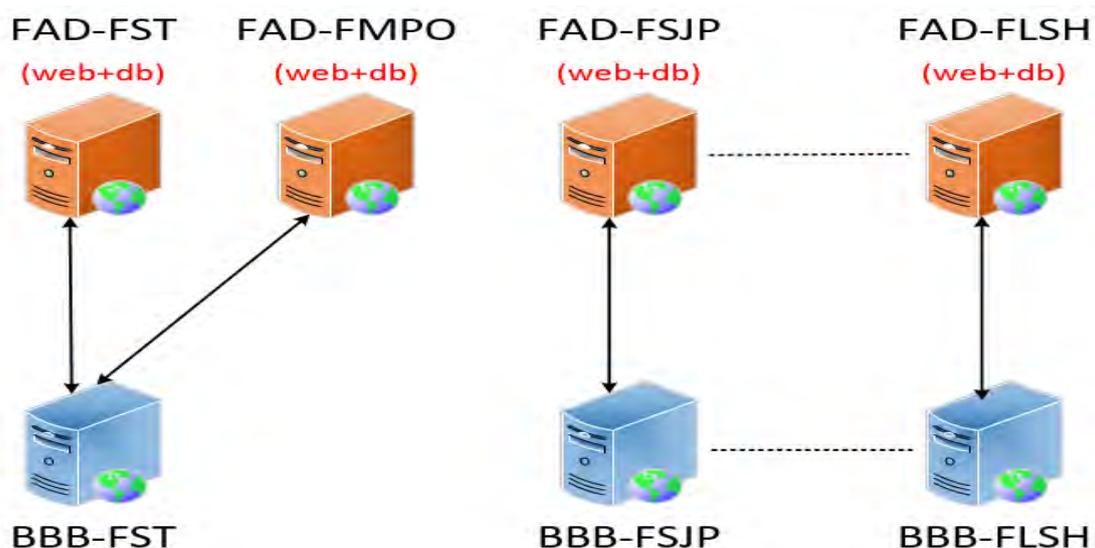
- La deuxième partie présente un état de l'art détaillé sur les cloud Computing, sur les plateformes de formations à distance. Une étude comparative de quelques plateformes est présentée à cet effet.
- Dans la troisième partie, nous proposons une architecture pour un dispositif d'enseignement à distance hautement disponible dans un environnement virtualisé. Après avoir présenté l'architecture, nous poursuivrons avec les installations et la configuration des différentes solutions décrites.

## I.2. Contexte

L'université Cheikh Anta Diop de Dakar (UCAD) utilise de nouvelles méthodes pédagogiques permettant d'exploiter au mieux les Technologies de l'Information et de la Communication (TIC).

Le Département de Ressources Technologiques et Pédagogiques (DRTP) rattaché à la Direction de l'Information et des Systèmes d'Information (DISI), offre des formations en ligne à travers ses plateformes de Formation Ouvertes À Distance (FOAD). Le département accompagne également chaque faculté de l'UCAD dans la mise en place de sa plateforme LMS, de la conception à la mise en ligne pour le suivi des apprenants et la diffusion en ligne de supports pédagogiques.

C'est le cas de la Faculté des Sciences Techniques (FST) (<https://fad.fst.ucad.sn/>), de la Faculté de Médecine, de Pharmacie et d'Odontologie (FMPO) (<https://fad.fmppo.ucad.sn/>), de la Faculté des Sciences Économiques et de Gestion (FASEG) (<https://fad.faseg.ucad.sn/>), de la Faculté des Sciences Juridiques et Politiques (FSJP) (<https://fad-fs.jp.ucad.sn/>), de la Faculté des Sciences et Technologies de l'Éducation et de la Formation (FASTEF) (<https://fad-fastef.ucad.sn/>), de la Faculté des Lettres et Sciences Humaines (FLSH) (<https://fad.flsh.ucad.sn/>).



**Figure 1: Architecture du système existant**

Dans la figure ci-dessus, chaque plateforme FAD (Formation A Distance) d'une faculté est connectée à un serveur BBB (BigBluButton) pour les conférences web à l'exception de la FMPO qui est connecté au serveur BBB de la FST ; et dans chaque serveur FAD se trouve tous les composants d'une plateforme à savoir le code, la base de données et le système de fichier.

### I.3. Problématique

Avec l'architecture du système existant des limites et risques se présentent comme :

- La surutilisation d'un serveur ;
- La sous-utilisation d'un serveur à cause de la faible utilisation ou du nombre d'étudiant ;
- Manques de ressources à cause d'un nombre d'étudiant très élevé d'une faculté.

Notons aussi que le système existant ne garantit pas la haute disponibilité horizontale des plateformes FAD et des serveurs de conférences web.

Puisque la haute disponibilité est une exigence clé pour les plateformes d'enseignement à distance que nous intégrons le plus souvent avec d'autres services d'enseignement, alors :

- Comment mettre en place une solution hautement disponible pour prendre en charge la disponibilité des plateformes pour un nombre d'utilisateur important ?
- Quel dispositif hautement disponible mettre en place pour les sessions synchrones des classes virtuelles ?

### I.4. Objectifs

L'objectif général de ce mémoire est de proposer une architecture hautement disponible pour un dispositif d'enseignement à distance dans un environnement virtualisé.

Avec cette architecture, les plateformes vont pouvoir partager les charges de travail ce qui permet de diminuer le temps d'exécution des demandes et d'optimiser leur utilisation. Elles vont également permettre aux classes virtuelles intégrées avec ces plateformes d'accueillir une quantité d'étudiants très importante.

Pour y parvenir, nous nous fixons les objectifs spécifiques suivants :

- Faire un état de l'art sur les systèmes de Cloud Computing, sur les plateformes d'e-learning et choisir les technologies à utiliser ;
- Proposer une architecture qui va assurer une haute disponibilité des plateformes Moodle et des services BigBluButton ;
- Déployer et mettre en service un prototype de test ;

## **CHAPITRE II : ETAT DE L'ART**

# Partie 1 : Etat de l'art sur le Cloud Computing

L'une des tendances technologiques susceptible d'avoir un impact significatif sur l'environnement d'apprentissage en ligne est le Cloud Computing.

L'infrastructure du Cloud Computing est généralement composée de plusieurs centres de données. La fourniture des ressources informatiques et services Cloud est contrôlée par un fournisseur et ces ressources sont allouées en fonction des besoins des clients.

Dans ce chapitre nous allons présenter les concepts de bases du Cloud Computing, le système Cloud Computing Mobile, parler sur les systèmes d'information Cloud et en dernier lieu sur la gestion de la haute disponibilité dans les systèmes Cloud Computing.

## 1.1. Définition du cloud computing

Il existe plusieurs définitions du concept Cloud Computing, ce qui prouve l'inexistence de définition exacte acceptée universellement.

Par exemple, selon le National Institut of Standards and Technology, NIST « Le cloud Computing est un modèle permettant un accès réseau pratique à la demande à un pool partagé de ressources informatiques configurables (par exemple réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement provisionnés et libérés avec une gestion minimale en interaction avec le fournisseur ou le prestataire de service » [1].

Certains le définissent comme un ensemble de service déployé sur un réseau accessible n'importe quand et n'importe où ; un concept qui consiste à déplacer sur des serveurs distants le stockage et le traitement informatique généralement situé sur des serveurs locaux.

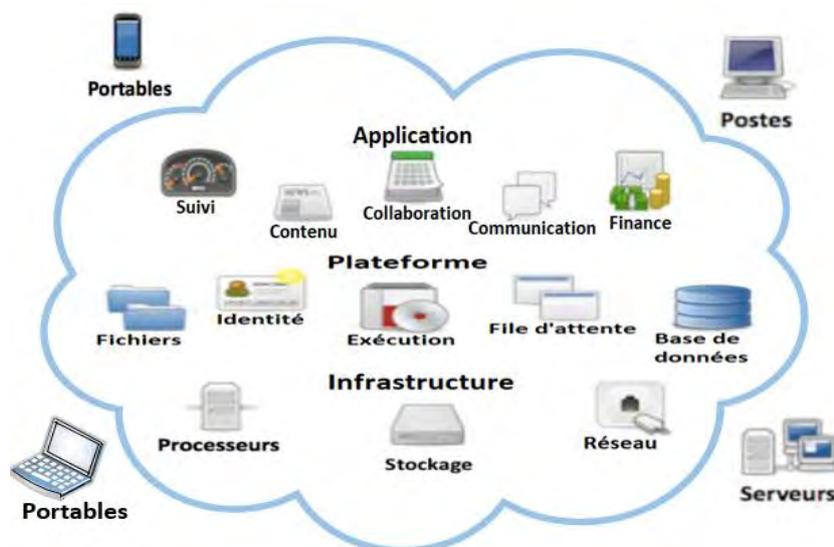
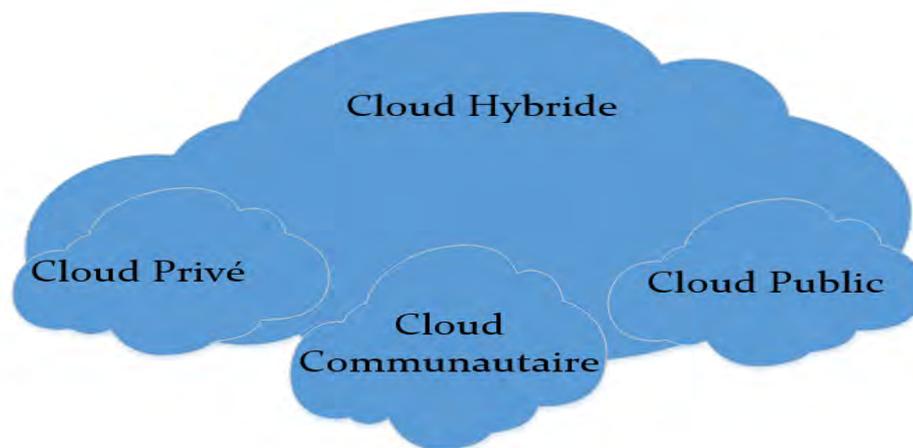


Figure 2: Cloud Computing

On peut cependant, s'accorder sur le fait que le Cloud Computing est l'exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement internet [10].

### 1.1.1. Modèles de déploiement du Cloud Computing

Il existe quatre modèles de déploiement selon le NIST comme le montre la figure ci-après :



**Figure 3: Modèles de déploiement du Cloud Computing**

- **Cloud privé** : l'infrastructure cloud est provisionnée pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (par exemple, des unités commerciales). Il peut être détenu, géré et exploité par l'organisation, un tiers ou une combinaison des deux, et il peut exister sur place ou à l'extérieur. [11]
- **Cloud communautaire** : l'infrastructure cloud est provisionnée pour une utilisation exclusive par un communauté de consommateurs d'organisations partageant des préoccupations (p. ex., mission, exigences de sécurité, politique et considérations de conformité). Il peut être possédé, géré et exploité par une ou plusieurs des organisations de la communauté, un tiers partie, ou une combinaison d'entre eux, et il peut exister sur ou hors des lieux. [11]
- **Cloud public** : l'infrastructure cloud est configurée pour une utilisation ouverte par le grand public. Il peut être, géré et exploité par une entreprise, un universitaire ou une organisation gouvernementale, ou une combinaison d'entre eux. Il existe dans les locaux du fournisseur de cloud. [11]
- **Cloud hybride** : l'infrastructure cloud est une composition de deux ou plusieurs cloud distincts infrastructures (privées, communautaires ou publiques) qui restent des entités uniques, mais sont liées ensemble par une technologie standardisée ou propriétaire qui permet des données et des applications la portabilité (par exemple, l'éclatement du cloud pour l'équilibrage de charge entre les cloud). [11]

## 1.1.2. Types de service du Cloud Computing

Il existe trois principaux services proposés en Cloud :



Figure 4: Types de service du Cloud Computing

- **La couche l'Infrastructure en tant que Service (IaaS)** : les ressources du Cloud sont proposées sous forme de service et consiste à fournir l'accès à des machines virtuelles sur lesquelles le client devra installer le système d'exploitation et application.
- **La couche Plateforme en tant que Service (PaaS)** : composée des services de l'IaaS, et par-dessus lequel s'ajoute un environnement de développement d'applications, dans ce cas-là, le client est déchargé de la gestion du système d'exploitation et de l'infrastructure. Le client peut contrôler et ajouter des outils.
- **La couche Logiciel en tant que Service (SaaS)** : les fournisseurs de SaaS mettent, au niveau de cette couche, des applications à la disposition des utilisateurs. Le client n'a pas à se soucier de l'infrastructure sous-jacente. Généralement le service est accessible via internet mais peut être installé localement sur un PC.

## 1.1.3. Les principaux fournisseurs de services Cloud

De nos jours, de nombreuses entreprises proposent des services cloud et fonctionnent mieux de jour en jour. Ces entreprises de fournisseurs de services cloud fournissent la mise en réseau des serveurs de bases de données de stockage et les logiciels grâce auxquels une entreprise peut se développer.

Parmi ces entreprises fournisseurs de services, nous avons Amazon Web Service (AWS), Microsoft Azure et Google Cloud Platform qui sont les meilleurs fournisseurs de services cloud du marché, voir figure ci-après :



**Figure 5: Principaux fournisseurs services Cloud**

- **Amazon Web Services** : c'est une plate-forme de cloud computing qui fournit des services tels que la puissance de calcul, le stockage de base de données, la livraison de contenu et de nombreuses autres fonctions qui aideront à intégrer une entreprise. Les services Web Amazon sont flexibles, évolutifs et fiables et, pour cette raison, de nombreuses entreprises les mettent en œuvre dans leur travail. Il n'y a aucun coût initial et le client ne doit payer que ce qu'il a utilisé. C'est l'un des principaux fournisseurs de services cloud parmi tous. Avec l'aide d'Internet, le client peut accéder à un stockage hautement durable tel qu'Amazon Glacier, Amazon S3 et Amazon EBS. Il dispose également d'une base de données hautes performances telle qu'Amazon Redshift, Amazon Dynamo DB, Amazon ElastiCache et **Amazon RDS**.
- **Microsoft Azure** : il est utilisé pour créer des tests, déployer et gérer l'application. Ce processus est effectué dans un réseau mondial du centre de données géré par Microsoft. Il s'agit d'une plateforme de cloud privé et public. Il utilise la virtualisation qui différencie le couplage entre le système d'exploitation et le CPU à l'aide d'une couche d'abstraction appelée hyperviseur. Cet hyperviseur émule toutes les fonctionnalités de la machine physique telles que le matériel et le serveur en une machine virtuelle. Il existe de nombreuses machines virtuelles disponibles et chaque machine virtuelle peut exécuter de nombreux systèmes d'exploitation. Dans le centre de données de Microsoft, il existe de nombreux serveurs et chaque serveur est constitué d'un hyperviseur à travers lequel plusieurs machines virtuelles peuvent fonctionner. Avec l'aide d'Azure, il est facile pour les développeurs et les professionnels de l'informatique de gérer et de déployer leurs applications et services.
- **La plate-forme cloud Google** : c'est l'un des principaux services de cloud computing proposés par Google et fonctionne sur la même infrastructure que celle utilisée par Google pour ses produits destinés aux utilisateurs finaux. La plate-forme cloud Google est essentiellement utilisée pour la recherche Google et YouTube.

Il existe différents services proposés par Google Cloud, tels que l'analyse de données, l'apprentissage automatique et le stockage de données.

Les données stockées dans Google Cloud sont sécurisées et peuvent y accéder facilement. Il offre une variété de services allant de l'infrastructure en tant que service à la plateforme en tant que service. Google s'engage également en matière de sécurité et de stabilité. Avec l'aide de la plate-forme cloud de Google, l'utilisateur est libre de réfléchir au code et aux fonctionnalités nécessaires pour se développer sans se soucier du côté des opérations. Ici, la plupart des services sont entièrement gérés et les détails sont assez faciles pour que le client se concentre sur son travail. En cela, l'apprentissage automatique et l'utilisation de l'API sont très simples.

## 1.2. Le Cloud Computing Mobile (MCC)

Les appareils mobiles deviennent de plus en plus un élément essentiel de la vie humaine en tant qu'outils de communication les plus efficaces et les plus pratiques non limités dans l'espace et par le temps. A travers ces appareils mobiles le Cloud Computing Mobile peut répondre à toutes les attentes des utilisateurs.

Ainsi ces derniers pourront accumuler une riche expérience de différents services à partir d'applications mobiles qui s'exécutent sur des appareils ou serveurs distants à travers des réseaux sans fil.

Le Cloud Computing Mobile est introduit comme une intégration du Cloud Computing dans l'environnement mobile. Il accorde de nouveaux types de services aux utilisateurs.

### 1.2.1. Qu'est-ce que le Cloud Computing Mobile

Le Cloud Computing Mobile se réfère à une infrastructure dans laquelle le stockage et le traitement des données se déroulent en dehors de l'appareil mobile [2]

Il permet d'accéder à des services qui sont fournis par l'environnement du Cloud Computing à tout moment et n'importe où, à partir d'appareils mobiles.

Nous avons les applications de l'entreprise qui sont accessibles via des appareils mobiles libérant ainsi à l'utilisateur des contraintes de temps et d'espace. Ces appareils mobiles n'ont pas besoin d'une puissante configuration (par exemple : la vitesse du processeur et la capacité de mémoire) puisque tous les modules complexes de calcul peuvent être traités dans les nuages [2]

## 1.2.2. Architecture du Cloud Computing Mobile

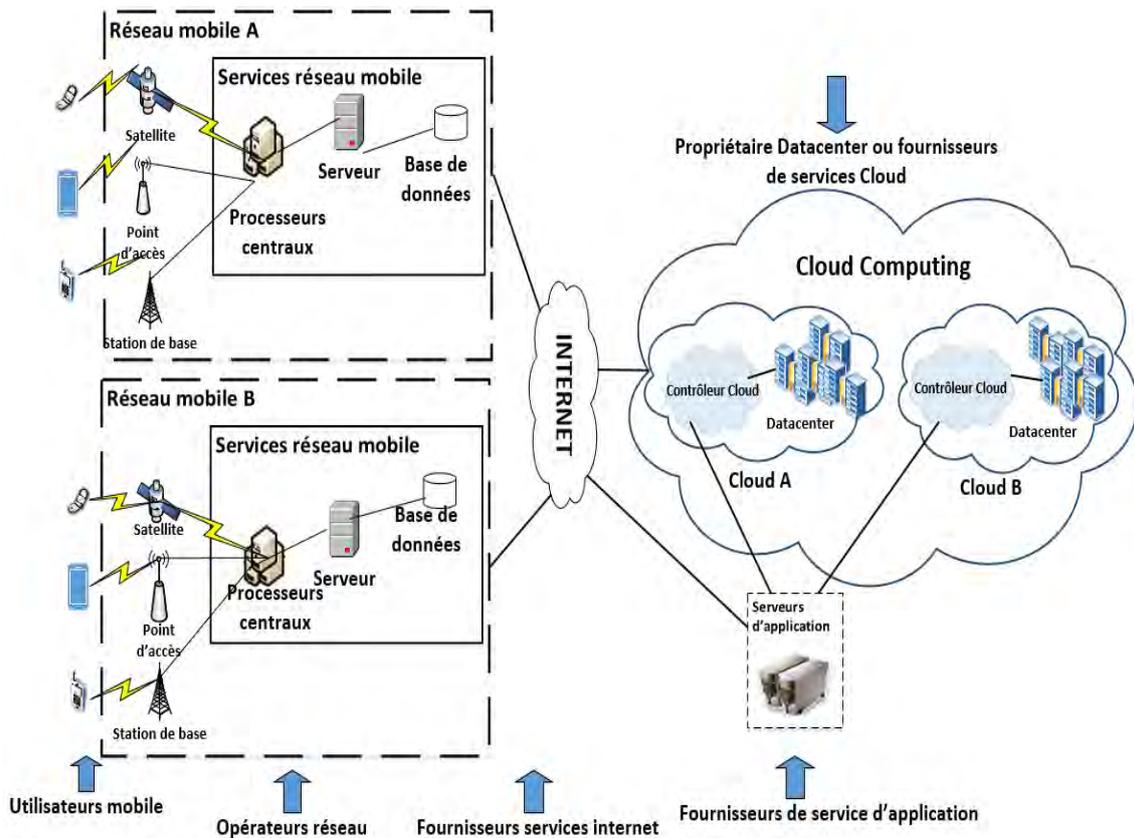


Figure 6: Architecture du Cloud Computing Mobile [2]

Dans cette figure, les appareils mobiles sont connectés aux réseaux mobiles via des stations de base (par exemple, station émettrice-réceptrice de base, point d'accès ou satellite) qui établissent et contrôlent les connexions (liaisons aériennes) et les interfaces fonctionnelles entre les réseaux et appareils mobiles. Les demandes et informations des utilisateurs mobiles (par exemple ; ID et emplacement) sont transmises aux processeurs centraux qui sont connectés aux serveurs fournissant des services de réseau mobile. Ici, les opérateurs de réseau mobile peuvent fournir des services aux utilisateurs mobiles en tant qu'authentification, autorisation et compatibilité en fonction de l'agent domestique et les données des abonnés stockés dans des bases de données.

Après cela, les demandes des abonnés sont transmises à un cloud via internet. Dans le cloud, les contrôleurs cloud traitent les demandes pour fournir aux utilisateurs mobiles les services cloud correspondants.

## 1.2.3. Les applications mobiles du MCC

Les applications mobiles gagnent de plus en plus de parts de marché sur le marché mondial de la téléphonie mobile.

Les applications en nuage mobiles déplacent la puissance de calcul et le stockage de données des téléphones mobiles vers le nuage, apportant des applications et des contrôleurs de périphérique non seulement aux utilisateurs de smartphones, mais également à un plus grand nombre d'abonnés mobiles [2].

Diverses applications mobiles ont profité des avantages du MCC. Nous pouvons citer quelques applications comme :

- **M-santé (santé mobile) :** le but de l'application du MCC dans les applications médicales est de minimiser les limites du traitement médical traditionnel (par exemple, petit stockage physique, sécurité et confidentialité, et erreurs médicales). Les soins de santé mobiles (m - healthcare) fournissent aux utilisateurs mobiles des aides pratiques pour accéder facilement et efficacement aux ressources (par exemple, les dossiers de santé des patients) [2].
- **M-commerce (commerce mobile) :** le commerce mobile (m - commerce) est un modèle commercial pour le commerce utilisant des appareils mobiles. Les applications m-commerce remplissent généralement certaines tâches qui nécessitent une mobilité. Par exemple, les transactions et les paiements mobiles, la messagerie mobile et la billetterie mobile [2].
- **M-Learning (l'apprentissage mobile) :** l'apprentissage mobile (m-Learning) est conçu sur la base de l'apprentissage électronique (e-learning) et de la mobilité. Cependant, les applications traditionnelles de m-Learning ont des limites en termes de coût élevé des appareils et du réseau, un faible débit de transmission du réseau et des ressources pédagogiques limitées. Des applications de m-Learning basées sur le cloud sont introduites pour résoudre ces limitations. Par exemple, en utilisant un cloud avec une grande capacité de stockage et une capacité de traitement puissante, les applications fournissent aux apprenants des services beaucoup plus riches en termes de taille de données (informations), de vitesse de traitement plus rapide et de durée de vie de la batterie plus longue [2].

### 1.3. La virtualisation

La virtualisation est l'ensemble de technique matérielles et logicielles qui permettent de faire fonctionner sur une seule machine plusieurs systèmes d'exploitation ou plusieurs applications, séparément les uns des autres, comme s'ils fonctionnaient sur des machines physiques distinctes [3].

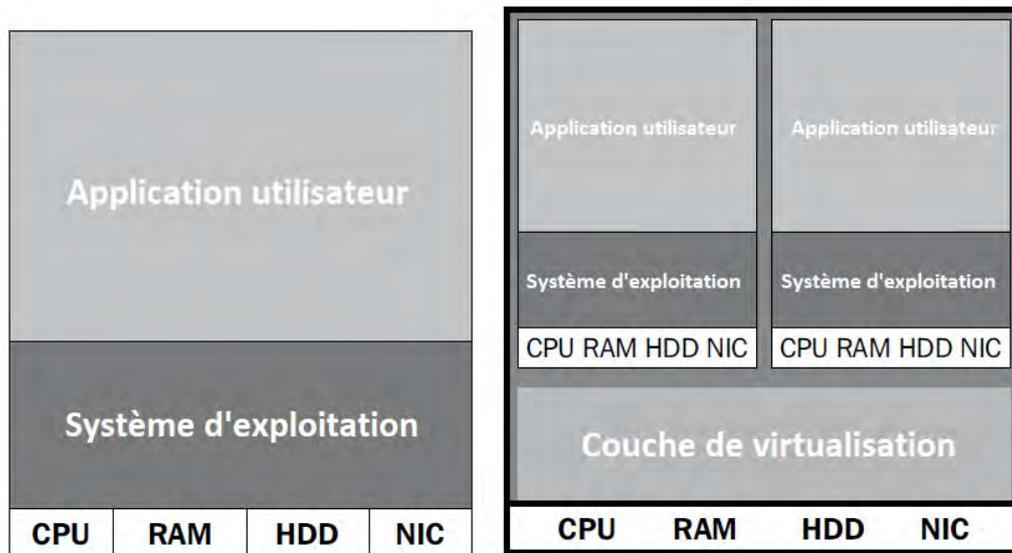


Figure 7: Serveur virtualisé

En d'autres termes, la virtualisation est une technique qui permet de partager une seule instance physique d'une ressource ou d'une application entre plusieurs clients et organisations.

Le Cloud Computing s'appuie sur la technologie de virtualisation pour atteindre le but de fournir des ressources informatiques selon l'utilité de façon dynamique et de nombreux services tels que l'infrastructure, les logiciels et les plates-formes.

### 1.3.1. Types de virtualisations

Il existe quatre types de virtualisations : la virtualisation de serveur, la virtualisation d'application, la virtualisation de postes de travaux et la virtualisation de stockage.

- **La virtualisation d'application** : elle permet de séparer complètement l'application du système d'exploitation hôte et des autres applications présentes afin d'éviter les conflits. En outre elle peut être définie comme la technologie qui permet de séparer l'environnement du bureau et des applications associées de la machine physique [12].
- **La virtualisation des postes de travail** : la virtualisation des postes de travail permet aux administrateurs systèmes et réseaux de gérer beaucoup plus facilement les postes de travail de l'entreprise et de répondre avec flexibilité aux demandes des utilisateurs. Le poste de travail virtualisé ou bureau virtuel peut se trouver sur l'ordinateur du client ou sur un serveur du centre de données [12].

- **La virtualisation de stockage** : un des plus grands défis de la virtualisation reste le stockage. La virtualisation du stockage consiste à regrouper le stockage physique de plusieurs périphériques de stockage réseau afin qu'il ressemble à un seul périphérique de stockage [12].  
La virtualisation de stockage permet d'exploiter au maximum les ressources, d'exploiter au mieux le stockage des disques durs.
- **La virtualisation de serveur** : elle permet de regrouper plusieurs serveurs physiques sous-employés sur un seul hôte qui exécute des systèmes virtuels. Il permet aussi de réduire la consommation électrique et le nombre d'administrateurs. Il participe beaucoup à la réalisation des économies (consommation électrique) [12].

### 1.3.2. Les Techniques de Virtualisation

Nous avons quelques techniques de virtualisation : le cloisonnement ou l'isolation, la paravirtualisation, la virtualisation complète, la virtualisation assistée par le matériel

- **Le cloisonnement ou l'isolation** vise à isoler chaque processus dans un conteneur. Ce conteneur va s'exécuter sous Linux de manière native et va **partager le noyau** de la machine hôte avec d'autres conteneurs ; ne prenant pas plus de mémoire que tout autre exécutable, ce qui le rend **léger**.

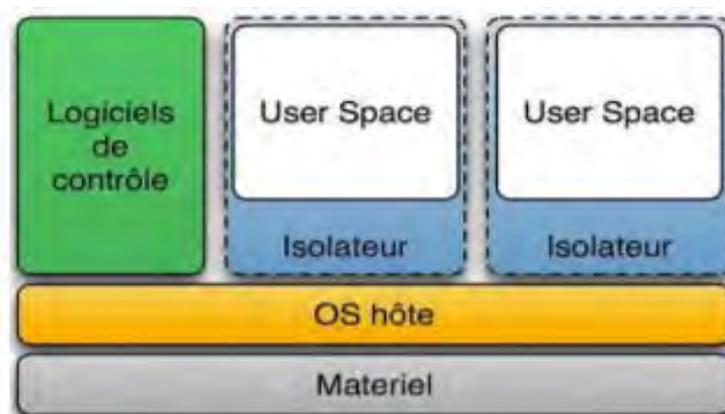


Figure 8: Isolation [13]

- **La paravirtualisation** présente aux systèmes d'exploitation une machine générique spéciale, qui requiert donc des interfaces spéciales, intégrées aux systèmes invités sous la forme de drivers ou de modifications du noyau. Il s'agit d'un compromis entre un niveau d'abstraction élevé et un niveau de performance satisfaisant [13].

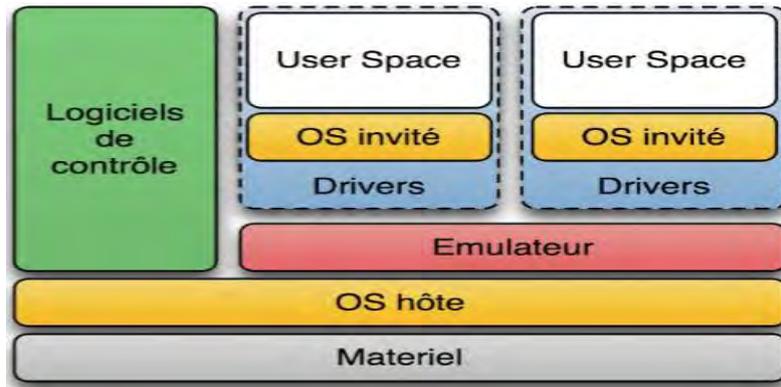


Figure 9: La paravirtualisation [13]

- Dans la **virtualisation complète**, l'hyperviseur intercepte de manière transparente tous les appels que le système d'exploitation peut faire aux ressources matérielles, et supporte donc des systèmes invités non-modifiés [13].

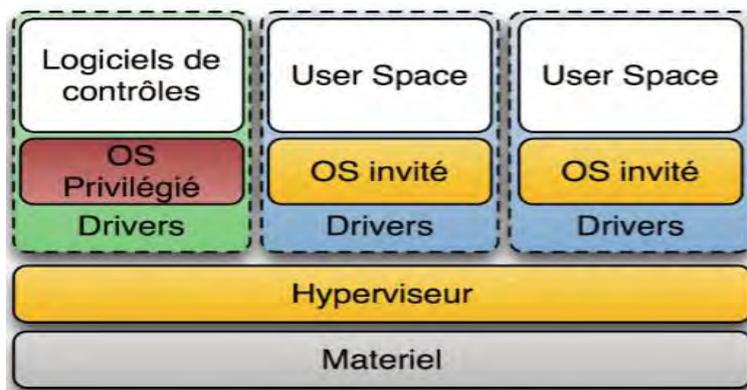


Figure 10 : La virtualisation complète [14]

- La **virtualisation assistée par le matériel** a pour but de faire fonctionner des systèmes invités dont les OS sont non-modifiés et peuvent être différents. La virtualisation assistée par matérielle est donc une évolution de la virtualisation totale puisqu'elle émule toujours le matériel nécessaire au bon fonctionnement des systèmes invités, mais avec une perte de performances moindre.

### 1.3.3. Quelques solutions de virtualisation

Dans cette partie nous allons étudier quelques solutions de virtualisation comme Hyper-V, VMWare vSphere Hypervisor, Citrix XenServer et Proxmox VE :

- **Hyper-V** : est un système de virtualisation basé sur un hyperviseur 64 bits de la version de Windows Server 2008 également connu sous le nom de Windows Server Virtualisation.

Il permet à un serveur physique de devenir Hyperviseur et ainsi manager et accueillir des machines virtuelles appelées communément VM. Avec cette technologie il est possible d'exécuter plusieurs systèmes d'exploitation virtuellement sur une même machine physique et ainsi séparer ces systèmes d'exploitation les uns des autres. Pour différentes VM, les ressources de l'hyperviseur sont alors mutualisées, ce qui présente économiquement un intérêt car auparavant il fallait une machine physique pour chaque serveur. Sur Windows 7, il est possible d'utiliser la console Hyper-V. Dans le sens inverse, plusieurs systèmes d'exploitation peuvent tourner à l'intérieur d'Hyper-V [15].

- **VMWare vSphere Hypervisor** : C'est un hyperviseur bare-metal de VMware qui permet aux utilisateurs de virtualiser leurs serveurs et de consolider leurs applications. L'un des avantages de vSphere Hypervisor 6.0 réside dans sa légèreté : l'hyperviseur n'occupe quelques centaines de Mo d'espace disque. vSphere Hypervisor 6 est toutefois et c'est logique moins complet que vSphere. Ainsi, on doit se contenter d'une console d'administration minimaliste en lieu et place d'un véritable environnement. Par ailleurs, tous les agents VMware sont exécutés directement sur le VMkernel. La version 6 de vSphere améliore un certain nombre de caractéristiques de l'hyperviseur. Il est par exemple possible de migrer des VMs à chaud entre deux Distributed Switch, entre deux switches ou d'un vSwitch vers un Distributed Switch (mais pas l'inverse) grâce à Cross vSwitch vMotion. Une migration d'une VM d'un vCenter à un autre est possible, même s'ils ne partagent aucun Datastores. Il est en outre possible de mettre en place du Fault Tolerance avec un maximum de 4 vCPU, contre 1 seul vCPU auparavant. Enfin, vSphere Hypervisor 6 apporte le VM Hardware en version 11 et VSAN version 6.0 [16].
- **Citrix XenServer** : est une plate-forme de gestion de virtualisation optimisée pour les infrastructures de virtualisation d'applications, de postes de travail et de serveurs. La consolidation et le confinement des charges de travail sur l'hyperviseur Citrix permettent aux entreprises de gérer différents types de charges de travail, des systèmes d'exploitation mixtes et des configurations de stockage ou de mise en réseau. Il s'agit d'un hyperviseur natif basé sur GNU / Linux développé par le projet Xen et publié sous GNU AGPL v2 en open source. Pour XenServer, un concept de domaine est utilisé pour toutes les machines virtuelles, et le domaine le plus privilégié (par exemple, un domaine qui permet un accès direct au matériel) utilisé par l'hyperviseur pour gérer d'autres machines virtuelles. Il prend en charge la paravirtualisation, qui permet à un utilisateur d'exécuter des invités virtualisés sur un processeur sans prise en charge de la virtualisation ; par exemple, aucun Intel VT-x ou AMD-V n'est nécessaire. Amazon Web Service (AWS) est un exemple de production d'utilisation de XenServer [4].
- **Proxmox Virtual Environnement (PVE)** : est un hyperviseur open source basé sur GNU / Linux (basé sur Debian) avec un noyau basé sur RHEL et publié sous GNU

AGPL v3. Il diffère du logiciel de virtualisation alternatif car Proxmox fournit une gestion sans autre installation. Les technologies sous-jacentes utilisées sont Open Virtuozzo (OpenVZ) et Kernel-based Virtual Machine (KVM). Des plans d'abonnement sont disponibles pour accéder au référentiel d'entreprise, aux mises à jour logicielles et au support utilisateur [4]. Proxmox est une solution de virtualisation de type "bare metal". Deux types de virtualisation sont supportés par Proxmox VE : une virtualisation matérielle et complète via l'hyperviseur (qui permet de virtualiser des systèmes Solaris, Linux ou Windows), qui demande toutefois un (ou plusieurs) processeurs, et une isolation grâce à des conteneurs LXC. Dans ce dernier cas de figure, seuls des systèmes d'exploitation de type Linux peuvent être virtualisés. Il n'y a qu'une seule version libre et gratuite de la distribution, la société Proxmox Server Solutions proposant du support payant pour ce qui en auraient besoin.

Les fonctionnalités sont impressionnantes :

- Une interface en ligne de commande et web (disponible en français) permettant de gérer les machines virtuelles tant au format KVM que des conteneurs OpenVZ.
- Une belle intégration de la couche cluster de linux, permettant de reproduire les fonctionnalités comme HA (High Availability à la VMWare), et la migration à chaud.
- La prise en charge simplifiée des sauvegardes et des restaurations.
- Les snapshots.

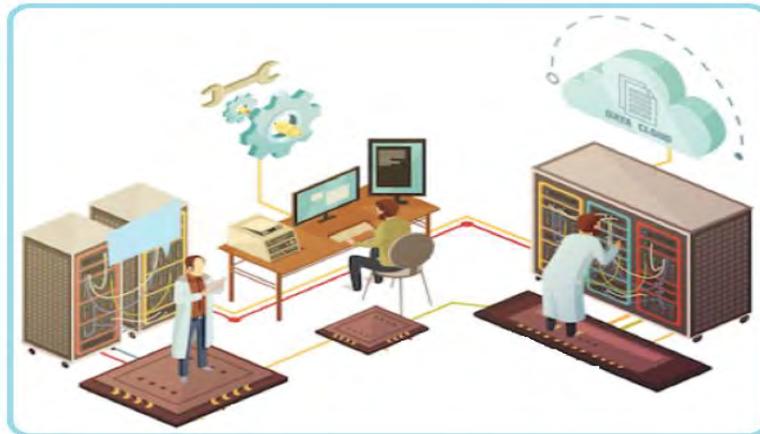
Après une étude des différentes solutions nous allons faire une étude comparative de ces dernières.

**Tableau 1: Etude comparative des solutions de virtualisations**

Solutions / Critères	VMWare ESXi	Windows Hyper-V	Citrix XenServer	Proxmox VE
Open source	NON	NON	OUI	OUI
SE hôte	RHEL 5	Windows	CentOS	Debian
SE invités	Linux, Windows	Linux (limité), Windows	Linux (limité), Windows	Linux, Windows et MacOS
Techniques de virtualisation	<ul style="list-style-type: none"> <li>• Complète</li> <li>• Paravirtualisation</li> </ul>	<ul style="list-style-type: none"> <li>• Paravirtualisation</li> </ul>	<ul style="list-style-type: none"> <li>• Paravirtualisation</li> <li>• Complète</li> </ul>	<ul style="list-style-type: none"> <li>• Complète</li> <li>• Isolation</li> </ul>
HA Clustering	OUI	OUI	OUI	OUI
Haute disponibilité	OUI	NON	OUI	OUI
Interface web native	OUI	NON	NON	OUI

## 1.4. Cloud Computing et système d'information

La plupart des ressources informatiques (IT) d'un système d'information (SI) détenues par une organisation résident dans les locaux de l'organisation. L'organisation doit gérer ces ressources pour s'assurer qu'elles peuvent être utilisées pour aider l'organisation à atteindre ses objectifs : voir figure



**Figure 11: Système d'Information interne**

Dans ce cas l'organisation engage tous les coûts d'investissements, d'exploitation et d'entretien. Etant donné que certaines organisations sont souvent confrontées à des conditions économiques difficiles, des concepts tels que l'externalisation, la gestion du changement et la réduction des coûts reçoivent de plus en plus d'attention.

En effet, ces concepts visent tous à économiser sur les budgets et à faire face à des changements inattendus. La technologie Cloud Computing promet de transformer l'informatique en une source d'économie de flexibilité et d'agilité pour l'entreprise.

Avec un système d'information dans le Cloud Computing, l'organisation n'a pas besoin de posséder la plupart de ses ressources informatiques. Elle utilisera les services offerts par un fournisseur et bien sûr payer pour l'utilisation des ressources avec un mode de paiement à l'utilisation.



**Figure 12: Système d'Information externalisé**

Ainsi l'organisation va gagner en matière de [5] :

- **Simplicité et flexibilité** : contrairement au modèle informatique traditionnel, les utilisateurs peuvent augmenter et réduire l'utilisation selon leurs besoins. Les fournisseurs de cloud computing mettent généralement à niveau leurs systèmes pour améliorer les services et les utilisateurs doivent bénéficier de la technologie toujours à jour, à la fois matérielle et logicielle.
- **Agilité organisationnelle** : l'agilité organisationnelle est la capacité d'une organisation à s'adapter rapidement et efficacement à un environnement changeant, comme une réponse rapide et efficace aux préférences des clients tout en réalisant une adaptation stratégique (en s'adaptant aux tendances et aux problèmes) à moyen et long terme mise en forme.
- **Réduction des coûts** : la réduction des coûts des dépenses informatiques est très attendue car le cloud computing peut éliminer de nombreux coûts informatiques conventionnels tels que les coûts d'investissement des licences matérielles et logicielles ainsi que les coûts de maintenance. Comme les coûts d'investissement et de maintenance peuvent être supprimés des dépenses informatiques, les dépenses informatiques globales dans le cloud computing sont moins chères que les coûts conventionnels.
- **Evolutivité** : dans le système de cloud computing, du nouveau matériel et de nouveaux nœuds peuvent être ajoutés facilement avec des modifications limitées de la configuration de l'infrastructure et des logiciels, car l'architecture cloud est conçue pour s'adapter aux expansions horizontales et verticales. En outre, l'échelle du service peut être étendue à la demande. Par conséquent, les utilisateurs peuvent commencer avec des services de petite taille et relativement peu coûteux, puis étendre les services au besoin ultérieurement.
- **Mise à niveau et la migration** : la mise à niveau et la migration vers les nouvelles technologies du cloud computing sont transparentes et l'interruption des services n'est pas nécessaire. Les fournisseurs de cloud computing maintiennent normalement leur système à jour et la mise à jour du système n'aura aucune incidence sur les services.
- **Fiabilité et disponibilité** : pour garantir la fiabilité et la disponibilité, les systèmes informatiques conventionnels utilisent des centres de données redondants. Un fournisseur de services de cloud computing assurera certainement une fiabilité et une disponibilité élevée de ses services car cela est à la base de sa crédibilité. Dans les systèmes de cloud computing, la maintenance ou les mises à niveau de serveur n'affecteront pas les services car ils sont traités par d'autres serveurs de manière transparente.

## **Partie 2 : Etat de l'art sur les plateformes de formation à distance**

### **2.1. Introduction**

Un système de gestion de l'apprentissage est une application logicielle Web conçue pour gérer le contenu d'apprentissage, l'interaction des étudiants, les outils d'évaluation et les rapports sur les progrès de l'apprentissage et les activités des étudiants.

Le contenu d'apprentissage en ligne est accessible via une plateforme LMS, qui permet aux étudiants de voir et d'interagir avec des outils d'apprentissage via des navigateurs Web utilisant n'importe quel système d'exploitation, ordinateur ou appareil mobile.

Les systèmes de gestion de l'apprentissage sont également des plates-formes qui comprennent des systèmes d'apprentissage, des systèmes de gestion de cours, des systèmes de gestion de contenu, des portails et des systèmes de gestion pédagogique. Les systèmes de gestion de l'apprentissage représentent une évolution à partir des processus et des systèmes développés par certains établissements pour inscrire les étudiants à des cours spécifiques et conserver des registres des activités des étudiants.

Divers choix d'apprentissage ont été développés pour permettre aux étudiants de suivre des cours en ligne, parfois dans le cadre du programme officiel et parfois en raison de la nécessité d'une certification institutionnelle. Une plateforme LMS peut également aider les étudiants à accéder aux informations d'apprentissage via des directives de cours, en téléchargeant des devoirs et en téléchargeant des notes, des interactions actives entre étudiants et enseignants, des interactions entre étudiants, des interactions entre étudiants et des outils d'apprentissage, en partageant des connaissances et en passant des examens en ligne.

Il existe deux types de plateformes LMS à savoir open source et commercial. Cependant, les plates-formes open source deviennent un choix pour toute institution, car ils sont avantageux pour les utilisateurs en permettant aux plates-formes d'être modifiées en fonction des besoins des utilisateurs, et en raison des faibles coûts facturés pour obtenir un meilleur service, par rapport à une plateforme commerciale qui nécessite une licence par utilisateur, avec un abonnement supplémentaire et des frais de maintenance pour vous assurer que votre LMS est à jour.

### **2.2. Quelques plateformes de formation à distance**

Dans cette partie nous allons présenter quelques plateformes e-learning open source comme Sakai, ATutor et Moodle.

- **Sakai** : la plate-forme Sakai a été développée à des fins communautaires et LMS, et le logiciel a été conçu pour soutenir l'apprentissage, l'enseignement, la recherche et la coopération. Le logiciel a été développé en 2004 et la dernière version est Sakai 10.5, qui a été développée en 2014. Sakai a été développé sous les termes d'une licence open source. Les caractéristiques de la plateforme Sakai sont la convivialité, la facilité d'utilisation, la flexibilité et la compatibilité avec les outils Web 2.0. Les outils de Sakai sont renforcés par un outil de conception pédagogique et un e-portfolio. De plus, ce logiciel a été traduit dans plus de 20 langues, permet une interaction synchrone et asynchrone, et offre un espace personnel pour la rédaction de brouillons et la journalisation ainsi que la gestion des informations personnelles et privées. Les utilisateurs peuvent également envoyer et recevoir des messages personnels d'autres utilisateurs [6].
  
- **ATutor** : la plate-forme ATutor est un LMS basé sur une technologie Web open source utilisée pour développer et présenter des cours en ligne. Le leader peut copier, distribuer et éditer ATutor sous les conditions de licence publique d'une licence publique générale GNU (GPL). La principale caractéristique de ATutor est son accessibilité, car elle garantit que les utilisateurs peuvent participer à des activités en tant qu'étudiants, instructeurs et administrateurs. De plus, l'utilisation d'IMS / ISO pour le support permet aux étudiants d'adapter l'environnement et le contenu aux besoins de leurs cours. Les instructeurs et les étudiants peuvent gérer leurs cours ATutor, et les utilisateurs peuvent envoyer et recevoir des messages personnels d'autres utilisateurs. Chaque utilisateur dispose de son propre utilitaire de stockage de fichiers, et ce stockage peut être partagé avec d'autres utilisateurs. Ainsi, tout le contenu et la structure du cours peut être stockée et sauvegardée dans ATutor [6].
  
- **Moodle** : la première plate-forme open source à considérer est Moodle. Moodle est une abréviation de Modular Object Oriented Dynamic Learning Environment. C'est un LMS qui a été développé en 2001 et la dernière version est Moodle 3.9, développée le 13 juillet 2020. Il est basé sur le principe pédagogique social constructionniste. Il est développé sous les termes d'une licence publique générale GNU (GPL), ce qui signifie que toute modification peut être apportée au code source tant que la licence d'origine n'est pas modifiée. Les caractéristiques de la plate-forme Moodle sont la convivialité, l'accessibilité et la flexibilité. Il existe également une liste d'étudiants sur chaque cours permettant au conférencier de savoir la dernière fois que chaque étudiant a accédé à la plateforme, la possibilité d'intégrer Moodle dans d'autres systèmes, la possibilité de permettre une interaction synchrone et asynchrone, un espace personnel pour la rédaction de brouillons et la journalisation ainsi que la gestion des informations personnelles et privées, et des contenus qui se développent en fonction des besoins d'enseignement et d'apprentissage et qui peuvent être réutilisés [6].

Après une étude de ces plateformes nous allons maintenant faire une étude comparative entre elles :

**Tableau 2: Etude comparative de ces plateformes d'e-learning**

Critères	Plateformes LMS		
	Sakai	ATutor	Moodle
Version	20.0	2.2.4	3.9
Flexible	OUI	OUI	OUI
Accessible	OUI	OUI	OUI
Facile à utiliser	OUI	OUI	OUI
Interaction synchrone	NON	NON	OUI
Interaction asynchrone	OUI	NON	OUI
Intégration avec d'autres services	OUI	NON	OUI

## Partie3 : Choix technologique

Après une étude détaillée sur des solutions de virtualisation et sur quelques plateformes de formation à distance, nous avons choisi :

- La solution de virtualisation **Proxmox VE** du fait qu'elle est open source, stable, simple à utiliser et administrer grâce à une interface graphique de type web etc...
- La plateforme de formation à distance **Moodle** car elle est open source, accessible à tous, permet un apprentissage facile, des interactions synchrones etc... ; elle sera intégrée avec **BigBluButton** qui est un système de conférence web open source pour l'apprentissage en ligne. Il permet le partage en temps réel de l'audio, de la vidéo, de la présentation et de l'écran, ainsi que des outils de collaboration tels que le tableau blanc, les notes partagées, les sondages et les ateliers. BigBluButton peut également enregistrer les sessions pour une lecture ultérieure.

**CHAPITRE III : PROPOSITION D'UNE  
ARCHITECTURE HAUTEMENT DISPONIBLE  
POUR UN DISPOSITIF D'E-LEARNING**

## III.1. Rappels sur la gestion de la haute disponibilité des systèmes dans le Cloud Computing

La haute disponibilité (HA) permet d'assurer et de garantir le bon fonctionnement en continu, sans défaillance des services ou applications et ce 7j/7 et 24h/24.

De nombreux services peuvent être utilisés pour assurer la disponibilité d'un service tels que : l'équilibrage de charge (Load Balancing), la tolérance aux pannes (Failover), la réplication des données, etc. pour limiter l'indisponibilité d'un SI.

### III.1.1. La tolérance aux pannes

La tolérance aux pannes est la capacité du système à résister aux changements brusques qui se créent en raison de pannes matérielles, logicielles, congestions du réseau, etc.

La gestion des pannes dépend de deux paramètres importants que sont : le RPO (Recovery Point Objective) qui définit la quantité de données à perdre lors d'un sinistre et le RTO (Recovery Time Objective) qui détermine le temps d'arrêt minimum pour la récupération après un sinistre.

Il existe deux stratégies standards de tolérance aux pannes, à savoir la stratégie proactive et la stratégie réactive.

- **La politique de tolérance aux pannes proactive [7]** : le principe de cette politique est d'éviter les pannes en prenant de manière proactive des mesures préventives et remplacer les composants susceptibles de tomber en panne. Les techniques basées sur la politique proactive sont :
  - **La migration préventive** : qui utilise un système de contrôle en boucle de rétroaction où les applications sont constamment surveillées et analysées.
  - **Rajeunissement du logiciel** : avec cette technique des redémarrages périodiques sont programmés pour le système et après chaque redémarrage le système reprend avec un état neuf.
- **La politique de tolérance aux pannes réactive [7]** : cette politique est utilisée pour réduire l'effet des pannes qui sont déjà survenues. Sur la base de la politique de tolérance aux pannes réactive deux techniques sont utilisées :
  - **Checkpointing (point de contrôle) / Restart (redémarrage)** : les tâches échouées sont redémarrées à partir du point de contrôle avant le point de défaillance plutôt que le point de départ. Cette technique est efficace pour les applications de grandes envergures dans le Cloud.
  - **Réplication** : cette technique de réplication consiste à conserver plusieurs copies de données. Différents réplicas vont s'exécuter en utilisant des ressources différentes jusqu'à ce que la tâche soit terminée.

### III.1.2. La réplication des données

La réplication est l'élément clé pour augmenter l'accessibilité, la tolérance aux pannes et la disponibilité dans tous les types de systèmes distribués.

Il existe deux types de répliquions : la réplication synchrone et la réplication asynchrone

- **La réplication synchrone** : elle garantit que lorsqu'une transaction met à jour une réplique primaire, toutes ses répliques secondaires sont mises à jour dans la même transaction. L'avantage de la mise à jour synchrone est de garder toutes les données au même niveau de mise à jour. Alors le système peut garantir la fourniture de la dernière version des données quel que soit la réplique accédée [8].
- **La réplication asynchrone** : c'est le mode de distribution dans lequel certaines sous-opérations locales effectuée suite à une mise à jour globale sont accomplies dans des transactions indépendantes en temps différé [8].

Elle est basée sur deux approches :

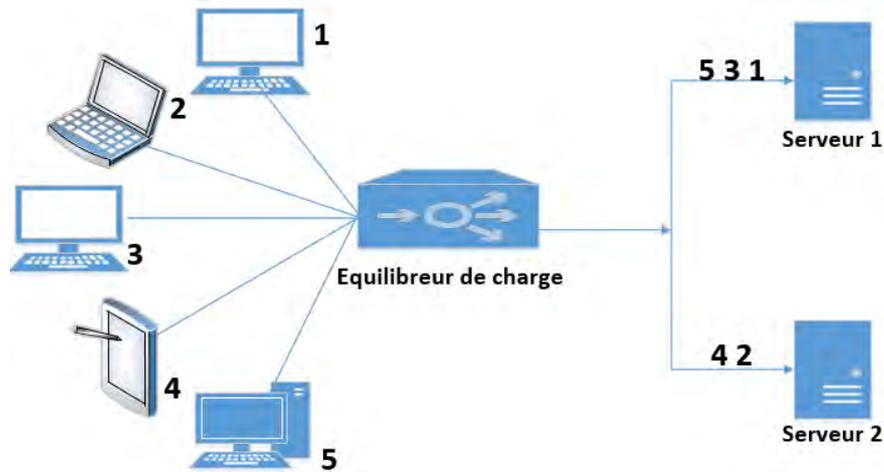
- **Approche basée sur l'état** : la réplique source est mise à jour ensuite le sous-système transmet l'état sur les répliques par la fusion de l'état livré à l'état local
- **Approche basée sur les opérations** : le sous-système envoie l'opération de mise à jour et ses paramètres à toutes les répliques.

### III.1.3. L'équilibrage de charge (Load Balancing)

L'équilibrage de charge est une partie importante qui garantit que tous les appareils ou processus effectuent la même quantité de travail dans le même temps. Avec le partage de charge, chaque nœud peut fonctionner et les données peuvent aussi être reçues et envoyées sans délai [9].

Dans le but de rendre les ressources accessibles aux utilisateurs finaux avec facilité et commodité, différents algorithmes pour l'équilibrage de charge ont été développés tels que : Round Robin, Weighted Round Robin, Least Connections, Weighted Least Connections, Random etc.

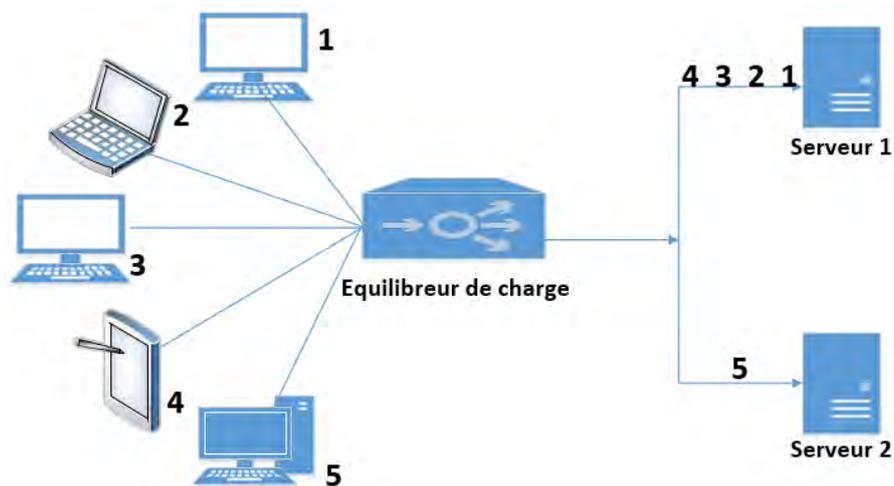
- **Round Robin** : c'est un algorithme qui est facile à mettre en œuvre. Par exemple soit deux serveurs en attente de requêtes derrière notre équilibreur de charge. Une fois que la première demande arrive, l'équilibreur transmettra cette demande dans le premier serveur. Lorsque la demande arrive (peut-être d'un autre client), cette demande sera transmise au deuxième serveur. Comme, le deuxième serveur est le dernier de ce cluster, la troisième requête sera renvoyée au premier serveur, la quatrième au deuxième serveur et ainsi de suite de manière cyclique.



**Figure 13: L'algorithme Round Robin**

L'algorithme Round Robin est le meilleur pour les clusters composés de serveurs avec les mêmes spécifications. Dans d'autres situations nous pouvons observer d'autres algorithmes, comme ceux-ci-dessous.

- **Weighted Round Robin** : cet algorithme est similaire au Round Robin car la façon dont les demandes sont attribuées aux nœuds est cyclique bien qu'elle soit tordue. Le nœud avec les spécifications les plus élevées aura un plus grand nombre de demandes. Généralement, on spécifie des poids proportionnellement aux capacités réelles. De ce fait si la capacité du serveur 1 est 4 fois plus grande à celle du serveur 2, on peut lui donner un poids de 4 et le serveur 2 un poids de 1.



**Figure 14: L'algorithme Weighted Round Robin**

De la sorte, lorsque les clients commencent à arriver, les 4 premiers seront affectés au serveur 1 et le 5<sup>ème</sup> client au serveur 2. Si plus de clients arrivent, la même séquence sera suivie. Autrement dit le 6<sup>ème</sup>, 7<sup>ème</sup>, 8<sup>ème</sup> et le 9<sup>ème</sup> client iront dans le serveur 1 et le 10<sup>ème</sup> dans le serveur 2, ainsi de suite.

- **Least Connections** : cet algorithme prend en compte l'état (nombre de connexions) actuel de chaque serveur. Lorsqu'un client tente de se connecter, l'équilibreur essaie de déterminer quel serveur a le moins de connexion, puis attribue la nouvelle connexion à ce serveur.

Alors, si le 5<sup>ème</sup> client tente de se connecter après que le 1<sup>ère</sup> et le 3<sup>ème</sup> client soient déjà déconnectés mais que le 2<sup>ème</sup> et le 4<sup>ème</sup> soient toujours connectés, l'équilibreur de charge attribuera le 5<sup>ème</sup> client au serveur 1 au lieu du serveur 2.

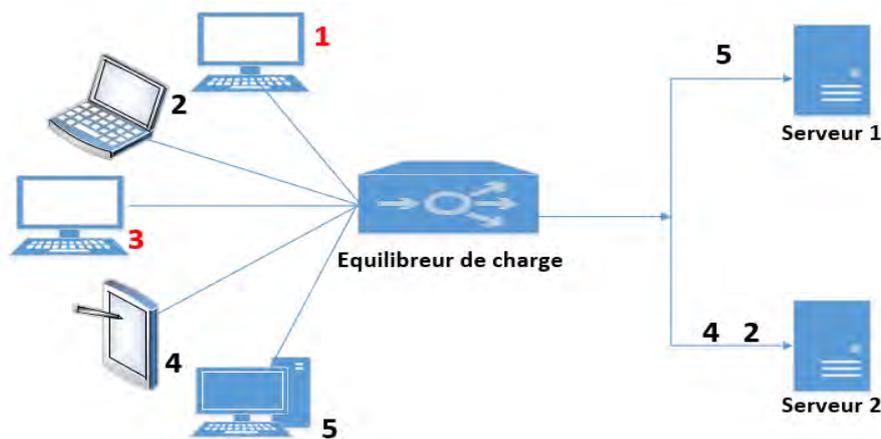


Figure 15: L'algorithme Least Connection

- **Weighted Least Connections** : avec cet algorithme, un composant poids basé sur les capacités de chaque serveur est introduit. On doit spécifier au préalable le poids de chaque serveur comme dans le Weighted Round Robin.

Un équilibreur de charge qui implémente le Weighted Least Connections prendra désormais deux choses en considération : les poids/capacités de chaque serveur et le nombre actuel de clients connectés sur chaque serveur.

- **Random** : il met une correspondance aléatoire les clients et les serveurs, c'est-à-dire en utilisant un générateur de nombres aléatoires sous-jacent. Au cas où l'équilibreur de charge reçoit un grand nombre de requêtes, l'algorithme Random sera en mesure de distribuer les requêtes uniformément aux nœuds. Comme le Round Robin, le Random est suffisant pour les clusters constitués de nœuds avec des configurations identiques.

## III.2. Gestion de la haute disponibilité des dispositifs

De nos jours, les systèmes de gestion de l'apprentissage (LMS) sont largement utilisés pour ajouter une valeur supplémentaire au processus d'apprentissage traditionnel. L'un des LMS open source les plus populaires est Moodle et est utilisé dans de nombreuses universités. Les données dans Moodle augmentent constamment avec de plus en plus d'utilisateurs (étudiants, etc.), de cours, de matériaux, de journaux, etc. Cela augmente le temps de réponse du serveur et entraîne une forte charge du serveur.

L'objectif principal de cette partie est de proposer une architecture Moodle hautement disponibilité.

Puisque la plateforme e-learning choisi (Moodle) va s'intégrer avec BigBluButton, alors l'architecture va garantir la haute disponibilité de ces deux dispositifs.

Cependant, notons que la manière dont la haute disponibilité est gérée pour la plateforme Moodle est différente de celle utilisée pour BigBluButton.

### **III.2.1. Gestion de la haute disponibilité de Moodle**

Une plateforme Moodle est constituée de trois (3) éléments que sont :

- Le code de Moodle (les fichiers PHP, JS, CSS, ...);
- Le contenu de la base de données stocké dans une autre application type MySQL, PostgreSQL, ...;
- Et le dossier Moodledata.

Sans l'un de ces trois (3) éléments, la plateforme ne sera pas complète.

La conception de Moodle avec une séparation des couches d'application permet des configurations évolutives. D'ailleurs les grands sites séparent généralement le serveur web et la base de données sur des serveurs distincts.

Il est aussi possible d'équilibrer la charge d'une installation Moodle en utilisant plusieurs serveurs Web qui doivent interroger la même base de données et faire référence au même magasin de fichiers. De même, cette base de données peut être un cluster de serveurs.

Mais notons que, la méthode la plus rapide et la plus efficace que nous puissions faire pour améliorer les performances d'un serveur web Moodle est d'augmenter sa quantité de RAM par exemple 4 Go ou plus. L'augmentation de la mémoire permettra à un serveur web Moodle de gérer plus d'utilisateurs simultanément (50 utilisateurs simultanés par giga de ram).

Alors pour atteindre à la fois la haute disponibilité et la scalabilité de Moodle, nous allons :

- Utiliser un pool de serveurs web qui vont interroger un cluster de base de données.
- Les nœuds du cluster vont être tous des maîtres car Moodle repose beaucoup sur les interactions synchrones. Ainsi les utilisateurs peuvent écrire et lire sur n'importe quel nœud du cluster.
- Chaque serveur du pool sera configuré avec une quantité de RAM en fonction des besoins en nombre de connections.

### **III.2.2. Gestion de la haute disponibilité de BigBluButton**

BigBluButton (BBB) : il fournit un large éventail de fonctionnalités de vidéoconférence (audio, vidéo, partage d'écran, tableau blanc, enregistrement de session, etc.) le tout sur une interface utilisateur simple et intuitive.

BBB fournit une solution libre-service, évolutive et conviviale qui permet aux utilisateurs de créer des comptes d'utilisateurs, de créer des réunions, d'inviter des utilisateurs qui peuvent ensuite confirmer l'invitation et de suivre si les utilisateurs y participent ou non, ainsi que de nombreuses autres fonctionnalités. La création de l'API BigBluButton sur les serveurs BigBluButton est nécessaire pour permettre aux applications clientes comme Moodle de créer des salles et des réunions sur l'une des instances de serveur.

Pour gérer la haute disponibilité de BigBluButton nous allons utiliser :

- **Scalelite** qui est un équilibreur de charge open-source, conçu spécifiquement pour BigBluButton, qui répartit uniformément la charge de réunion sur un pool de serveurs BigBluButton. Scalelite fait apparaître le pool de serveurs BigBluButton à notre application frontale Moodle, comme un serveur BigBluButton unique mais très évolutif. **Scalelite** a été publié par Blindside Networks sous licence AGPL le 13 mars 2020, en réponse à la forte demande des universités cherchant à faire évoluer BigBluButton suite à la pandémie COVID-19 [16].

Cependant pour accueillir un nombre important d'utilisateurs sur chaque serveur, les serveurs BigBluButton doivent chacun assurer une scalabilité verticale. Ainsi chacun d'eux sera configuré avec des capacités importantes pour garantir une scalabilité verticale. Et notons qu'un seul serveur BigBluButton qui répond à la configuration minimale prend en charge environ 200 utilisateurs simultanés.

### III.3. Architecture proposée

Pour la réalisation, nous proposons une architecture illustrée par la figure ci-après :

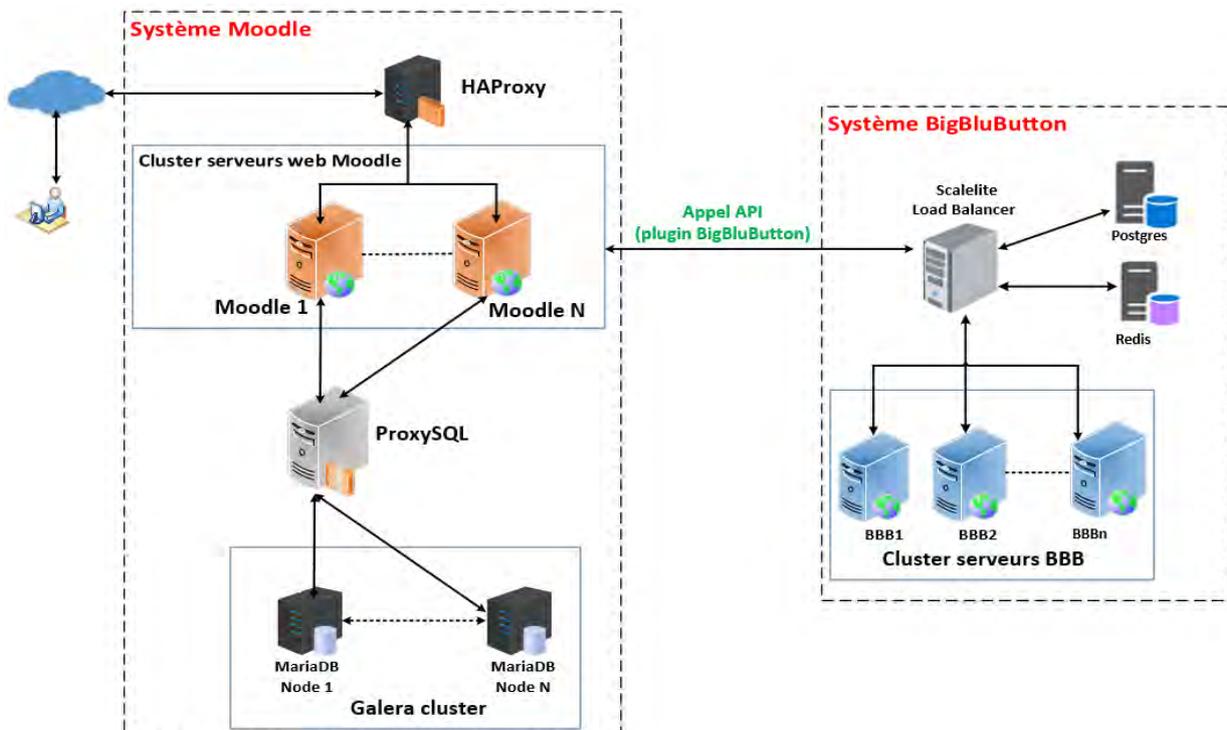


Figure 16: Architecture proposée

Notre architecture est composée de deux parties : la partie pour le système Moodle et la partie pour le système BigBluButton.

### III.3.1. Système Moodle

Dans cette partie nous avons :

- **Un serveur HAProxy** : qui est une machine dédiée faisant office de frontal, nous permettant aussi de redisperser les différents accès aux serveurs d'application Moodle suivant l'algorithme d'équilibrage de charge **least connexion** ;
- **Un pool de serveurs web Moodle** : sur chaque serveur est installé le code Moodle et le dossier Moodledata ;
- **Un serveur ProxySQL** : qui se charge de déplacer le trafic d'écriture vers l'un des nœuds du cluster ;
- **Un cluster de base de données basé sur Galera** : MariaDB Cluster est un système de réplication Multi Master, utilisant Galera qui est basé sur une méthode de réplication synchrone (ou pratiquement synchrone), qui garantit les données appliquées à d'autres nœuds avant leur validation. Avoir les mêmes données sur tous les nœuds signifie que les pannes de nœuds peuvent être facilement tolérées et qu'aucune donnée n'est perdue. Il est également plus facile de basculer vers un autre nœud, car tous les nœuds sont à jour avec les mêmes données. Il est juste de dire que MariaDB Cluster est une solution de haute disponibilité qui peut assurer une disponibilité élevée.

### III.3.2. Système BigBluButton

Dans cette partie nous avons :

- **Un serveur Scalelite** : un équilibreur de charge qui a quatre composants principaux : le proxy, l'API frontend, le scrutateur de réunion, l'importateur d'enregistrements.
  - **Le proxy** : qui inclut des fichiers de configuration pour gérer les requêtes de la même manière que BigBluButton.
  - **L'API frontend** : gère la demande et la distribue aux serveurs BigBluButton réels.
  - **Le scrutateur de réunion** : un script Ruby exécuté en tant que tâche planifiée pour interroger le statut des serveurs BigBluButton enregistrés.
  - **L'importateur d'enregistrements** : un script Ruby qui est exécuté en tant que tâche planifiée pour importer le flux d'enregistrement par les serveurs BigBluButton dans la base de données.
- **Un serveur PostgreSQL et un serveur Redis** : connectées à Scalelite servant de bases de données pour gérer la persistance des enregistrements ;
- **Un pool de serveur BigBluButton** : qui sera utilisé par le serveur Scalelite qui à son tour va choisir le moins chargé et le lier avec le serveur Moodle qui va demander un établissement de session.

- **Des utilisateurs** : qui peuvent accéder à la plateforme Moodle pour suivre une réunion grâce à BigBluButton, ils peuvent aussi simplement se connecter sur le serveur web Moodle choisi par l'équilibreur de charges pour faire leurs devoirs, examens, télécharger des supports de cours, participer à des chats etc...

Chaque jour, la plateforme Moodle accueille un nombre important étudiants et enseignants répartis entre plusieurs classes virtuelles. Un équilibreur de charge Scalelite absorbe les pics de connexion entre les différents serveurs BBB et adoucit la charge.

Afin de participer aux cours, les étudiants se connectent à la plateforme Moodle fourni par leur faculté et rejoignent une salle de classe préconfigurée dans Moodle. Ils sont automatiquement identifiés et ajoutés à la « liste des apprenants ». Les enseignants assument le rôle de modérateur de la classe et peuvent gérer leur classe avec toute l'autonomie dont ils ont besoin.

Avec cette architecture, les utilisateurs (enseignants et étudiants) sont connectés sur Moodle (à charge équilibrée) et le cluster de serveurs BigBluButton fournit les salles de visioconférence. Celui-ci est réparti en charge par le loadbalancer Scalelite qui interroge périodiquement chaque BigBluButton pour vérifier s'il est joignable en ligne, prêt à recevoir des requêtes d'API et pour déterminer sa charge actuelle (nombre de réunions en cours).

Avec ces informations, lorsque Scalelite reçoit un appel API entrant pour créer une nouvelle réunion, il place la nouvelle réunion sur le serveur le moins chargé du pool. De cette manière, Scalelite peut équilibrer la charge des demandes de réunion de manière uniforme dans le pool.

### III.4. Principe et fonctionnement

Le fonctionnement de l'architecture est le suivant :

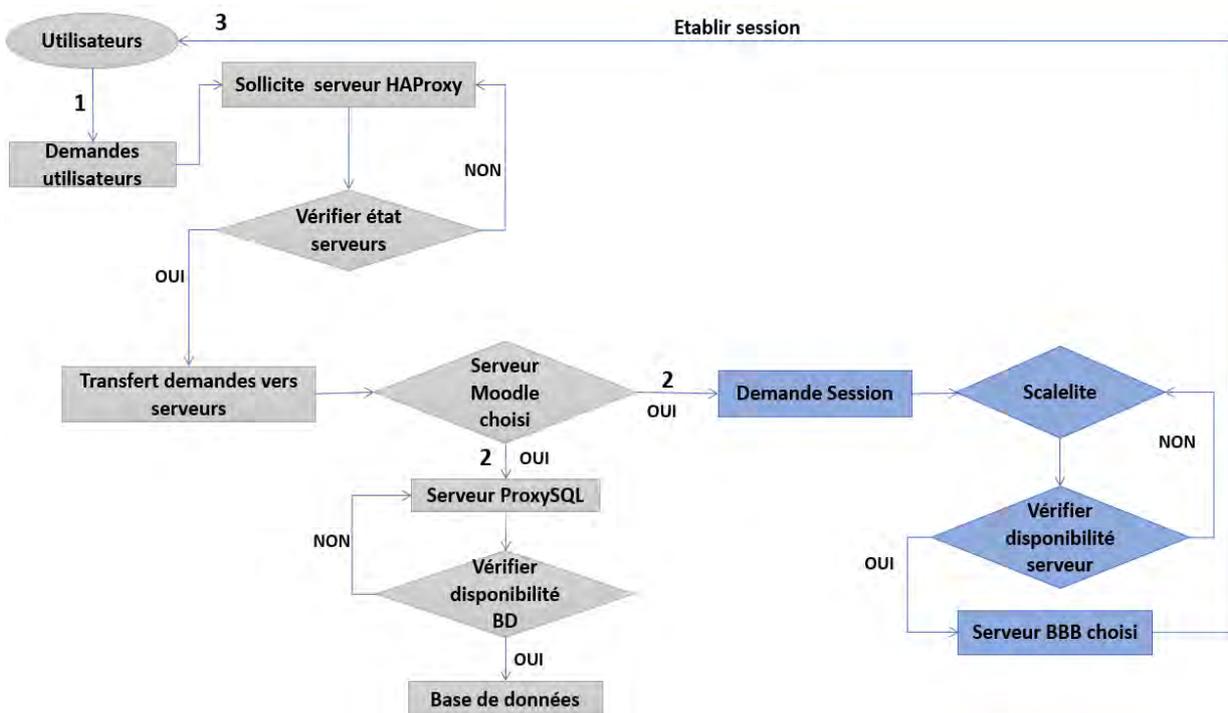


Figure 17: Diagramme fonctionnel

Les utilisateurs se connectent sur leurs terminaux et tapent l'URL d'accès de leur plateforme de cours en ligne et arrivent directement sur la machine dédiée qui fait office de frontal dans laquelle nous utilisons le loadbalancer HAProxy qui va nous servir de redistribuer les différents accès.

Donc une fois que les utilisateurs arrivent sur le frontal, le frontal aiguille les requêtes vers l'un des serveurs suivant l'algorithme least connexion.

Une fois dans la plateforme, les utilisateurs peuvent demander des ressources, ou solliciter l'accès à une classe virtuelle.

- Alors s'ils font des demandes de ressources, le serveur Moodle reposant sur des sessions et requêtes de traitement peut facilement saturer la base de données avec trop de connexions en cours d'exécution simultanées.

ProxySQL va assurer la mise en file d'attente et la limitation des connexions vers les bases de données MariaDB et empêche un seul serveur d'être surchargé avec trop de requêtes.

Tous les clients se connectent à l'instance ProxySQL qui va transmettre la connexion à l'un des serveurs MariaDB disponibles en fonction de l'algorithme d'équilibrage de charge Round Robin.

- Lors que les utilisateurs sollicitent l'accès à une classe virtuelle, les serveurs Moodle dans lequel ils trouvent vont envoyer des requêtes API BigBlueButton standard au serveur Scalelite qui distribue ces requêtes au serveur BigBlueButton le moins chargé du pool.

Le serveur Scalelite interroge périodiquement chaque BigBlueButton pour vérifier s'il est joignable en ligne, prêt à recevoir des requêtes API et pour déterminer sa charge actuelle (nombre de réunions en cours). De ce fait Scalelite reçoit un appel API entrant pour créer une nouvelle réunion, il place la nouvelle réunion sur le serveur le moins chargé du pool pour équilibrer la charge des demandes de réunion de manière uniforme dans le pool.

A la sortie la demande de session est établie pour les utilisateurs finaux.

## III.5. Diagramme des flux

Le diagramme ci-après représente les différentes requêtes des dispositifs :

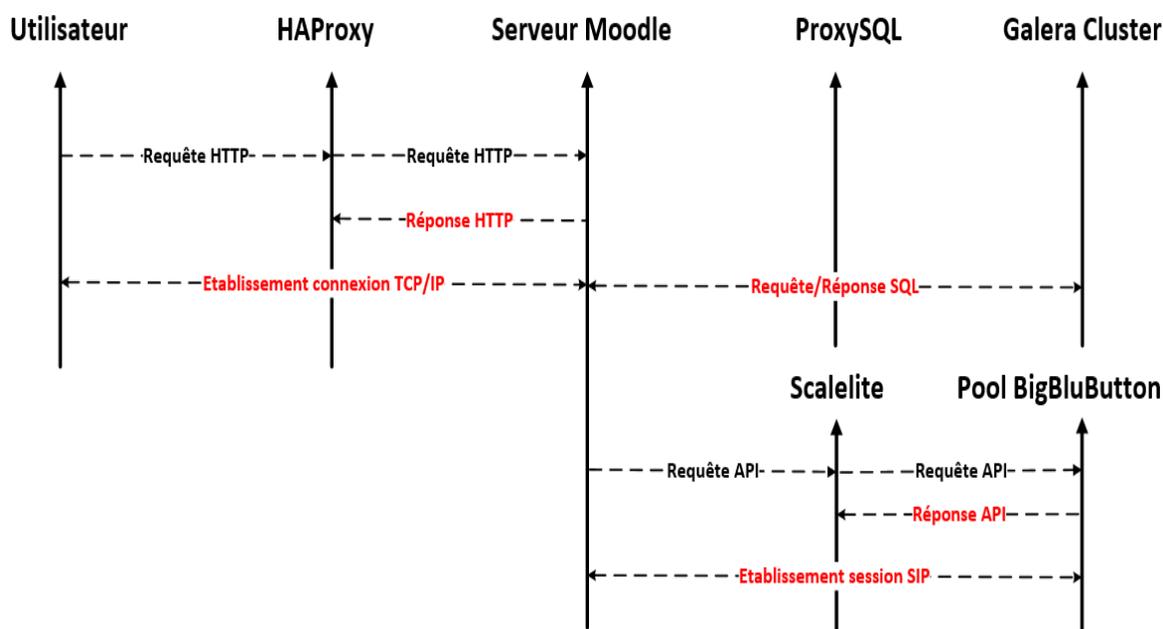


Figure 18: Diagramme des flux

Dans cette figure, la requête HTTP (HyperText Transfer Protocol) envoyée par l'utilisateur est adressée au serveur HAProxy qui détermine, selon l'algorithme least connexion, le serveur Moodle auquel il va affecter la connexion, parmi les serveurs du pool. Alors une fois le serveur Moodle envoie une réponse HTTP au serveur HAProxy, une connexion TCP/IP (Transmission Control Protocol/Internet Protocol) est établie et l'équipement de répartition de charge devient pratiquement transparent : dans son rôle de base, il transfère les paquets IP de l'utilisateur vers le serveur sélectionné et vice versa jusqu'à fermeture de la connexion.

Une fois dans le serveur web Moodle, les requêtes SQL envoyées passent d'abord par le ProxySQL qui va ensuite les transmettre au serveur choisi dans le cluster de base de données. Ainsi une liaison requête/réponse SQL sera établie entre le serveur web Moodle et le nœud de base données choisi.

Pour établir une session SIP (Session Initiation Protocol), le serveur web Moodle va envoyer une requête API au serveur Scalelite qu'il croit être le serveur BigBluButton. Le serveur Scalelite va ensuite interroger le pool de serveur BigBluButton en envoyant une requête API afin de pouvoir établir une session SIP entre le serveur web Moodle et le serveur BigBluButton choisi.

## III.6. Maquette de test

Dans la maquette de test nous avons :

- Un serveur HAProxy ;
- Deux serveurs web Moodle ;
- Un serveur ProxySQL ;
- Un cluster de base de données MariaDB composé de trois nœuds ;
- Un serveur Scalelite ;
- Trois serveurs BigBluButton.

Tous ces serveurs sont virtuels et nous allons utiliser Proxmox VE comme environnement de virtualisation.

Le cluster de base de données et Scalelite fonctionnent avec un minimum de trois serveurs raison pour laquelle, nous allons utiliser trois serveurs de base de données et trois serveurs de BigBluButton.

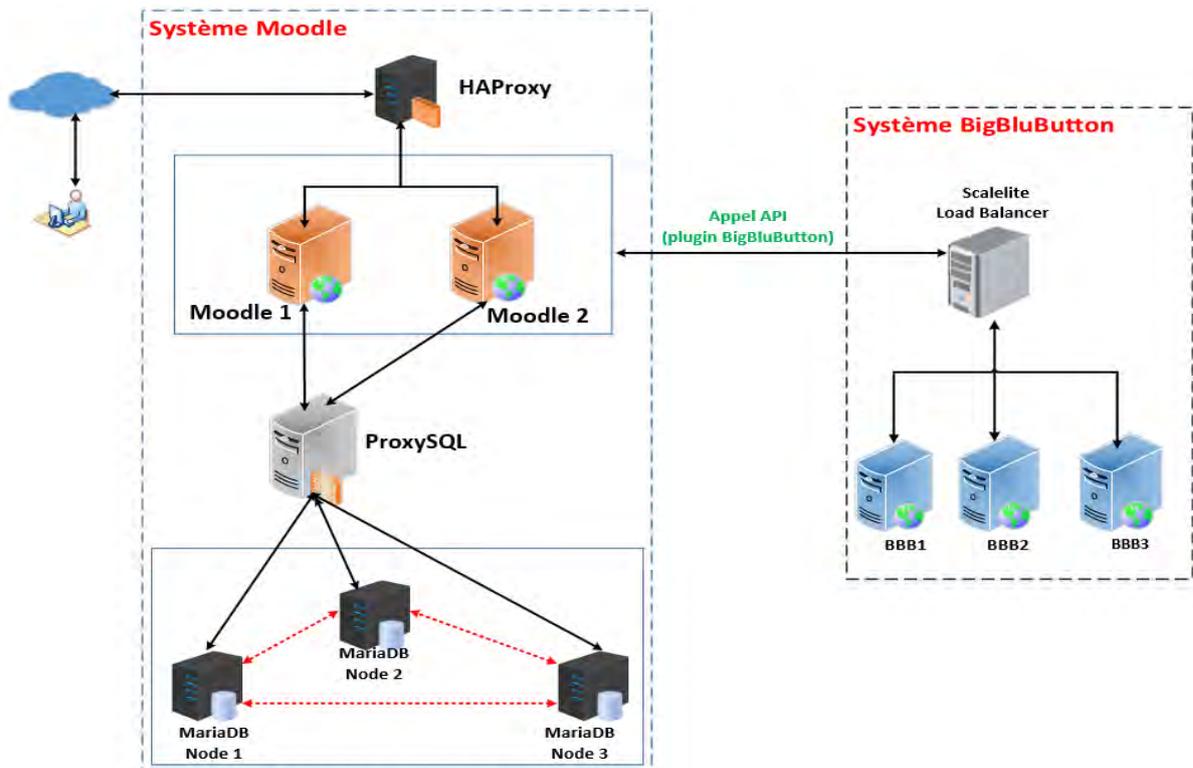


Figure 19: Prototype de test

## III.7. Implémentation

Pour la mise en place de l'environnement virtualisé, nous avons utilisé deux serveurs physiques dont leurs caractéristiques sont les suivantes :

### Serveur physique 1 :

- ✓ Processeurs : 24 CPU(s)
- ✓ Mémoire (RAM) : 32 GiB
- ✓ Disque (Storage) : 290 GiB
- ✓ ISO Proxmox Virtual Environment (PVE) : Version 6.2-4
- ✓ Système d'exploitation : Debian

### Serveur physique 2 :

- ✓ Processeurs : 8 CPU(s)
- ✓ Mémoire (RAM) : 5 GiB
- ✓ Disque (Storage) : 146 GiB
- ✓ ISO Proxmox Virtual Environment (PVE) : Version 6.2-4
- ✓ Système d'exploitation : Debian

### III.7.1. Mise en place de Proxmox Virtual Environment

Dans cette partie, nous allons nous initier à l'installation d'un serveur dédié en utilisant l'outil « Proxmox VE », un outil Open source.

Une fois l'ISO lancé sur notre serveur physique, nous allons voir la fenêtre ci-dessous. Nous sélectionnons ensuite « Install Proxmox VE » :



**Figure 20: Installation Proxmox VE**

Acceptons la licence d'utilisation de Proxmox en cliquant sur « **I agree** ».

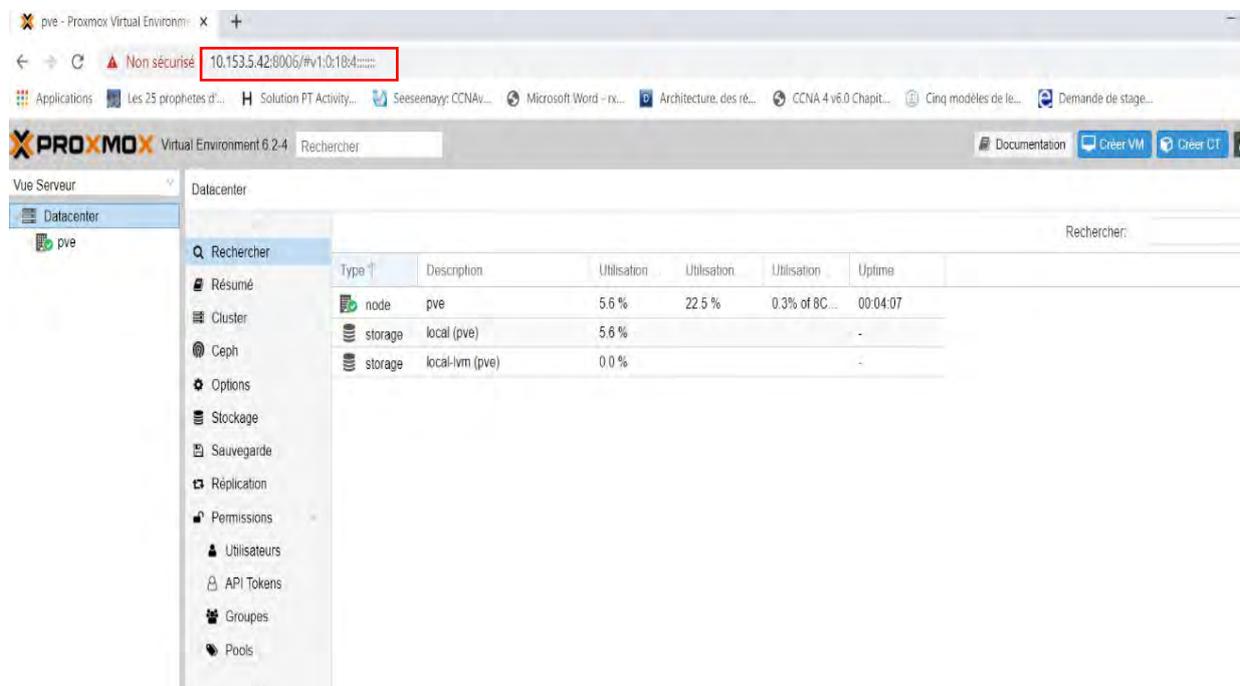
Ensuite nous allons choisir le disque sur lequel nous souhaitons installer notre système, ici nous avons qu'un seul disque qui est déjà sélectionné par défaut.

Nous devons renseigner notre zone « **Africa/Dakar** », afin d'avoir automatiquement la bonne configuration de l'heure et de notre clavier.

Renseignons **un mot de passe** pour le compte **root** de Proxmox ainsi que notre adresse mail pour recevoir des alertes sur le fonctionnement de votre serveur.

Configurons les paramètres de notre carte réseau pour avoir une adresse IP afin d'être connecté à Internet. Ensuite nous allons cliquer sur « Install » en bas à droite pour lancer l'installation.

Après installation, nous pouvez cliquer sur « reboot » pour démarrer sur notre nouveau système. A présent nous pouvons nous connecter à l'interface du serveur en renseignant notre adresse IP sur le navigateur, ainsi que le port d'écoute.



**Figure 21: Interface Proxmox**

L'installation de notre Proxmox est terminée.

### III.7.2. Implémentation du système Moodle

Dans cette partie, nous allons implémenter les composants du système Moodle : le cluster Galera MariaDB, le serveur ProxySQL, le serveur frontal HAProxy et terminer avec le cluster Moodle.

Pour pallier au problème de connectivité du réseau interne, nous avons jugé de continuer le déploiement sur un environnement propriétaire pour acheter des machines virtuelles.

### III.7.2.1. Installation du cluster Galera MariaDB

Un cluster MariaDB doit toujours être monté au moins avec trois nœuds. La raison est simple s'il n'y a que deux nœuds et que l'un des deux échoue, il n'y a aucun moyen de savoir lequel est corrompu.

Dans cette partie, nous allons utiliser trois (3) serveurs pour la mise en place du cluster de base de données.

Prérequis :

- ✓ Système d'exploitation : Ubuntu 18.04 LTS
- ✓ RAM : 8GiB
- ✓ CPU : 2
- ✓ Disque : 100GiB
- ✓ MariaDB version : 10.3
- ✓ Nœud\_galera1 : 23.102.152.204
- ✓ Nœud\_galera2 : 13.66.57.183
- ✓ Nœud\_galera3 : 40.74.244.231

Nous avons décidé d'aller avec la convention de dénomination 'galera1, galera2, galera3', donc nous avons édité le fichier `/etc/hosts` sur tous les nœuds et les lister avec leurs adresses IP respectives. Nous avons également fait le changement dans le fichier `/etc/hostname` pour utiliser le nouveau nom pour les nœuds.

```
root@galera1: ~
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
23.102.152.204 galera1
13.66.57.183 galera2
40.74.244.231 galera3

root@galera2: ~
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
23.102.152.204 galera1
13.66.57.183 galera2
40.74.244.231 galera3

root@galera3: ~
GNU nano 2.9.3 /etc/hosts
127.0.0.1 localhost
23.102.152.204 galera1
13.66.57.183 galera2
40.74.244.231 galera3
```

Figure 22: Configuration des noms d'hôtes

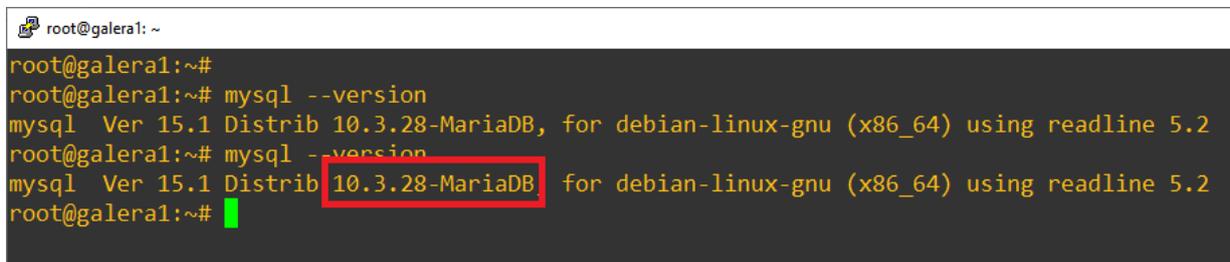
Lorsque cela est fait, nous pouvons commencer à configurer notre cluster Galera.

### ➤ Installer et configurer MariaDB sur le nœud\_1

Pour installer MariaDB, la première chose que nous allons faire est d'ajouter les référentiels MariaDB correspondant à Ubuntu 18.04 puis nous installerons MariaDB :

```
apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8
add-apt-repository 'deb [arch=amd64] http://mirror.zol.co.zw/mariadb/repo/10.3/ubuntu bionic
main'
apt update && apt install mariadb-server
```

Après installation nous pouvons voir la version de MariaDB installé avec succès.



```
root@galera1: ~
root@galera1:~# mysql --version
mysql Ver 15.1 Distrib 10.3.28-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2
root@galera1:~# mysql --version
mysql Ver 15.1 Distrib 10.3.28-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2
root@galera1:~#
```

**Figure 23: Vérification de la version MariaDB**

Maintenant que MariaDB est installée, nous allons procéder à la configuration du cluster. Nous allons spécifier le nom de notre cluster dans **wsrep\_cluster\_name**. Nous spécifierons l'adresse IP du nœud 1 dans **wsrep\_cluster\_address**. Enfin, nous allons configurer l'IP du galera1 dans **wsrep\_node\_address** et le nom galera1 dans **wsrep\_node\_name**. Nous ferons tout cela dans le fichier **/etc/mysql/conf.d/cluster.cnf**.

```
root@galera1:~# nano /etc/mysql/conf.d/cluster.cnf
```

```
GNU nano 2.9.3 /etc/mysql/conf.d/cluster.cnf
[mysql]
query_cache_size=0
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
query_cache_type=0
bind-address=0.0.0.0
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
join_buffer_size = 1M
# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
#wsrep_provider_options="gcache.size=32G"
# Galera Cluster Configuration
wsrep_cluster_name="RETEL"
wsrep_cluster_address="gcomm://23.102.152.204"

# Galera Synchronization Configuration
wsrep_sst_method=rsync
#wsrep_sst_auth=user:pass

# Galera Node Configuration
# ESPECIFICAR LOS DATOS DEL SERVIDOR LOCAL
wsrep_node_address="23.102.152.204"
wsrep_node_name="galera1"
```

Figure 24: Configuration nœud 1

Nous allons maintenant procéder à l'arrêt du service MariaDB.

```
root@galera1:~# systemctl stop mariadb
```

Nous allons configurer les variables et faire écouter MariaDB sur toutes les IP, pour cela nous éditerons `/etc/mysql/my.cnf`.

```
nano /etc/mysql/my.cnf
#bind-address = 127.0.0.1
max_connections = 500
expire_logs_days = 5
wait_timeout = 28800
max_allowed_packet = 1G
connect_timeout = 6
```

**Remarque :** Cette partie ne s'applique que sur le nœud 1

### ➤ Installer et configurer MariaDB nœud\_2

La même procédure que l'installation du galera1 sera appliquée pour le galera 2, seulement dans le fichier `/etc/mysql/conf.d/cluster.cnf`, nous allons renseigner l'adresse IP de galera1 et de galera2 dans `wsrep_cluster_address` et le nom galera2 dans `wsrep_node_name`.

```
GNU nano 2.9.3 /etc/mysql/conf.d/cluster.cnf

[mysqld]
query_cache_size=0
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
query_cache_type=0
bind-address=0.0.0.0
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
join_buffer_size = 1M
# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
#wsrep_provider_options="gcache.size=32G"
# Galera Cluster Configuration
wsrep_cluster_name="RETEL"
wsrep_cluster_address="gcomm://23.102.152.204,13.66.57.183"

# Galera Synchronization Configuration
wsrep_sst_method=rsync
#wsrep_sst_auth=user:pass

# Galera Node Configuration
# ESPECIFICAR LOS DATOS DEL SERVIDOR LOCAL
wsrep_node_address="13.66.57.183"
wsrep_node_name="galera2"
```

Figure 25: Configuration nœud 2

Nous allons maintenant procéder à l'arrêt du service MariaDB :

```
root@galera2:~# systemctl stop mariadb
```

### ➤ Installer et configurer Mariadb nœud\_3

La même procédure que l'installions du galera1 sera appliquée pour le galera 3, seulement dans le fichier `/etc/mysql/conf.d/cluster.cnf`, nous allons renseigner l'adresse IP de galera1, galera2 et galera 3 dans `wsrep_cluster_address` et le nom galera3 dans `wsrep_node_name`.

```
GNU nano 2.9.3 /etc/mysql/conf.d/cluster.cnf

[mysqld]
query_cache_size=0
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
query_cache_type=0
bind-address=0.0.0.0
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
join_buffer_size = 1M
# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so
#wsrep_provider_options="gcache.size=32G"
# Galera Cluster Configuration
wsrep_cluster_name="RETEL"
wsrep_cluster_address="gcomm://23.102.152.204,13.66.57.183,40.74.244.231"
# Galera Synchronization Configuration
wsrep_sst_method=rsync
#wsrep_sst_auth=user:pass
# Galera Node Configuration
# ESPECIFICAR LOS DATOS DEL SERVIDOR LOCAL
wsrep_node_address="40.74.244.231"
wsrep_node_name="galera3"
```

Figure 26: Configuration nœud 3

Nous allons maintenant procéder à l'arrêt du service MariaDB :

```
root@galera3:~# systemctl stop mariadb
```

### ➤ Démarrage du cluster

Nous avons déjà toute la configuration prête, nous allons procéder au démarrage du cluster MariaDB, pour cela nous allons exécuter dans galera1 la commande suivante :

```
root@galera1:~# galera_new_cluster
```

Maintenant, nous allons démarrer MariaDB sur le deuxième et le troisième nœud :

```
root@galera2:~# systemctl start mariadb
```

```
root@galera3:~# systemctl start mariadb
```

Nous pouvons voir que le cluster fonctionne en exécutant la commande suivante sur le galera1 :

```
root@galera1: ~  
root@galera1:~# mysql -u root -p -e "SHOW STATUS LIKE 'wsrep_cluster_size';"  
Enter password:  
+-----+  
| Variable_name | Value |  
+-----+  
| wsrep_cluster_size | 3 |  
+-----+  
root@galera1:~#
```

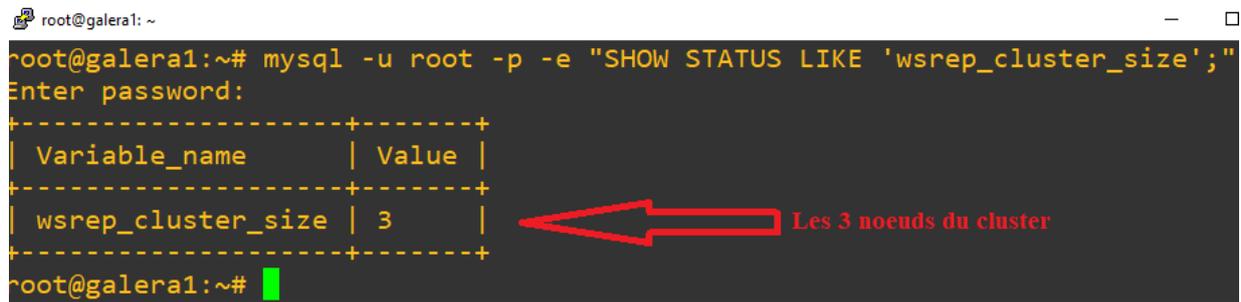


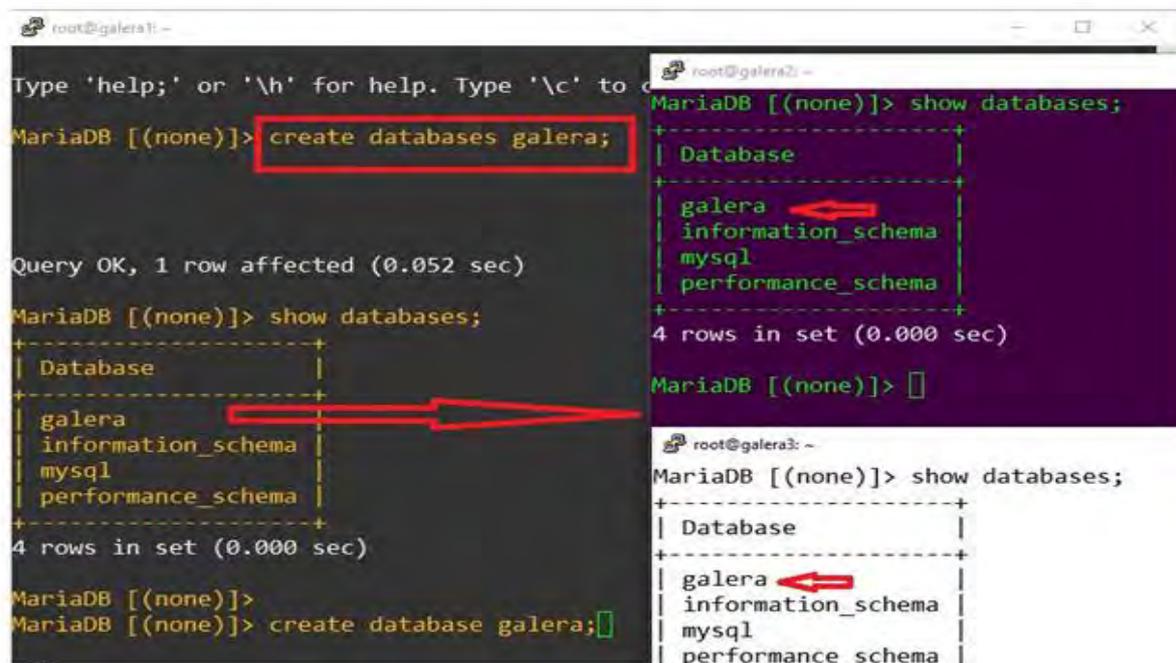
Figure 27: Taille du cluster

**Wsrep\_cluster\_size =3** montre que les trois nœuds (galera1, galera2, galera3) sont interconnectés.

➤ **Test de la réplication entre les trois nœuds du cluster galera**

Nous pouvons maintenant créer une base de données sur n'importe quel nœud du cluster et elle sera répliquée sur les deux autres nœuds.

Sur le nœud 1 nous allons créer une base de données nommé « **galera** » :



```
root@galera1: ~  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> create databases galera;  
Query OK, 1 row affected (0.052 sec)  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| galera |  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.000 sec)  
MariaDB [(none)]> create database galera;  
root@galera2: ~  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| galera |  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.000 sec)  
MariaDB [(none)]>   
root@galera3: ~  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| galera |  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.000 sec)  
MariaDB [(none)]>
```

Figure 28: Test de la réplication entre les nœuds

A ce stade notre cluster est fonctionnel et se compose de trois nœuds. Nous avons réussi à déployer MariaDB 10.3 Galera Cluster. Nous passons maintenant à l'installation de ProxySQL.

### III.7.2.2. Installation du ProxySQL

ProxySQL est un proxy équilibreur de charge hautes performances pour les clusters de bases de données MySQL. ProxySQL est gratuit et open source et il est distribué sous licence GPL. ProxySQL prend en charge des milliers de connexions simultanées. ProxySQL est hautement personnalisable via une interface de ligne de commande de type SQL.

Prérequis :

- ✓ Système d'exploitation : CentOS 7
- ✓ RAM : 8GiB
- ✓ CPU : 2
- ✓ Disque : 100GiB

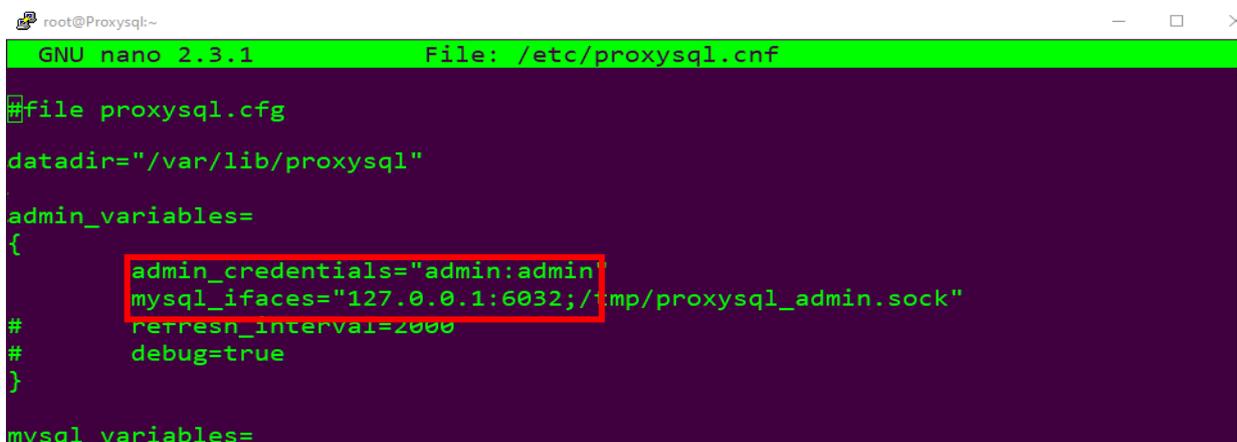
Tout d'abord, nous devons installer et démarrer ProxySQL :

```
wget https://github.com/sysown/proxysql/releases/download/v1.2.1/proxysql-1.2.1-1-centos7.x86_64.rpm
rpm -i proxysql-1.2.1-1-centos7.x86_64.rpm
service proxysql start
```

Ensuite, nous devons télécharger un script que nous utiliserons pour surveiller l'état de Galera. Le script doit se trouver dans `/var/lib/proxysql`.

```
wget
https://raw.githubusercontent.com/sysown/proxysql/master/tools/proxysql_galera_checker.sh
mv proxysql_galera_checker.sh /var/lib/proxysql/
chmod u+x /var/lib/proxysql/proxysql_galera_checker.sh
```

Nous devons nous connecter à l'interface d'administration de ProxySQL. Pour cela, Nous devons connaître les informations d'identification qui se trouve dans le fichier `/etc/proxysql.cnf`



```
root@Proxysql:~
GNU nano 2.3.1 File: /etc/proxysql.cnf
#file proxysql.cfg
datadir="/var/lib/proxysql"
admin_variables=
{
  admin_credentials="admin:admin"
  mysql_ifaces="127.0.0.1:6032;/tmp/proxysql_admin.sock"
#   resresn_interval=2000
#   debug=true
}
mysql_variables=
```

Figure 29: Informations d'authentification

### ➤ Connexion au serveur ProxySQL

Connaissant ces informations d'identification et les interfaces sur lesquelles ProxySQL écoute, nous pouvons nous connecter à l'interface d'administration et commencer la configuration.

```
mysql -P6032 -uadmin -padmin -h 127.0.0.1
```

Tout d'abord, nous devons remplir la table `mysql_servers` avec des informations sur nos nœuds Galera. Nous les ajouterons deux fois, à deux groupes d'hôtes différents :

- ✓ groupe d'hôtes (avec `hostgroup_id` de 0) gèrera les écritures
- ✓ groupe d'hôtes (avec `hostgroup_id` de 1) gèrera les lectures

```
MySQL [(none)]> INSERT INTO mysql_servers (hostgroup_id, hostname, port) VALUES (0, '23.102.152.204', 3306), (0, '13.66.57.183', 3306), (0, '40.74.244.231', 3306);
```

```
MySQL [(none)]> INSERT INTO mysql_servers (hostgroup_id, hostname, port) VALUES (1, '23.102.152.204', 3306), (1, '13.66.57.183', 3306), (1, '40.74.244.231', 3306);
```

### ➤ Ajout d'un utilisateur

Nous devons ajouter des informations sur les utilisateurs qui seront utilisées par l'application.

Nous avons utilisé un mot de passe en texte brut ici, mais ProxySQL accepte également les mots de passe hachés au format MySQL.

```
MySQL [(none)]> INSERT INTO mysql_users (username, password, active, default_hostgroup) VALUES ('admin', 'admin', 1, 0);
```

Nous devons définir des règles de requête qui gèreront la division lecture / écriture. Tout d'abord, nous voulons faire correspondre toutes les requêtes SELECT :

```
MySQL [(none)]> INSERT INTO mysql_query_rules (active, match_pattern, destination_hostgroup, apply) VALUES (1, '^SELECT.*', 1, 0);
```

```
MySQL [(none)]> INSERT INTO mysql_query_rules (active, match_pattern, destination_hostgroup, apply) VALUES (1, '^SELECT.*FOR UPDATE', 0, 1);
```

Ces paramètres fonctionneront correctement si l'autocommit est activé et qu'aucune transaction explicite n'est utilisée.

```
SET autocommit=0;  
BEGIN;
```

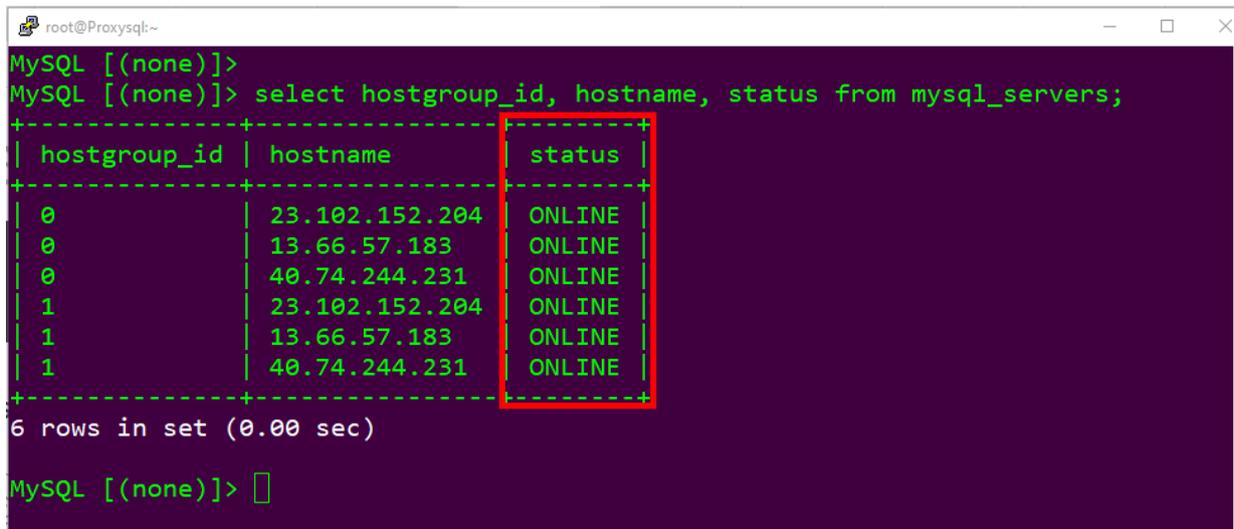
En raison de la façon dont Galera gère les nœuds abandonnés, nous voulons augmenter le nombre de tentatives que ProxySQL fera avant de décider qu'un hôte ne peut pas être atteint.

```
MySQL [(none)]> SET mysql-query_retries_on_failure=10;
```

Enfin, nous devons appliquer toutes les modifications que nous avons apportées à la configuration d'exécution et les enregistrer sur le disque.

```
MySQL [(none)]> LOAD MYSQL USERS TO RUNTIME; SAVE MYSQL USERS TO DISK;  
LOAD MYSQL QUERY RULES TO RUNTIME; SAVE MYSQL QUERY RULES TO DISK;  
LOAD MYSQL SERVERS TO RUNTIME; SAVE MYSQL SERVERS TO DISK; LOAD  
SCHEDULER TO RUNTIME; SAVE SCHEDULER TO DISK; LOAD MYSQL VARIABLES  
TO RUNTIME; SAVE MYSQL VARIABLES TO DISK;
```

Nous pouvons maintenant vérifier la table **mysql\_servers**:

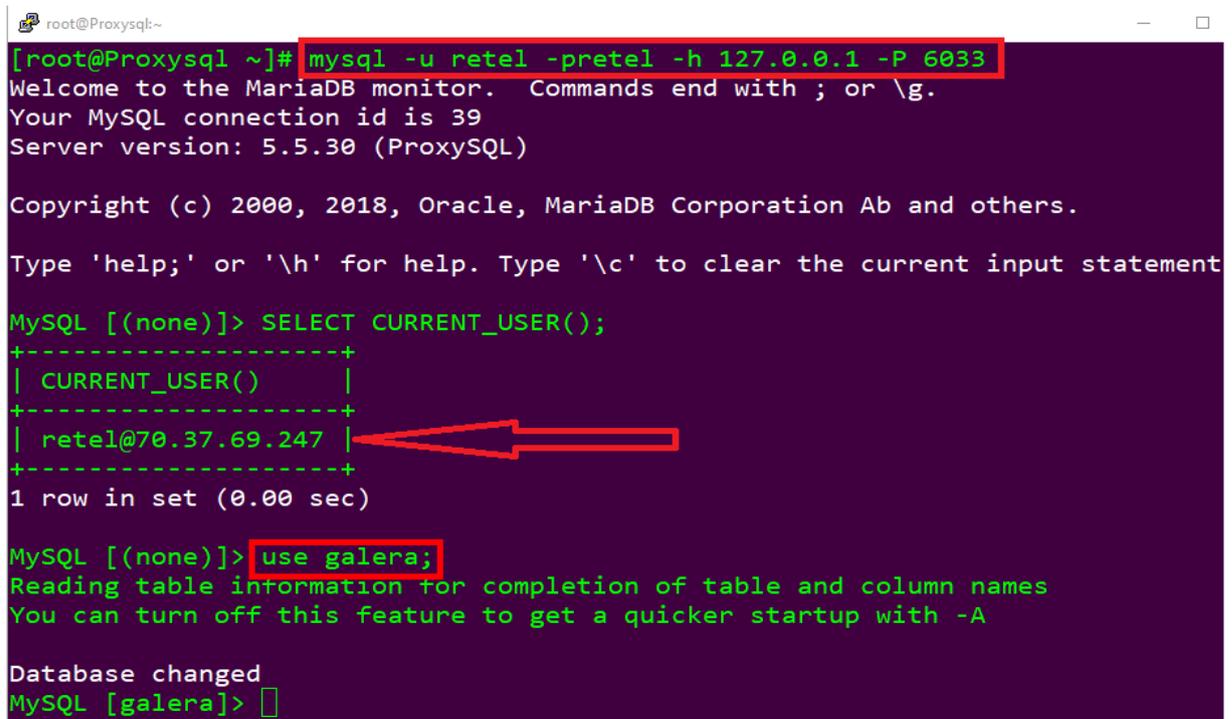


```
root@ProxySQL:~  
MySQL [(none)]>  
MySQL [(none)]> select hostgroup_id, hostname, status from mysql_servers;  
+-----+-----+-----+  
| hostgroup_id | hostname          | status |  
+-----+-----+-----+  
| 0             | 23.102.152.204   | ONLINE |  
| 0             | 13.66.57.183    | ONLINE |  
| 0             | 40.74.244.231   | ONLINE |  
| 1             | 23.102.152.204   | ONLINE |  
| 1             | 13.66.57.183    | ONLINE |  
| 1             | 40.74.244.231   | ONLINE |  
+-----+-----+-----+  
6 rows in set (0.00 sec)  
MySQL [(none)]> □
```

**Figure 30: Etat de la table mysql\_servers**

### ➤ Test d'accès aux nœuds du cluster à partir du ProxySQL

Un utilisateur sera créé sur l'un des nœuds et nous allons l'autoriser à accéder à la base de données du cluster à partir du ProxySQL : l'utilisateur « retel » avec comme mot de passe « retel » est créé :



```
root@Proxysql:~  
[root@Proxysql ~]# mysql -u retel -pretel -h 127.0.0.1 -P 6033  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MySQL connection id is 39  
Server version: 5.5.30 (ProxySQL)  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
  
MySQL [(none)]> SELECT CURRENT_USER();  
+-----+  
| CURRENT_USER() |  
+-----+  
| retel@70.37.69.247 |  
+-----+  
1 row in set (0.00 sec)  
  
MySQL [(none)]> use galera;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
MySQL [galera]> 
```

Figure 31: Connection aux nœuds depuis le ProxySQL

Après avoir installer cluster galera et ProxySQL, nous passons à l'étape d'installation du serveur frontal HAProxy.

### III.7.2.3. Installation du HAProxy

HAProxy est un équilibreur de charge TCP/HTTP open source, un serveur proxy et un terminateur SSL/TLS avec des performances et une fiabilité élevée pour les sites Web ayant un volume de trafic élevé.

Dans cette partie, nous allons montrer comment installer et configurer HAProxy sur CentOS 7.

Prérequis :

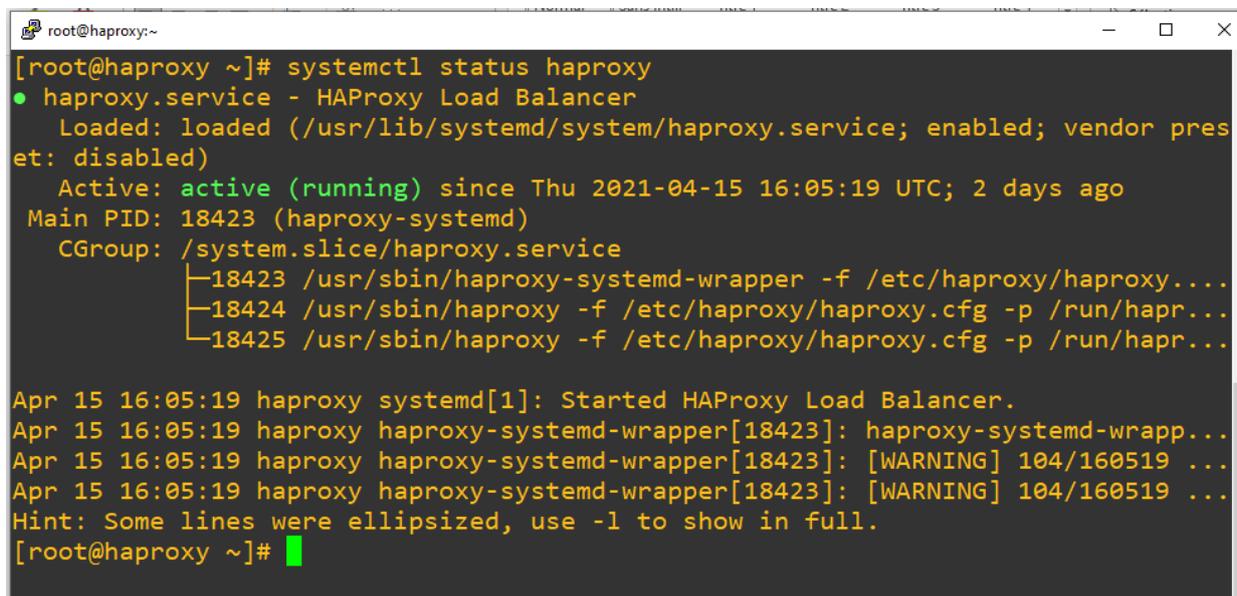
- ✓ Système d'exploitation : CentOS 7
- ✓ Firewall : enable firewalld
- ✓ SELinux : mode enabled
- ✓ HAProxy Adresse IP : 70.37.111.91

Pour installer HAProxy, nous le faisons simplement via le référentiel :

```
yum installer haproxy
```

Puis nous démarrons et activons le service HAProxy :

```
systemctl start haproxy
systemctl enable haproxy
systemctl status haproxy
```



```
root@haproxy:~
[root@haproxy ~]# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-04-15 16:05:19 UTC; 2 days ago
     Main PID: 18423 (haproxy-systemd)
    CGroup: /system.slice/haproxy.service
            └─18423 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy...
              └─18424 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/hapr...
                └─18425 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/hapr...

Apr 15 16:05:19 haproxy systemd[1]: Started HAProxy Load Balancer.
Apr 15 16:05:19 haproxy haproxy-systemd-wrapper[18423]: haproxy-systemd-wrapp...
Apr 15 16:05:19 haproxy haproxy-systemd-wrapper[18423]: [WARNING] 104/160519 ...
Apr 15 16:05:19 haproxy haproxy-systemd-wrapper[18423]: [WARNING] 104/160519 ...
Hint: Some lines were ellipsized, use -l to show in full.
[root@haproxy ~]#
```

**Figure 32: Statut du service HAProxy**

### ➤ Configurer SELinux

Pour autoriser l'accès HAProxy au port 5002 lorsque SELinux est activé, nous devons modifier les politiques SELinux. Alors émettez les commandes suivantes :

```
grep haproxy /var/log/audit/audit.log | audit2allow -M mypol
semodule -i mypol.pp
```

```
root@haproxy:~  
[root@haproxy ~]# nano /etc/selinux/config  
[root@haproxy ~]# sestatus  
SELinux status: enabled  
SELinuxfs mount: /sys/fs/selinux  
SELinux root directory: /etc/selinux  
Loaded policy name: targeted  
Current mode: enforcing  
Mode from config file: enforcing  
Policy MLS status: enabled  
Policy deny_unknown status: allowed  
Max kernel policy version: 31  
[root@haproxy ~]# █
```

Figure 33: Statut de SELinux

### ➤ Configure HAProxy

Il est maintenant temps de configurer HAProxy. Le fichier de configuration associé se trouve dans `/etc/haproxy/haproxy.cfg`. Ce fichier est divisé en quatre sections principales :

- ✓ Paramètres globaux : définit les paramètres à l'échelle du processus.
- ✓ Defaults : cette section définit les paramètres par défaut pour toutes les autres sections après sa déclaration.
- ✓ Frontend : cette section décrit un ensemble de sockets d'écoute acceptant les connexions client.
- ✓ Backend : cette section décrit un ensemble de serveurs auxquels le proxy se connectera pour transférer les connexions entrantes.

Nous devons indiquer au serveur syslog comment recevoir et traiter les journaux HAProxy :

```
nano /etc/rsyslog.d/haproxy.conf
```

```
root@haproxy:/etc/rsyslog.d  
GNU nano 2.3.1 File: haproxy.con  
$ModLoad imudp  
$UDPServerAddress 127.0.0.1  
$UDPServerRun 514  
local2.* /var/haproxy-traffic.log  
local2.notice /var/log/haproxy-admin.log
```

Figure 34: Configuration du serveur syslog

Ensuite nous redémarrons le serveur syslog :

```
systemctl restart rsyslog
```

### ➤ Configurer le frontend et les backends HAProxy

La configuration suivante définit une section pour accéder à la page HAProxy Stats, aux serveurs frontaux et principaux.

Ici, nous implémentons un équilibreur de charge basé sur l'algorithme least connections.

```
nano /etc/haproxy/haproxy.cfg
```

```
listen stats
  bind 70.37.111.91:9000
  mode http
  option forwardfor
  option httpclose
  stats enable
  stats uri /stats
  stats refresh 5s
  stats hide-version
  stats show-legends
  stats realm Haproxy\ Statistics
  stats auth admin:123

frontend TT
  bind 70.37.111.91:80
  mode tcp
  option tcplog
  option contstats
  option tcpka
  default_backend moodle_backend

backend moodle_backend
  balance leastconn
  mode tcp
  option tcpka
  option srvtcpka
  server moodle1 40.74.228.156:80 check
  server moodle2 40.74.249.86:80 check
```

Figure 35: Configuration du HAProxy

Maintenant, nous redémarrons le serveur HAProxy pour que la configuration soit appliquée :

```
systemctl restart rsyslog
```

## ➤ Configure Firewall

Nous devons maintenant ouvrir les ports 80, 443 et 9000 :

```
firewall-cmd --zone=public --permanent --add-service=http
firewall-cmd --zone=public --permanent --add-service=https
firewall-cmd --zone=public --permanent --add-port=9000/tcp
firewall-cmd --reload
```

## ➤ Test du fonctionnement du HAProxy

Le serveur HAProxy enverra la demande aux deux serveurs Web en se base sur l'algorithme utilisé « leastconne ». Nous pouvons tester cela en rechargeant la page Web plusieurs fois :

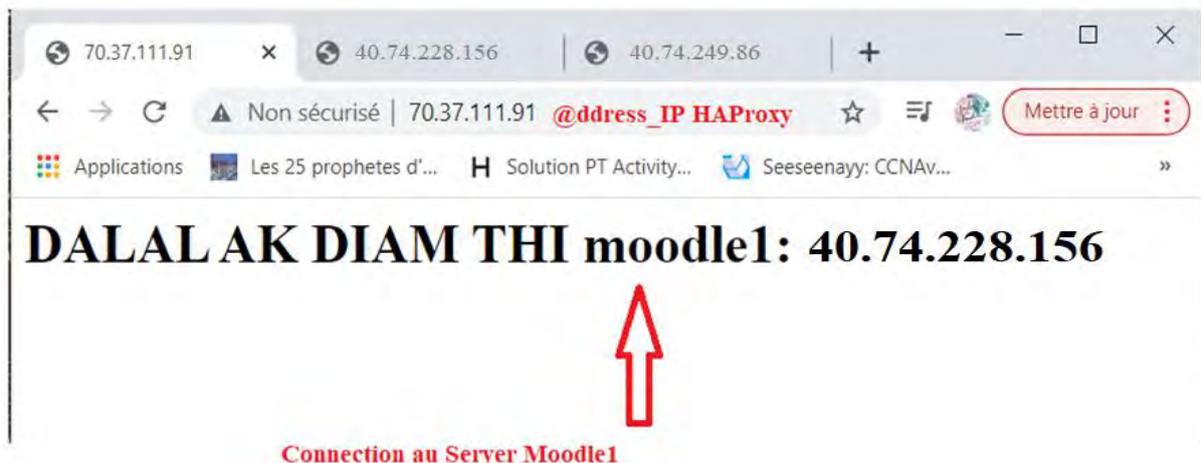


Figure 36: Connection au serveur web\_1 depuis le HAProxy



Figure 37: Connection au serveur web\_2 depuis le HAProxy

Après avoir installé notre HAProxy, nous allons déployer notre cluster de serveurs web composé de deux serveurs : « moodle1 » et « moodle2 ».

### III.7.2.4. Installation du cluster Moodle

Pour les deux serveurs nous aurons besoin de :

- ✓ Système d'exploitation : Ubuntu Server 18.04 LTS
- ✓ RAM : 4GiB
- ✓ CPU : 2
- ✓ Disque : 100GiB
- ✓ MariaDB : 10.3
- ✓ PHP : 7.3
- ✓ Apache2

#### ➤ Installation de Apache2

Moodle a besoin d'un serveur Web et le serveur Web le plus utilisé aujourd'hui est apache2.

```
apt update
apt-get install apache2
```

Après avoir installé Apache2, exécutez les commandes ci-dessous pour désactiver la liste des répertoires.

```
sudo sed -i "s/Options Indexes FollowSymLinks/Options FollowSymLinks/"
/etc/apache2/apache2.conf
```

Ensuite, exécutons les commandes ci-dessous pour arrêter, démarrer et permettre au service Apache2 de toujours démarrer avec le serveur démarre.

```
systemctl stop apache2.service
systemctl start apache2.service
systemctl enable apache2.service
```

#### ➤ Installation du serveur de base de données MariaDB

Moodle a besoin d'un serveur de base de données et le serveur de base de données MariaDB est un excellent point de départ. Pour l'installer, exécutons les commandes ci-dessous.

```
sudo apt-get install mariadb-server mariadb-client
```

Après l'installation, les commandes ci-dessous peuvent être utilisées pour arrêter, démarrer et activer le service MariaDB pour qu'il démarre toujours lorsque le serveur démarre.

Après cela, exécutons les commandes ci-dessous pour sécuriser le serveur MariaDB.

```
sudo mysql_secure_installation
```

Nous allons éditer le fichier de configuration par défaut de MariaDB en exécutant les commandes ci-dessous :

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Ajoutez ensuite les lignes ci-dessous juste en dessous de la section [mysqld].

```
default_storage_engine = innodb
innodb_file_per_table = 1
innodb_file_format = Barracuda
innodb_large_prefix = 1
```

Enregistrons pour redémarrer le serveur MariaDB

### ➤ **Installation de PHP 7.3**

PHP 7.3 peut ne pas être disponible dans les référentiels par défaut d'Ubuntu... pour l'installer, nous devons l'obtenir à partir de référentiels tiers.

Exécutons les commandes ci-dessous pour ajouter le référentiel tiers ci-dessous pour mettre à niveau vers PHP 7.3

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:ondrej/php
```

Ensuite, mettons à jour et mettez à niveau vers PHP 7.3

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php7.3 libapache2-mod-php7.3 php7.3-common php7.3-mbstring php7.3-xmlrpc php7.3-soap php7.3-gd php7.3-xml php7.3-intl php7.3-mysql php7.3-cli php7.3-mcrypt php7.3-ldap php7.3-zip php7.3-curl
```

Après avoir installé PHP, Ouvrons le fichier php.ini et renseignons les paramètres ci-dessous :

```
sudo nano /etc/php/7.3/apache2/php.ini
file_uploads = On
allow_url_fopen = On
memory_limit = 256M
upload_max_filesize = 100M
max_execution_time = 360
cgi.fix_pathinfo = 0
date.timezone = America/Chicago
```

### ➤ Création d'une base de données pour Moodle

Maintenant que nous avons installé tous les packages requis, pour commencer à configurer les serveurs. Nous allons utiliser la base de données crée dans le cluster et créer un utilisateur de la base de données. Cet utilisateur sera configuré dans Moodle pour exécuter des lectures. Le serveur ProxySQL pourra envoyer tout le trafic exécuté depuis cet utilisateur vers les nœuds du cluster.

```
retel@galera3: ~
MariaDB [(none)]> CREATE USER 'moodle'@'40.74.228.156' IDENTIFIED BY 'moodle';

MariaDB [(none)]> GRANT ALL ON galera.* TO 'moodle'@'40.74.228.156' IDENTIFIED BY 'moodle' WITH GRANT OPTION;

MariaDB [(none)]> FLUSH PRIVILEGES;

MariaDB [(none)]> SELECT user, host FROM mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| proxysql     | %             |
| root         | 127.0.0.1    |
| moodle       | 40.74.228.156 |
+-----+-----+
```

**Figure 38: Utilisateur de la base de données du serveur Moodle**

### ➤ Téléchargement de la dernière version de Moodle

Ensuite, nous allons exécuter les commandes ci-dessous pour télécharger la dernière version de Moodle et extraire le fichier téléchargé vers la racine par défaut d'Apache2.

```
cd /tmp && wget https://download.moodle.org/download.php/direct/stable39/moodle-latest-39.tgz
tar -zxvf moodle-latest-39.tgz
sudo mv moodle /var/www/html/moodle
sudo mkdir /var/www/html/moodledata
```

Modifions l'autorisation du répertoire. Puis redémarrer le serveur apache2.

```
sudo chown -R www-data:www-data /var/www/html/moodle/
sudo chmod -R 755 /var/www/html/moodle/
sudo chown www-data /var/www/html/moodledata
systemctl restart apache2.service
```

Après cela, ouvrons notre navigateur et accédons avec l'adresse IP ou le nom d'hôte du serveur ; nous verrons la page de l'assistant de configuration par défaut de Moodle :

Non sécurisé 40.74.249.86/moodle/install.php

# Installation

Language

Choose a language

Please choose a language for the installation. This language will also be used as the default language for the site, though it may be changed later.

Language English (en)

Next >>

**Figure 39: Choix de la langue de configuration**

Après avoir choisi le langage de configuration, nous allons cliquer sur suivante pour passer à la deuxième étape :

Le chemin d'accès complet au dossier contenant le code source de Moodle.

**Dossier de données**  
Un dossier dans lequel Moodle stockera tous les fichiers qui seront déposés par les utilisateurs.  
Ce dossier doit être accessible en lecture et en écriture par l'utilisateur système du serveur web (habituellement nommé « www-data », « nobody » ou encore « apache »).  
Il ne doit pas être accessible directement via le web.  
Si ce dossier n'existe pas encore, Moodle tentera de le créer au cours du processus d'installation.

Adresse web:

Dossier Moodle:

Dossier de données:

« Précédent » **Suivant »**

Figure 40: Configuration du dossier Moodledata

Après d'avoir cliqué sur **suivant**, il nous sera demandé de choisir le type de base de données : dans notre cas nous allons choisir MariaDB :

**Installation**

Base de données

Sélectionner un pilote de base de données

Moodle supporte plusieurs types de serveurs de base de données. Veuillez contacter l'administrateur du serveur si vous ne savez pas quel type utiliser.

Type: 

- MariaDB (native/mariadb)
- MySQL amélioré (natif, mysqli)
- Aurora MySQL (native/auroramysql)
- MariaDB (native/mariadb)**
- Non disponible**
- PostgreSQL (natif, pgsqll)
- Oracle (natif, oci)
- SQL\*Server Microsoft (native/sqldr)

Figure 41: Choix du type de base de données

Nous allons ensuite renseigner l'adresse IP du serveur de base de données, son utilisateur et son mot de passe, voir la capture ci-dessous :

Non sécurisé | 40.74.249.86/moodle/install.php

Les 25 prophètes d'... Solution PT Activity... Seeseenay: CCNAv... Microsoft Word - r...

Serveur de base de données: 70.37.69.247

Nom de la base de données: galera

Utilisateur de la base de données: moodle

Mot de passe de la base de données: moodle

Préfixe des tables: mdl\_

Port de la base de données:

Socket Unix:

« Précédent Suivant »

Figure 42: Enregistrement de la base de données de Moodle

En fin, nous passons maintenant à l'étape de la vérification des modules requis, voir capture ci-dessous :

Plugin	Statut
php_extension json	OK
php_extension hash	OK
php_extension fileinfo	OK
php_setting memory_limit	OK
php_setting file_uploads	OK
php_setting opcache.enable	OK

### Autres vérifications

Information	Rapport	Plugin	Statut
site not https	L'échec de ce test indique un problème potentiel Votre site n'est pas sécurisé au moyen de HTTPS. Il est vivement recommandé de passer votre site en HTTPS pour une sécurité accrue et une meilleure intégration à d'autres systèmes.		Vérifier
mysql_full_unicode_support	L'échec de ce test indique un problème potentiel La configuration actuelle de MySQL ou de MariaDB utilise le réglage « utf8 ». Cet encodage de caractères ne supporte pas les caractères encodés sur 4 octets, comme certains emoji. L'utilisation de ces caractères provoquera une erreur et les informations envoyées à la base de données seront perdues. Veuillez modifier votre réglage à « utf8mb4 ». Consultez la documentation pour plus de détails.		Vérifier

Votre environnement serveur possède les modules requis.

Continuer

Figure 43: Vérification des modules d'installation de Moodle

A présent, nous pouvons créer un administrateur de notre plateforme Moodle comme le montre la figure ci-dessous :

► Tout déplier

▼ Général

Nom d'utilisateur ? admin

Choisir une méthode d'authentification ? Comptes manuels

Le mot de passe doit comporter au moins 8 caractère(s), au moins 1 chiffre(s), au moins 1 minuscule(s), au moins 1 majuscule(s), au moins 1 caractère(s) non-alphanumérique(s) tels que \*, - ou #

Nouveau mot de passe ! ? Fatou@123  Imposer le changement du mot de passe ?

Prénom ! Admin

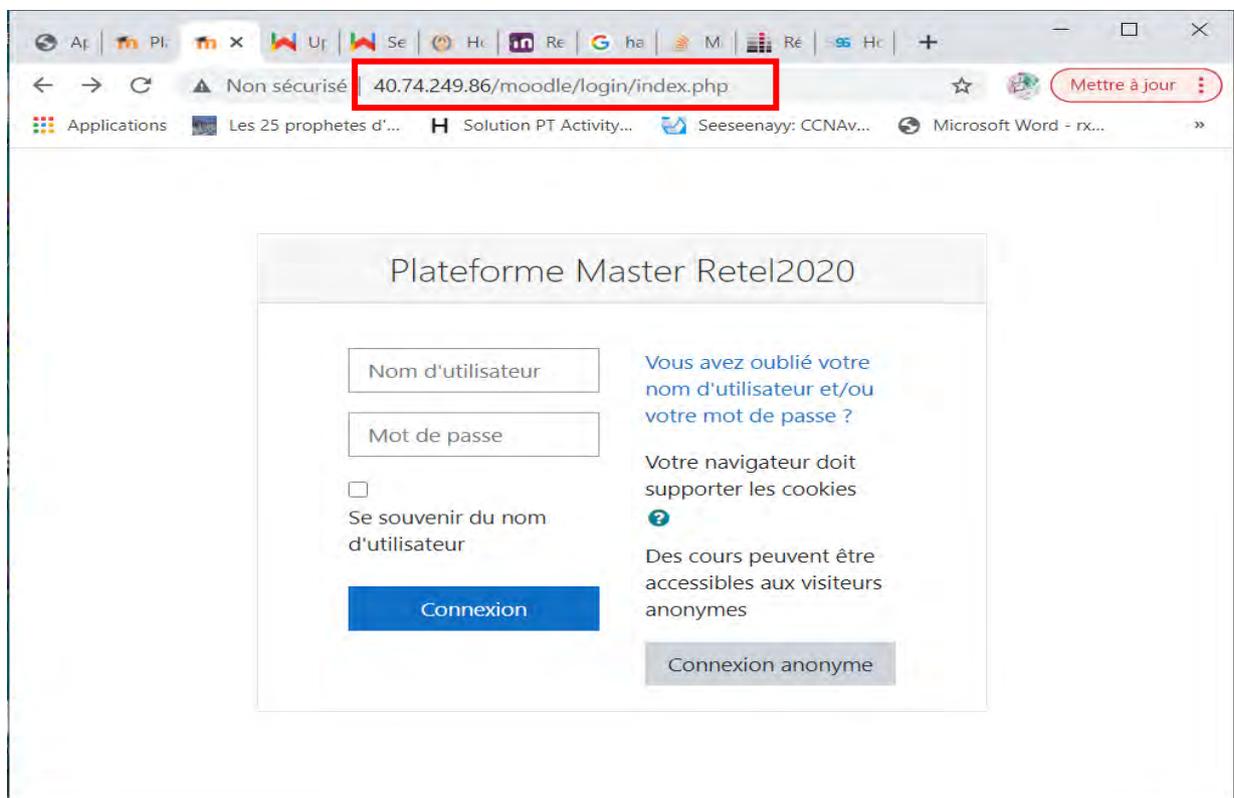
Nom ! Utilisateur

Adresse de courriel !

Affichage de l'adresse de courriel ? Autoriser tout le monde à voir mon adresse de courriel ▾

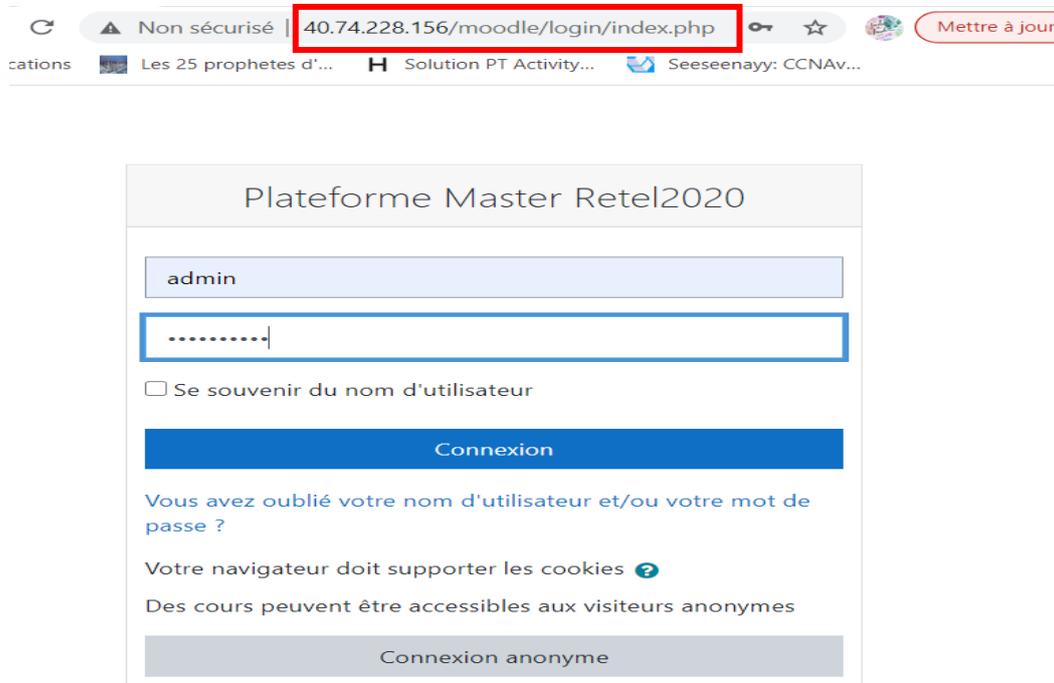
**Figure 44: Création d'un administrateur du site Moodle**

Après avoir renseigné toutes les informations nécessaires, nous allons enregistrer. Maintenant notre site est prêt à utiliser :



**Figure 45: Interface de connexion du site**

Pour le déploiement du deuxième serveur la même procédure d'installation du premier serveur sera faite. Après les étapes d'installation nous obtenons directement la page de connexion du site Moodle car n'oublions pas que les deux serveurs pointent sur le même cluster de base de données et sur le même dossier moodledata ; voir la figure ci-après :



Plateforme Master Retel2020

admin

.....

Se souvenir du nom d'utilisateur

Connexion

[Vous avez oublié votre nom d'utilisateur et/ou votre mot de passe ?](#)

Votre navigateur doit supporter les cookies ?

Des cours peuvent être accessibles aux visiteurs anonymes

Connexion anonyme

Figure 46: Interface de connexion du site (2)

### III.7.3. Implémentation du système BigBlueButton

Dans cette partie, nous allons implémenter les composants du système BigBlueButton, à savoir le dispositif Scalelite et le pool de serveurs BigBlueButton.

#### III.7.3.1. Installation du pool BigBlueButton

Pour mettre en place une installation réussie de BigBlueButton, il nous est recommandé de commencer avec un serveur Ubuntu 16.04 64 bits « propre » dédié à BigBlueButton.

Prérequis :

- ✓ Ubuntu 16.04 OS 64 bits exécutant le noyau Linux 4.x
- ✓ 16 Go de mémoire avec swap activé
- ✓ 4 cœurs de processeur, avec des performances mono-thread élevées
- ✓ 100 Go d'espace disque libre
- ✓ Les ports TCP 80 et 443 sont accessibles
- ✓ Les ports UDP 16384 - 32768 sont accessibles
- ✓ Bande passante de 250 Mbits/s (symétrique) ou plus

- ✓ Les ports TCP 80 et 443 ne sont pas utilisés par une autre application Web ou un proxy inverse
- ✓ Un nom d'hôte (tel que bbb1.example.com) pour la configuration d'un certificat SSL
- ✓ Adresse IPV4 et IPV6

Nous avons un serveur Ubuntu 16.04 64 bits prêt pour l'installation. Mais avant de nous lancer dans l'installation, effectuons quelques vérifications de configuration rapides pour nous assurer que notre serveur répond aux exigences minimales.

Tout d'abord, vérifions que les paramètres régionaux du serveur sont **en\_US.UTF-8**.

```
cat /etc/default/locale
```

```
root@bbb-drtp: ~
root@bbb-drtp:~#
root@bbb-drtp:~# cat /etc/default/locale
# File generated by update-locale
LANG=en_US.UTF-8
root@bbb-drtp:~#
```



**Figure 47: Paramètres régionaux du serveur**

**Notons bien :** Si nous ne voyions pas `LANG="en_US.UTF-8"`, nous allions entrer les commandes suivantes pour définir le local sur `en_US.UTF-8`

```
apt-get install -y language-pack-en
update-locale LANG=en_US.UTF-8
```

Vérifions que notre serveur dispose (au moins) de 4G de mémoire à l'aide de la commande **free -h**. Voici la sortie de l'un de nos serveurs de test :

```
root@bbb-drtp: ~
root@bbb-drtp:~# free -h
```

	total	used	free	shared	buff/cache	available
Mem:	15G	2.8G	7.1G	101M	5.8G	12G
Swap:	0B	0B	0B			

```
root@bbb-drtp:~#
```



**Figure 48: Vérification de la mémoire disponible**

Ensuite, vérifions que le serveur a la version d'Ubuntu 16.04 :

```
root@bbb-drtp:~# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.7 LTS"
root@bbb-drtp:~#
```

**Figure 49: Vérification de la version d'Ubuntu**

Ensuite, vérifions que notre serveur a (au moins) 4 cœurs de processeur

```
root@bbb-drtp:~#
root@bbb-drtp:~# cat /proc/cpuinfo | awk '/^processor/{print $3}' | wc -l
4
root@bbb-drtp:~#
```

**Figure 50: Vérification du nombre cœur du serveur**

Nous devons disposer d'un nom de domaine complet (FQDN) par exemple bbb1.example.com, d'un certificat SSL, avant que BigBlueButton ne s'exécute.

Pour obtenir un nom de domaine, il existe de nombreux registraires de noms de domaine tels que : GoDaddy et Network Solutions.

Mise à niveau à partir de BigBlueButton 2.2

```
rm -f /etc/apt/sources.list.d/jonathonf-ubuntu-ffmpeg-4-xenial.list
add-apt-repository ppa:bigbluebutton/support
add-apt-repository ppa:libreoffice/ppa
apt-get update
apt-get dist-upgrade
```

Après la mise à niveau, nous devons faire une **bbb-conf --setip** pour réattribuer l'adresse IP ou le nom d'hôte de vos serveurs aux fichiers de configuration de BigBlueButton.

```
bbb-conf --setip <IP_or_hostname>
bbb-conf --check
```

```
root@bbb-drtp: ~  
root@bbb-drtp:~#  
root@bbb-drtp:~# bbb-conf --check  
BigBlueButton Server 2.2.36 (2416)  
Kernel version: 4.15.0-1111  
Distribution: Ubuntu 16.04.7 LTS (64-bit)  
Memory: 16426 MB  
CPU cores: 4
```

**Figure 51: Information du serveur BigBlueButton**

### ➤ Installation de BigBlueButton

Nous devons d'abord vérifier les dépôts dans le fichier `/etc/apt/sources.list` :

```
root@bbb-drtp:~# grep "multiverse" /etc/apt/sources.list
```

```
root@bbb-drtp: ~  
root@bbb-drtp:~# grep "multiverse" /etc/apt/sources.list  
## multiverse WILL NOT receive any review or updates from the Ubuntu  
deb http://azure.archive.ubuntu.com/ubuntu/ xenial multiverse  
# deb-src http://azure.archive.ubuntu.com/ubuntu/ xenial multiverse  
deb http://azure.archive.ubuntu.com/ubuntu/ xenial-updates multiverse  
# deb-src http://azure.archive.ubuntu.com/ubuntu/ xenial-updates multiverse
```

**Figure 52: Vérification des dépôts**

Pour donner beaucoup d'entropie à la machine virtuelle, nous allons installer le package appeler **haveged** (un simple démon d'entropie) :

```
apt-get install haveged  
add-apt-repository ppa:bigbluebutton/support -y  
add-apt-repository ppa:rmescandon/yq -y  
add-apt-repository ppa:libreoffice/ppa  
apt-get update  
apt-get dist-upgrade
```

Le client HTML5 de BigBlueButton utilise MongoDB, une base de données très efficace utilisée pour synchroniser l'état des clients. Pour installer MongoDB, procédez comme suit :

```
wget -qO - https://www.mongodb.org/static/pgp/server-3.4.asc | sudo apt-key add -  
echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4  
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list  
apt-get update  
apt-get install -y mongodb-org curl
```

### ➤ Installation de la clé apt-get pour le référentiel BigBlueButton

Avant d'installer BigBlueButton, nous devons ajouter la clé publique du projet à la chaîne de clés de notre serveur en entrant la commande suivante :

```
wget https://ubuntu.bigbluebutton.org/repo/bigbluebutton.asc -O- | sudo apt-key add -
```

Ensuite, notre serveur doit savoir où télécharger les packages BigBlueButton 2.2. Pour configurer le référentiel de packages, exécutons la commande suivante :

```
echo "deb https://ubuntu.bigbluebutton.org/xenial-22/ bigbluebutton-xenial main" | sudo tee  
/etc/apt/sources.list.d/bigbluebutton.list  
  
apt-get install apt-transport-https  
  
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 6657DBE0CC86BB64  
apt-get update
```

Entrez les deux commandes suivantes pour installer BigBlueButton.

```
root@bbb-drtp:~# sudo apt-get install bigbluebutton  
root@bbb-drtp:~# sudo apt-get install bbb-html5
```

Une fois l'installation terminée, nous pouvons faire de HTML5 le client par défaut (recommandé), puis redémarrons notre BigBlueButton :

```
bbb-conf --restart
```

```

root@bbb-drtp: ~
root@bbb-drtp:~#
root@bbb-drtp:~# sudo bbb-conf --restart
Restarting BigBlueButton 2.2.36 ...
Stopping BigBlueButton

Applying updates in /etc/bigbluebutton/bbb-conf/apply-config.sh:
- Enable Firewall and opening 22/tcp, 80/tcp, 443/tcp and 16384:32768/udp
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
Firewall is active and enabled on system startup

Starting BigBlueButton
.....

```



**Figure 53: Redémarrage du serveur**

Nous pouvons également utiliser `bbb-conf --status` pour vérifier que tous les processus BigBlueButton ont démarré et sont en cours d'exécution.

```
bbb-conf --status
```

```

root@bbb-drtp:~#
root@bbb-drtp:~# sudo bbb-conf --status
nginx _____> [✓ - active]
freeswitch _____> [✓ - active]
redis-server _____> [✓ - active]
bbb-apps-akka _____> [✓ - active]
bbb-transcode-akka _____> [✓ - active]
bbb-fsesl-akka _____> [✓ - active]
red5 _____> [✓ - active]
tomcat7 _____> [✓ - active]
mongod _____> [✓ - active]
bbb-html5 _____> [✓ - active]
bbb-webrtc-sfu _____> [✓ - active]
kurento-media-server _____> [✓ - active]
etherpad _____> [✓ - active]
bbb-web _____> [✓ - active]
root@bbb-drtp:~# █

```

**Figure 54: Statut du serveur BigBlueButton**

Après avoir installé un certificat SSL, nous pouvons intégrer BigBlueButton à l'une des intégrations tierces en fournissant à l'intégration l'adresse du serveur et le secret partagé. Nous pouvons utiliser la commande **bbb-conf** pour afficher ces informations en utilisant la commande suivante :

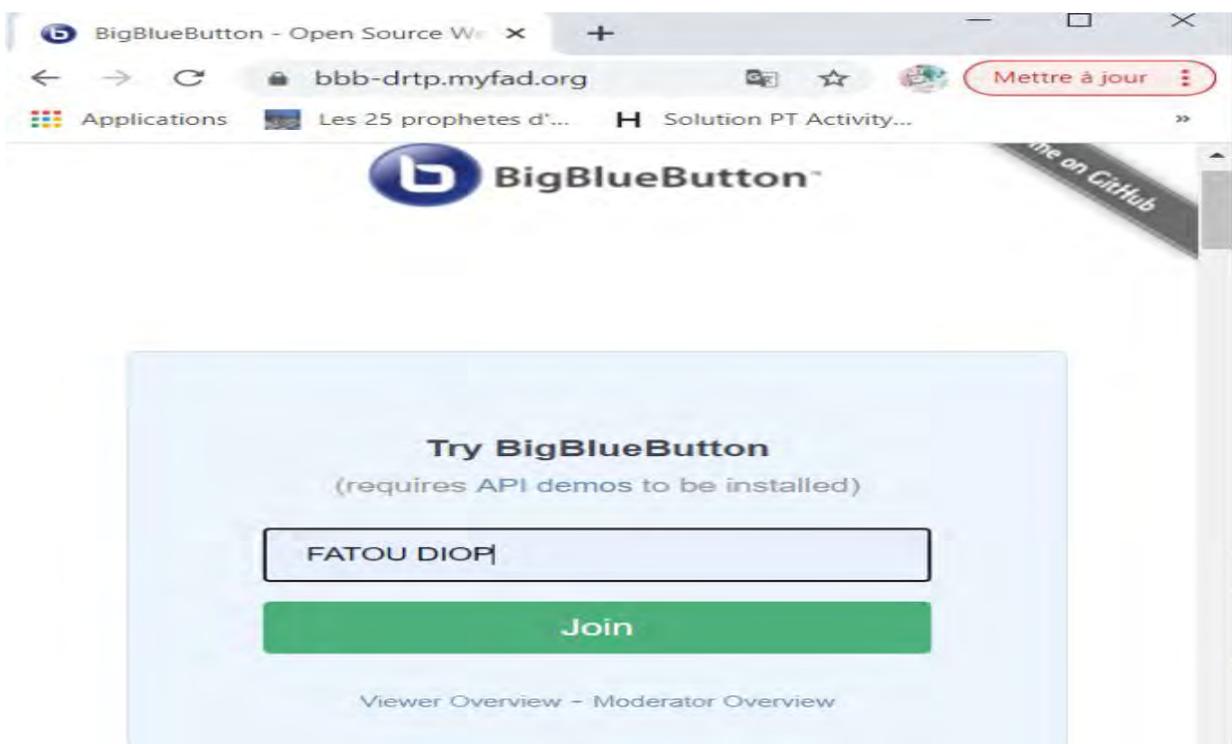
```
root@bbb-drtp:~# sudo bbb-conf --secret
URL: https://bbb-drtp.myfad.org/bigbluebutton/
Secret: 7PL30HrVZpw8CPRA4yf2DGaf9QqwofTchmY0kEKqo

Link to the API-Mate:
https://mconf.github.io/api-mate/#server=https://bbb-drtp.myfad.org/bigbluebutton/&sharedSecret=7PL30HrVZpw8CPRA4yf2DGaf9QqwofTchmY0kEKqo

root@bbb-drtp:~#
```

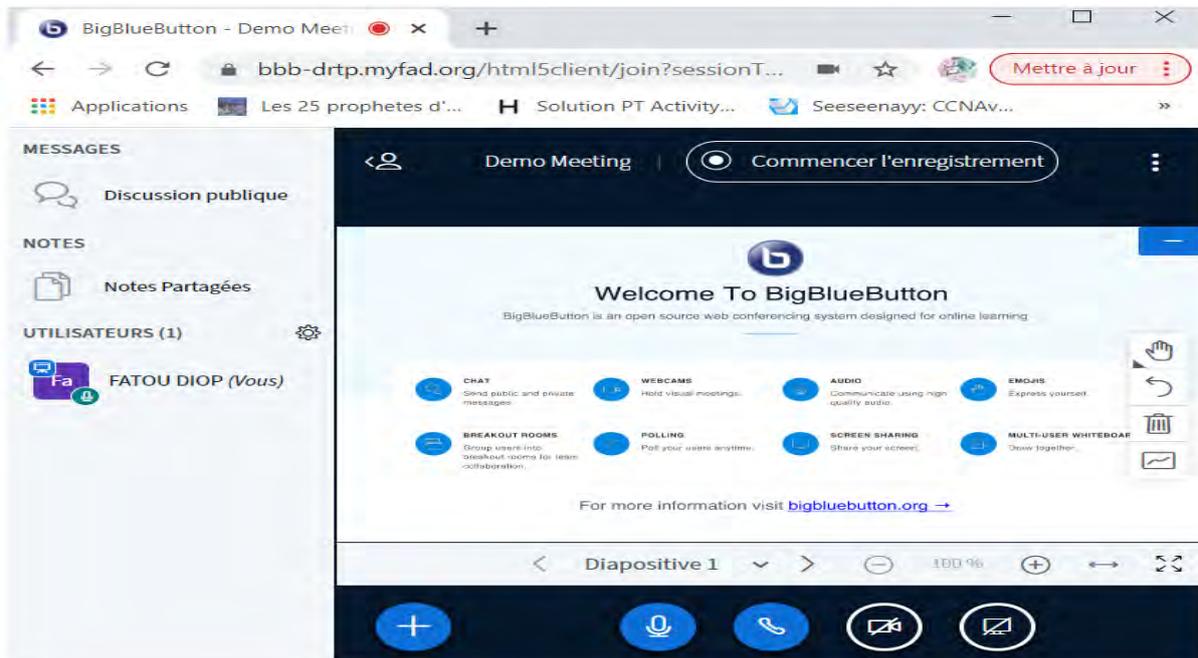
**Figure 55: Information du serveur BigBlueButton**

Pour accéder à l'interface tapons le nom de domaine sur notre navigateur :



**Figure 56: Interface de connexion au serveur BigBlueButton**

Maintenant nous pouvons accéder au meeting en tant qu'utilisateur Fatou DIOP, voir la figure ci-dessous :



**Figure 57: Connexion à un meeting**

A présent nous venons d'installer un serveur de BigBlueButton.

Le même processus doit se faire avec les deux autres serveurs pour terminer la création du pool de serveurs BigBlueButton.

Nous venons de déployer trois serveurs BigBlueButton de pas à pas. Maintenant nous allons passer l'installation de Scalelite afin d'y insérer nos serveurs BigBlueButton.

### III.7.3.2. Installation de Scalelite

Le processus d'installation de Scalelite nécessite des connaissances techniques que nous allons présenter dans les lignes qui suivent.

Prérequis :

- ✓ 4 cœurs de processeur
- ✓ Mémoire (RAM) : 8 GiB
- ✓ Disk de stockage 100GiB
- ✓ Systèmes d'exploitation : Ubuntu Server 18.04 LTS
- ✓ Docker

Avant de commencer nous allons vérifier la version d'OS requise, voir figure ci-dessous :

```
root@scalelite:~# lsb_release
No LSB modules are available.
root@scalelite:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.5 LTS
Release:        18.04
Codename:       bionic
root@scalelite:~#
```

**Figure 58: Vérification de l'OS du serveur**

Maintenant faisons une mise à jour de la machine virtuelle :

```
apt-get update
apt-get dist-upgrade
```

Ajoutons ensuite la mémoire d'échange :

```
fallocate -l 1G /swapfile
dd if=/dev/zero of=/swapfile bs=1024 count=1048576
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
```

Pour rendre le changement permanent, ajoutons la ligne suivante :

```
/swapfile swap swap defaults 0 0
```

### ➤ Installation de docker

Docker est un progiciel de plus en plus populaire qui crée un conteneur pour le développement d'applications. Le développement dans Docker accélère les applications, car il partage le noyau et d'autres ressources, au lieu de nécessiter des ressources de serveur dédié et l'installation se déroule en plusieurs étapes.

**Étape 1** : Mettre à jour les référentiels de logiciels

```
apt-get update
```

## Étape 2 : Désinstaller les anciennes versions de Docker

Il est recommandé de désinstaller tout ancien logiciel Docker avant de continuer en utilisant la commande suivante :

```
apt-get remove docker docker-engine docker.io
```

## Étape 3 : Téléchargez les dépendances

Nous devons exécuter ces commandes pour permettre à notre système d'exploitation d'accéder aux référentiels Docker via HTTPS :

```
apt install apt-transport-https ca-certificates curl software-properties-common
```

Pour clarifier, voici une brève description de chaque commande :

- ✓ apt-transport-https : permet au gestionnaire de packages de transférer des fichiers et des données via https
- ✓ ca-certificates : permet au système (et au navigateur Web) de vérifier les certificats de sécurité
- ✓ curl : il s'agit d'un outil de transfert de données
- ✓ software-properties-common : ajoute des scripts pour gérer les logiciels

## Étape 4 : Ajouter la clé GPG de Docker

La clé GPG est une fonction de sécurité. Pour nous assurer que le logiciel que nous avons installé est authentique, exécutons la commande ci-après :

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Nous pouvons voir que l'exécution de la commande c'est bien passé, voir la capture ci-après :



```
root@scalelite:~#  
root@scalelite:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key  
OK  
root@scalelite:~#
```

**Figure 59: Ajout de la clé CPG de docker**

## Étape 5 : Installation du référentiel Docker

Pour installer le référentiel Docker, exécutons la commande suivante :

```
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
```

## Étape 6 : Mettre à jour les référentiels

Mettons à jour les référentiels que nous avons ajoutés :

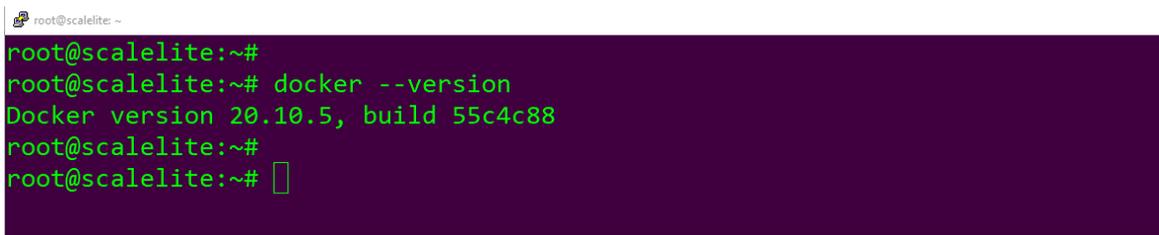
```
apt-get update
```

## Étape 7 : Installation de la dernière version de Docker

```
apt-get install docker-ce
```

À ce stade, exécutons la commande pour voir que notre docker est installé avec sa dernière version.

```
root@scalelite:~# docker --version
```



```
root@scalelite:~#  
root@scalelite:~# docker --version  
Docker version 20.10.5, build 55c4c88  
root@scalelite:~#  
root@scalelite:~#
```

Figure 60: Vérification de la version du docker

### ➤ Installation de docker Compose

Pour télécharger la dernière version de Docker Compose, utilisons la commande :

```
curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

L'option « -L » indique au système de suivre toutes les redirections, au cas où le fichier aurait été déplacé.

L'option « -o » pour changer le nom du fichier.

Le fichier sera enregistré dans **/usr/local/bin/**

Ensuite, modifions les autorisations de fichier pour permettre au nouveau logiciel d'être exécuté sur Ubuntu :

```
chmod +x /usr/local/bin/docker-compose
```

## ➤ Installation de Scalelite

Nous allons d'abord récupérer le script d'installation.

```
git clone https://github.com/jfederico/scalelite-run
cd scalelite-run
```

```
root@scalelite:~#
root@scalelite:~# git clone https://github.com/jfederico/scalelite-run
Cloning into 'scalelite-run'...
remote: Enumerating objects: 41, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 260 (delta 20), reused 27 (delta 9), pack-reused 219
Receiving objects: 100% (260/260), 69.05 KiB | 1.97 MiB/s, done.
Resolving deltas: 100% (128/128), done.
root@scalelite:~# cd scalelite-run
```

Figure 61: Téléchargement de Scalelite

## ➤ Initialisation des variables d'environnement

Créons un nouveau fichier « **.env** » basé sur le fichier **dotenv** inclus.

```
cp dotenv .env
```

La plupart des variables requises sont prédéfinies par défaut, celles qui doivent être définies avant de commencer sont :

```
SECRET_KEY_BASE=
LOADBALANCER_SECRET=
URL_HOST=
NGINX_SSL=
```

Nous allons à présent chercher ces éléments. Pour ce faire nous exécutons les commandes suivantes :

- ✓ Obtenons la valeur de **SECRET\_KEY\_BASE** avec :

```
openssl rand -hex 64
```

```
root@scalelite:~/scalelite-run#
root@scalelite:~/scalelite-run# openssl rand -hex 64
5f940a0cefcdfef789d697cc978b546adb238cc28f285e87462dd41efaf1d1b4ab1e52245a28405665cd4f5110b830a68bf
root@scalelite:~/scalelite-run#
```

Figure 62: Obtention de clé secret

✓ Obtenons la valeur de `LOADBALANCER_SECRET` avec :

```
openssl rand -hex 24
```

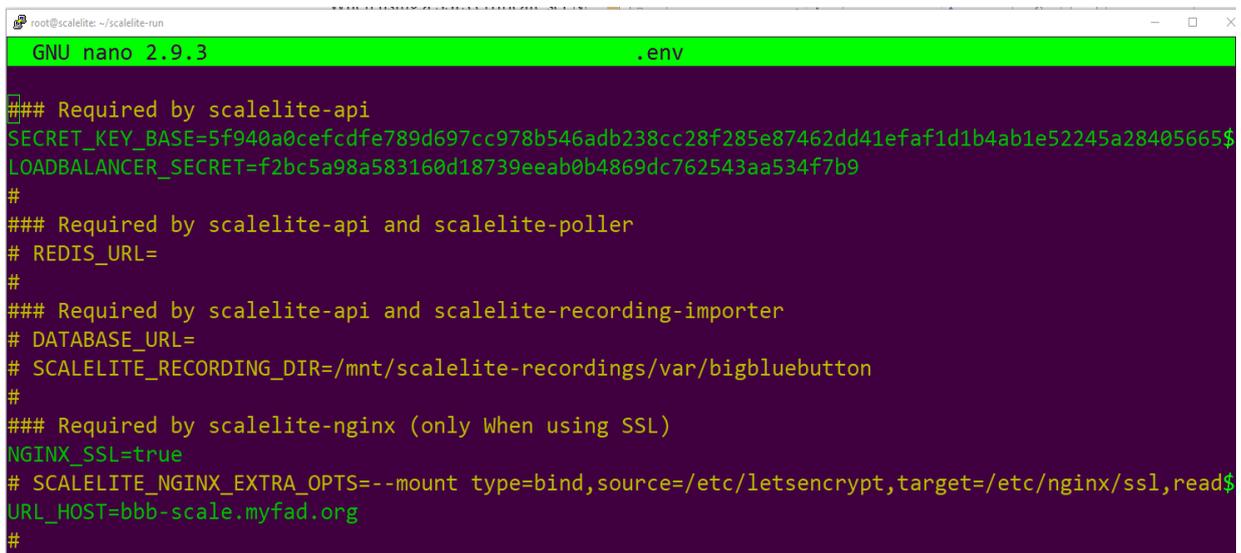
Nous devrions voir quelque chose comme ceci :

```
root@scalelite:~/scalelite-run#  
root@scalelite:~/scalelite-run# openssl rand -hex 64  
5f940a0cefcdfef789d697cc978b546adb238cc28f285e87462dd41efaf1d1b4ab1e52245a28405665cd4f5110b830a68bf  
root@scalelite:~/scalelite-run#  
root@scalelite:~/scalelite-run# openssl rand -hex 24  
f2bc5a98a583160d18739eeab0b4869dc762543aa534f7b9
```

**Figure 63: Obtention de la valeur du Loadbalancer**

Définissons maintenant le nom d'hôte sur `URL_HOST` (par exemple pour notre cas nous avons utilisé `bbb-scale.myfad.org`).

Notre fichier « `.env` » final devrait ressembler à ceci :



```
GNU nano 2.9.3 .env  
### Required by scalelite-api  
SECRET_KEY_BASE=5f940a0cefcdfef789d697cc978b546adb238cc28f285e87462dd41efaf1d1b4ab1e52245a28405665cd4f5110b830a68bf  
LOADBALANCER_SECRET=f2bc5a98a583160d18739eeab0b4869dc762543aa534f7b9  
#  
### Required by scalelite-api and scalelite-poller  
# REDIS_URL=  
#  
### Required by scalelite-api and scalelite-recording-importer  
# DATABASE_URL=  
# SCALELITE_RECORDING_DIR=/mnt/scalelite-recordings/var/bigbluebutton  
#  
### Required by scalelite-nginx (only When using SSL)  
NGINX_SSL=true  
# SCALELITE_NGINX_EXTRA_OPTS="--mount type=bind,source=/etc/letsencrypt,target=/etc/nginx/ssl,readonly"   
URL_HOST=bbb-scale.myfad.org  
#
```

**Figure 64: Contenu du fichier `.env`**

Pour utiliser un certificat SSL signé par Let's Encrypt, générons les certificats :

```
./init-letsencrypt.sh
```

Démarrons les services :

```
Docker-compose up -d
```

Maintenant, le serveur Scalelite est en cours d'exécution, mais il n'est pas encore tout à fait prêt. La base de données doit être initialisée :

```
docker exec -i scalelite-api bundle exec rake db:setup
```

Il est important de mentionner que Scalelite n'a pas d'interface utilisateur pour l'administration. Il est livré avec un ensemble de scripts backend (tâches de rake) qui doivent être exécutés en ligne de commande pour configurer et gérer les serveurs BigBlueButton qui seront utilisés.

Nous pouvons vérifier l'état du serveur avec les commandes suivantes :

```
docker exec -i scalelite-api bundle exec rake status
docker exec -i scalelite-api bundle exec rake servers
```

Nous devrions remarquer qu'aucun serveur n'est configuré. Cela peut également être vérifié en extrayant la liste des serveurs :

```
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake servers
No servers are configured
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake servers
No servers are configured
root@scalelite:~/scalelite-run#
root@scalelite:~/scalelite-run#
```

**Figure 65: Liste des serveurs BigBlueButton**

Maintenant nous pouvons ajouter nos serveurs BigBlueButton en adoptant ce modèle :

```
docker exec -i scalelite-api bundle exec rake
servers:add[https://bbb1.example.com/bigbluebutton/api/,bbb-secret]
```

Nous allons ajouter notre premier BigBlueButton

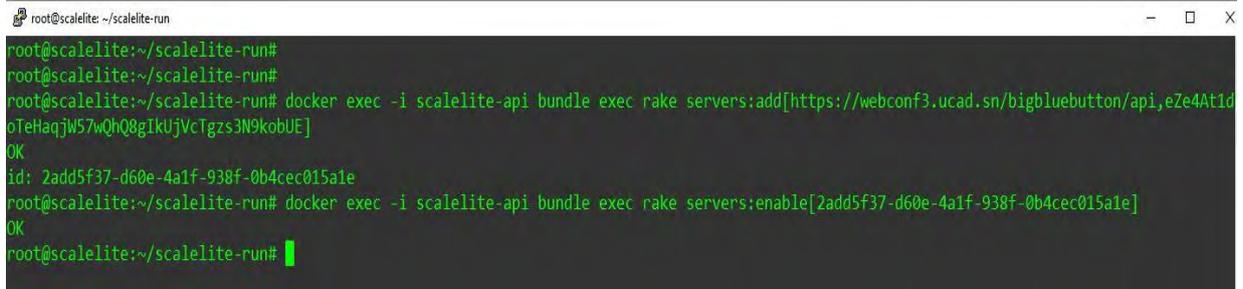
```
docker exec -i scalelite-api bundle exec rake
servers:add[https://webconf3.ucad.sn/bigbluebutton/api,eZe4At1doTeHaqjW57wQhQ8gIkUjV
cTgzs3N9kobUE]
```

Cela devrait nous donner un identifiant pour le serveur ajouté :

```
OK
id: 2add5f37-d60e-4a1f-938f-0b4cec015a1e
root@scalelite:~/scalelite-run#
```

Utilisez l'ID du serveur pour activer le serveur :

```
docker exec -i scalelite-api bundle exec rake servers:enable[2add5f37-d60e-4a1f-938f-0b4cec015a1e]
```

A screenshot of a terminal window titled 'root@scalelite: ~/scalelite-run'. The terminal shows the following commands and output:

```
root@scalelite:~/scalelite-run#  
root@scalelite:~/scalelite-run#  
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake servers:add[https://webconf3.ucad.sn/bigbluebutton/api,eZe4At1d  
oTeHaqjW57wQhQ8gIkUjVcTgzs3N9kobUE]  
OK  
id: 2add5f37-d60e-4a1f-938f-0b4cec015a1e  
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake servers:enable[2add5f37-d60e-4a1f-938f-0b4cec015a1e]  
OK  
root@scalelite:~/scalelite-run# █
```

**Figure 66: Ajout du premier serveur BigBlueButton**

Ainsi notre premier serveur BigBlueButton est activé. Ajoutons maintenant le deuxième serveur BigBlueButton :

```
docker exec -i scalelite-api bundle exec rake servers:add[https://secureconf-ucad.myfad.org/bigbluebutton/api/,DIgAZZ7M0nWhzqq8y7p0nt8fsL4ol2yTwYj4erdW0]
```

Cela devrait nous donner un identifiant le serveur ajouté :

```
OK  
id: 8c6a4c3d-5b41-4b3e-80a7-e199a6372488  
root@scalelite:~/scalelite-run#
```

Utilisons l'ID du serveur pour activer l'activer :

```
docker exec -i scalelite-api bundle exec rake servers:enable[8c6a4c3d-5b41-4b3e-80a7-e199a6372488]
```

Ajoutons le troisième serveur BigBlueButton :

```
root@scalelite:~/scalelite-run#  
docker exec -i scalelite-api bundle exec rake servers:add[https://bbb-  
drtp.myfad.org/bigbluebutton/api,7PL3OHRvZpw8CPRA4yf2DGaf9QqwofTchmY0kEKqo]
```

Cela devrait nous donner un identifiant pour chaque serveur ajouté.

```
OK
id: ce33d8c2-3e98-46f0-9a15-880bb0d3136f
root@scalelite:~/scalelite-run#
```

Utilisez l'ID du serveur pour activer le serveur.

```
docker exec -i scalelite-api bundle exec rake servers:enable[ce33d8c2-3e98-46f0-9a15-880bb0d3136f]
```

Après avoir ajouté et activé nos serveurs BigBlueButton, nous allons vérifier s'ils se trouvent dans le poolen exécutant la commande ci-dessous :

```
docker exec -i scalelite-api bundle exec rake status
```

```
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
              NG
secureconf-uc enabled online    0      0          0      0  1
ad.myfad.org

root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
              NG
secureconf-uc enabled online    0      0          0      0
ad.myfad.org
webconf3.ucad enabled online    0      0          0      0  2
.sn

root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
              NG
bbb-drtp.myfa enabled online    0      0          0      0  3
d.org
secureconf-uc enabled online    0      0          0      0
ad.myfad.org
webconf3.ucad enabled online    0      0          0      0
```

Figure 67: Statut des serveurs BigBlueButton ajoutés

A présent nous avons bien inséré nos trois serveurs et dans cette figure nous remarquerons que les métriques comme : le nombre de sessions, le nombres d'utilisateurs et les vidéos sont initialisés à 0.

Nous pouvons utiliser Scalelite avec n'importe quelle application externe en définissant son nom d'hôte comme URL BigBlueButton et le secret généré (LOADBALANCER\_SECRET) lors de l'installation comme BigBlueButton Secret.

```
root@scalelite: ~/scalelite-run
GNU nano 2.9.3 .env
### Required by scalelite-api
SECRET_KEY_BASE=5f940a0cefcdfc789d697cc978b546adb238cc28f285e87462dd41efaf1d1b4ab1e52245a28405665$
LOADBALANCER_SECRET=f2bc5a98a583160d18739eeab0b4869dc762543aa534f7b9 1
#
# SCALELITE_NGINX_EXTRA_OPTS=--mount type=bind,source=/etc/letsencrypt,target=/etc/nginx/ssl,read$
URL_HOST=https://bbb-scale.myfad.org/bigbluebutton/api 2
### Required when when specific repo or version (other than defaults) are needed
# SCALELITE_REPO=blindsidenetwks
# SCALELITE_TAG=v1
###
### Required by scalelite-nginx (only when using SSL)
NGINX_SSL=true
```

Figure 68: Informations de Scalelite

### III.7.4. Intégration du système Moodle et Scalelite

Après avoir installer nos serveurs Moodle, BigBlueButton et Scalelite, nous allons passer à la partie intégration de ses derniers.

#### III.7.4.1. Ajout du plugin BigBlueButton

➤ Connectons-nous à Moodle LMS

Connectons-nous à notre plateforme Moodle avec vos informations d'identification d'administrateur. Ensuite, nous serons redirigés vers notre page d'administration :

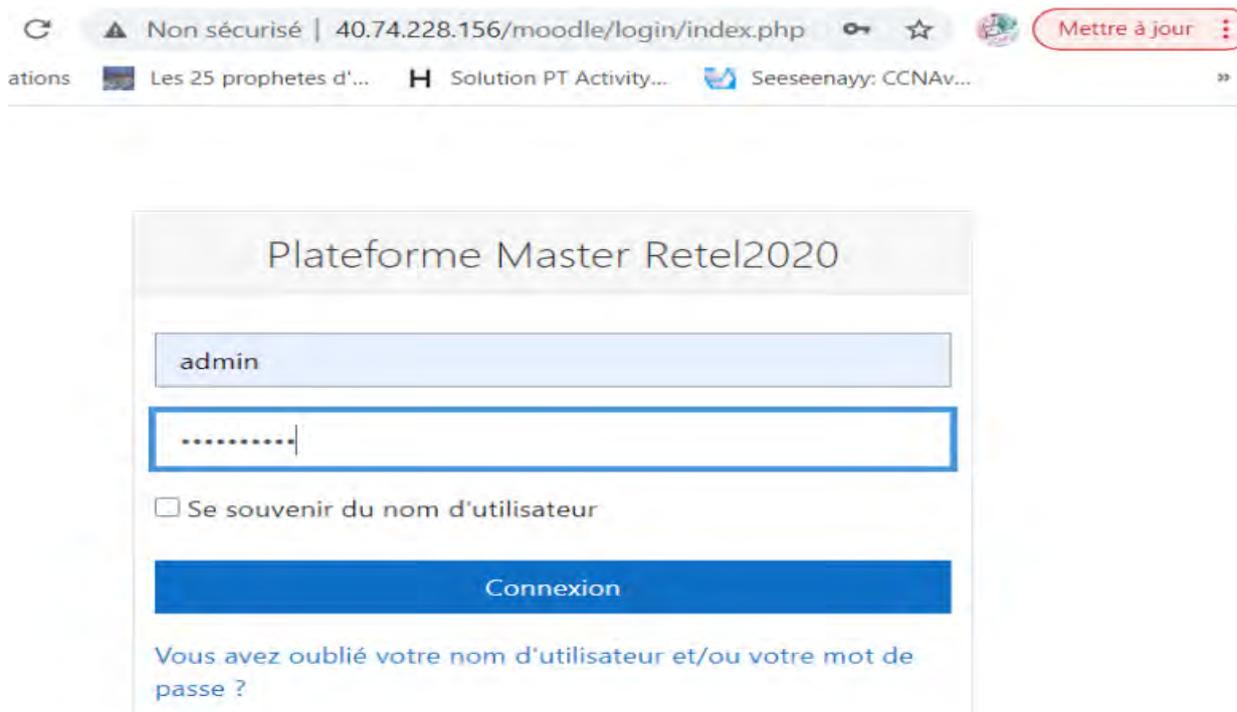


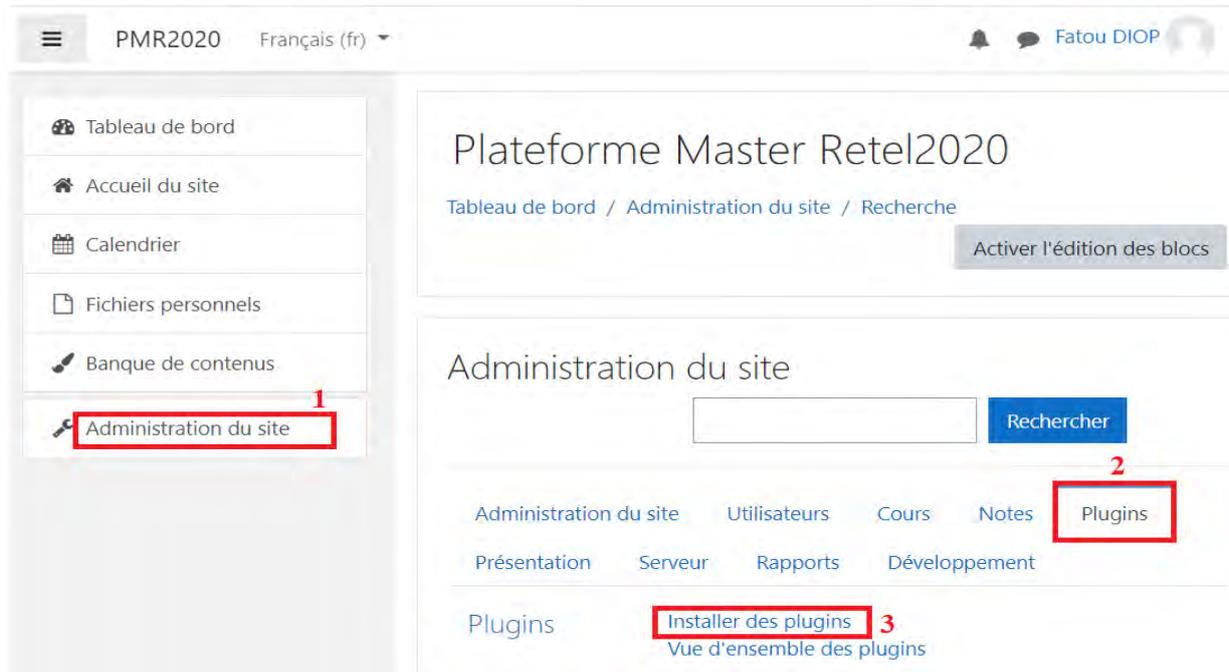
Figure 69: Connection dans notre plateforme Moodle

## ➤ Vers les plugins

Depuis notre tableau de bord d'administration, cliquons sur le lien « **Administration du site** » dans le menu de gauche.

Et dans l'administration du site, cliquons sur « **Plugins** » à partir duquel nous pouvons installer l'outil BigBlueButton.

Et encore une fois à partir des plugins, cliquons sur « **Installer les plugins** », voir la figure ci-dessous :



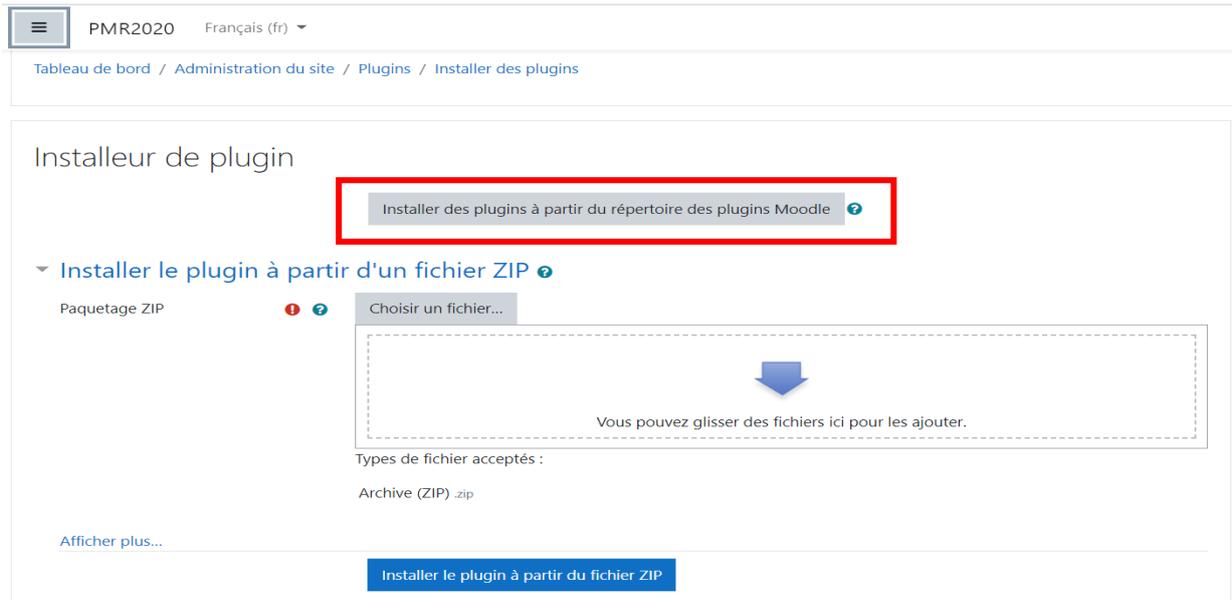
**Figure 70: Etape d'installation Plugin BigBlueButton**

## ➤ Installez le plug-in BigBlueButton

Dans cette étape, nous avons le choix entre deux options ;

1. Installer le plugin depuis le répertoire des plugins Moodle.
2. Installer le plugin à partir d'un fichier Zip.

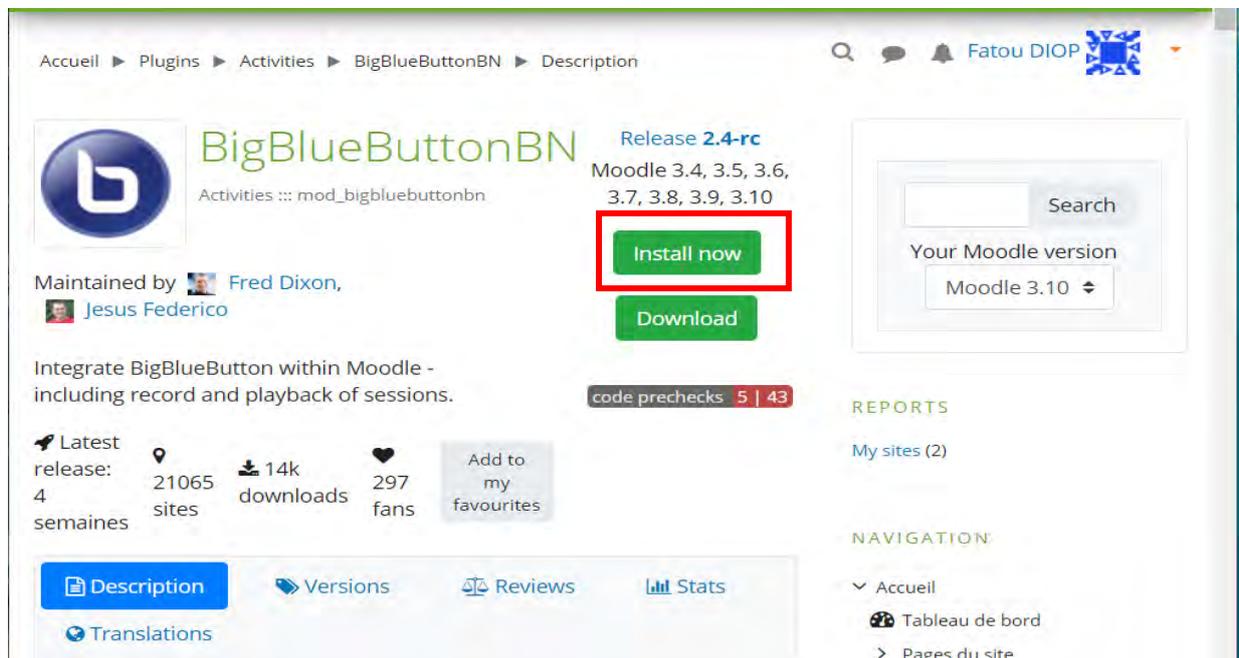
**Remarque :** comme je n'ai pas de fichier Zip pour une installation manuelle, alors je vais utiliser mon compte sur moodle.org pour télécharger le plugin, voir la figure ci-après :



**Figure 71: Installation du plugin depuis le compte Moodle**

### ➤ Télécharger du plugin

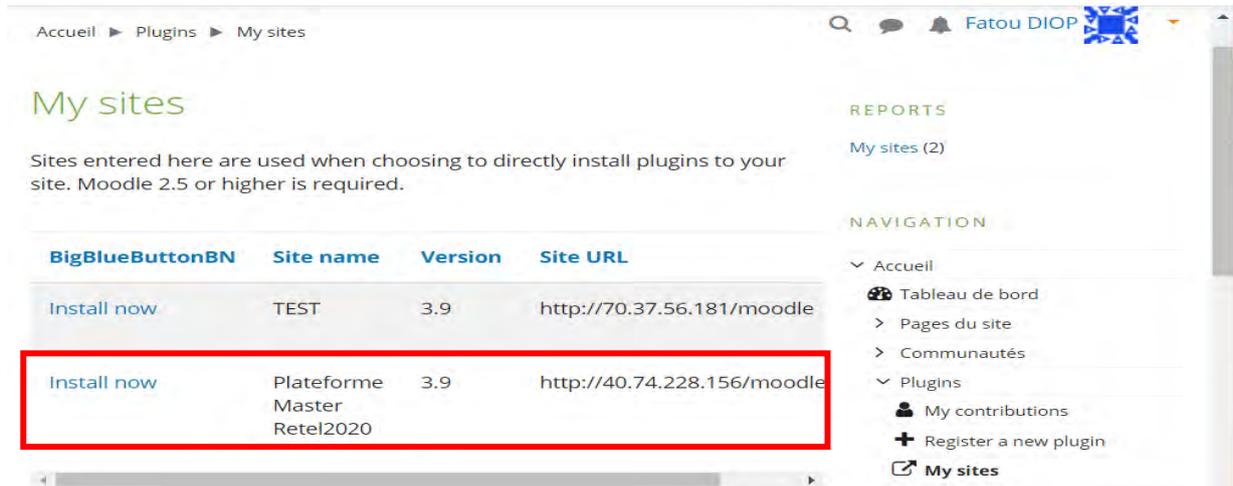
Une fois dans notre compte moodle.org, nous pouvons chercher le plugin BigBlueButton et ensuite cliquer sur « **Install now** », voir figure ci-dessous :



**Figure 72: Téléchargement du plugin BigBlueButton**

## ➤ Liaison du plugin avec nître site

Après avoir cliqué sur « **Install now** » il nous sera ensuite demandé de choisir la plateforme dans laquelle nous souhaitons installer le plugin, voir la figure ci-dessous :



**Figure 73: Choix du site d'installation**

L'installation se poursuit et nous voyons que tout se passe comme prévu, voir la figure ci-dessous :

## Plateforme Master Retel2020



**Figure 74: Installation à partir du répertoire des plugins**

Après avoir cliqué sur « **Continuer** », nous allons ensuite cliquer à nouveau sur « **Mettre à niveau la base de données Moodle maintenant** » pour continuer le processus d'installation, voir la figure ci-après :

Non sécurisé | 40.74.228.156/moodle/admin/index.php?cache=0&confirmrelease=1&confirm...

Les 25 prophètes d'... H Solution PT Activity... Seeseenay: CCNAv... Microsoft Word - rx... Architecture. d

additionnels sont affichés si une mise à jour est disponible. Il est recommandé que vous vérifiez si des versions plus récentes des plugins sont disponibles et de mettre à jour leur code source avant de poursuivre la mise à jour de ce Moodle.

Rechercher des mises à jour

Dernière vérification effectuée le 16 Avril 2021, 02:42

### Plugins requérant votre attention

Annuler les nouvelles installations (1) Plugins requérant votre attention 1

Tous les plugins 412

Nom du plugin / Dossier	Version actuelle	Nouvelle version	Requiert	Source / Statut
<b>Modules d'activité</b>				
 BigBlueButtonBN /mod/bigbluebuttonbn		2019101005	• Moodle 2016120500	<b>Additional</b> <b>A installer</b> Annuler cette installation

Actualiser

Mettre à jour la base de données maintenant

**Figure 75: Mise à jour de la base de données**

➤ **Nous y sommes presque.**

Une fois la mise à niveau effectuée, nous serons guidés pour appuyer à nouveau sur « **Continuer** ». Ensuite, nous serons dans une page de paramètres ÉNORME. Nous n'allons rien touché.

Cliquons à nouveau sur « **Enregistrer les modifications** » et ça y est l'installation est terminée :

Non sécurisé | 40.74.228.156/moodle/admin/index.php?cache=0&confirmrelease=1&confirm...

Les 25 prophètes d'... H Solution PT Activity... Seeseenay: CCNAv... Microsoft Word - rx... Architecture. de

### Mise à jour vers une nouvelle version

mod\_bigbluebuttonbn

Succès

Continuer

**Figure 76: Module ajouté**

Maintenant nous pouvons aller consulter la liste des plugins afin de voir que le plugin BigBlueButton est bien installé :

PMR2020 Français (fr) ▾

bleau de bord  
accueil du site  
calendrier  
chiers personnels  
 Banque de contenus  
Administration du site

### Vue d'ensemble des plugins

Rechercher des mises à jour

Dernière vérification effectuée le 16 April 2021, 02:42

Tous les plugins 412 Plugins additionnels 1

Nom du plugin	Version	Disponibilité	Actions	Annotations
<b>Modules d'activité</b>				
Devoir mod_assign	2020061500	Activé	Paramètres Désinstaller	
Devoir 2.2 (désactivé) mod_assignment	2020061500	Désactivé	Désinstaller	
BigBlueButtonBN mod_bigbluebuttonbn	2.4-rc 2019101005	Activé	Paramètres Désinstaller	Additional

Figure 77: Liste des plugins

### III.7.4.2. Intégration du système Moodle et Scalelite

L'intégration de nos deux systèmes est simple, il faut juste au niveau des paramètres du plugin BigBlueButton, renseigner et enregistrer l'URL\_HOST et le LOADBALANCER\_SECRET de Scalelite, voir la figure ci-dessous :

## Plateforme Master Retel2020

Les réglages ci-dessous ont été ajoutés lors de la dernière mise à jour de votre Moodle. Effectuez les modifications » au bas de cette page.

### Nouveaux réglages - BigBlueButtonBN

URL du serveur BigBlueButton  
bigbluebuttonbn\_server\_url  Défaut : http://t...

L'URL de votre serveur BigBlueButton doit se terminer par BigBlueButton fourni par Blindside Networks à utiliser pou...

Secret partagé BigBlueButton  
bigbluebuttonbn\_shared\_secret  Défaut : 8cd8ef5...

Le sel de sécurité de votre serveur BigBlueButton. (Ce sel p...  
Networks que vous pouvez utiliser pour les tests.)

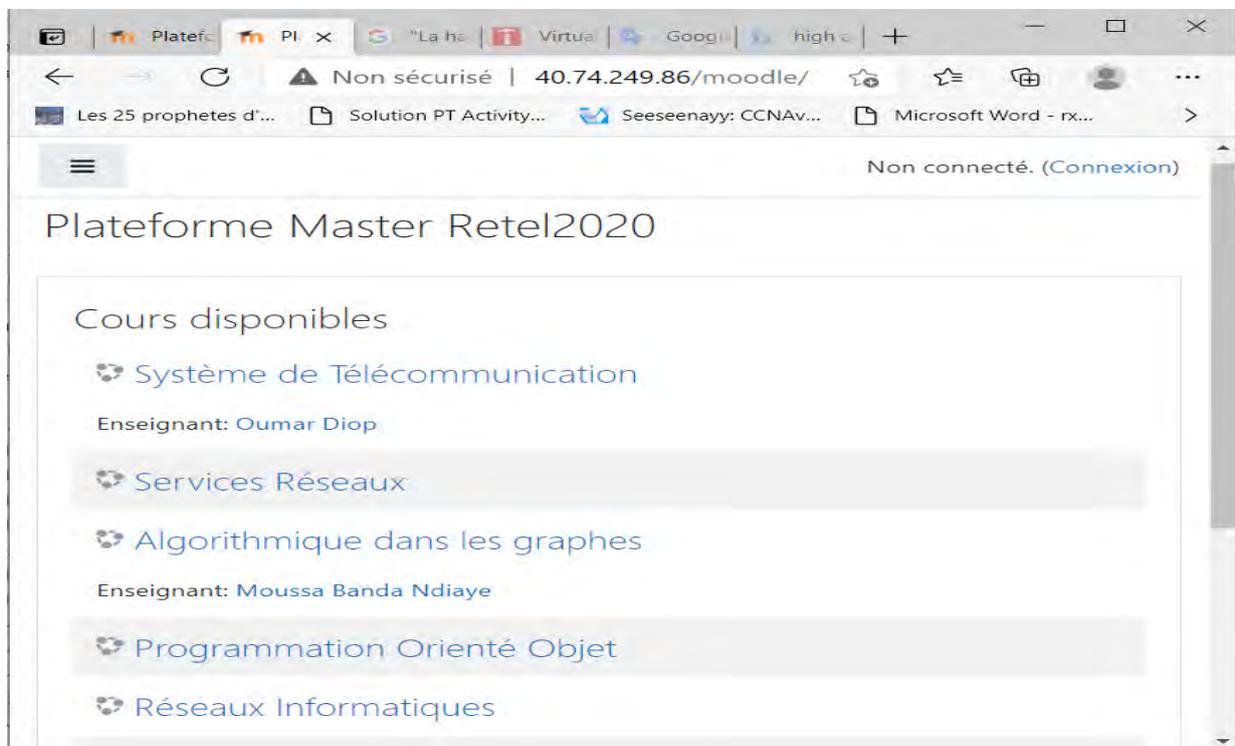
Figure 78: Intégration Moodle et Scalelite

### III.7.5. Tests et Validation

Dans cette partie, nous allons tester le fonctionnement de notre système composé de deux sous-systèmes : le système Moodle et le système BigBlueButton.

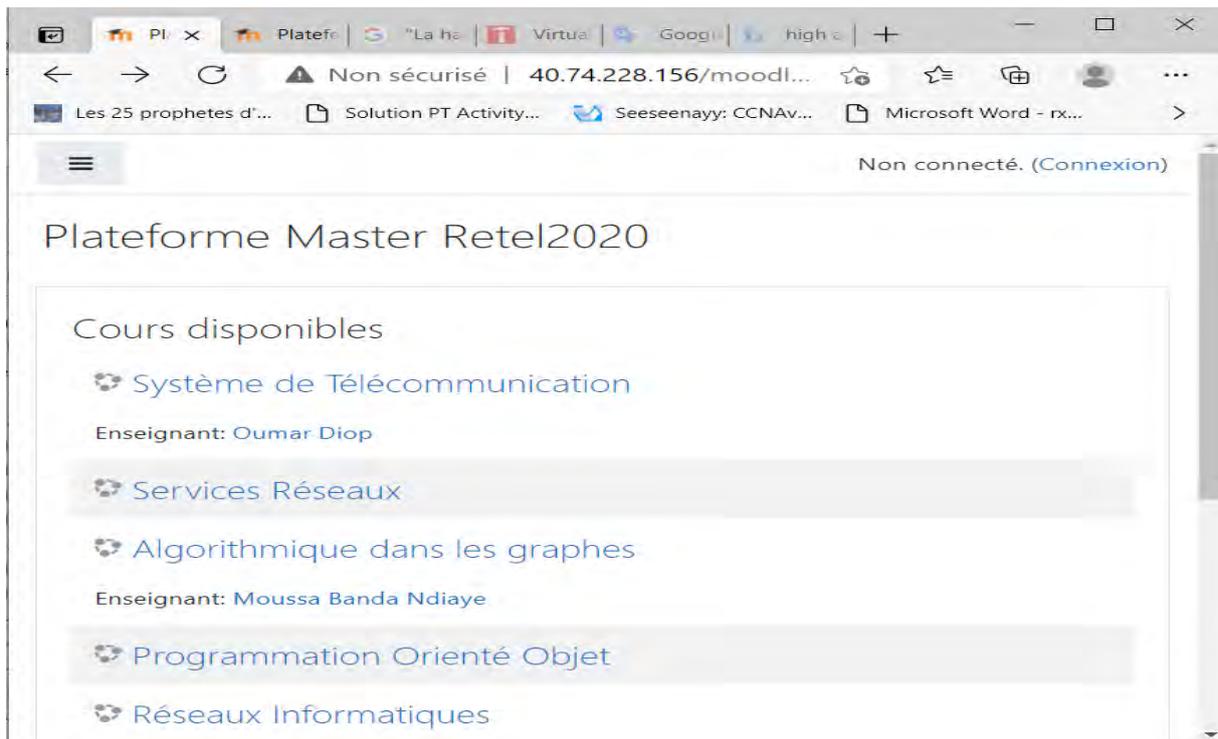
#### ➤ Test du Haproxy et de la réplication entre les serveurs Moodle

Dans le système Moodle nous pouvons voir que le serveur frontal Haproxy redirige les demandes d'accès aux plateformes suivant l'algorithme least connections. La première demande sera redirigée vers le serveur web moodle\_1 et pour les autres, il va appliquer la règle de l'algorithme least connections. Il suffit juste de taper l'adresse du serveur HAProxy suivi de /moodle : <http://70.37.111.91/moodle> pour accéder à la plateforme Moodle :



**Figure 79: Première connexion**

Pour la deuxième connexion nous serons redirigés vers la deuxième plateforme :



**Figure 80: Deuxième connexion**

Notons que ces deux plateformes sont identiques car toutes modifications apportées à l'une seront répliquées dans l'autre puisque les deux serveurs pointent sur le même cluster de base de données et sur le même dossier moodledata.

➤ **Test de la distribution des classes synchrones**

Dans cette partie nous allons connecter quelques utilisateurs dans des classes virtuelles différentes :

**Tableau 3 : Listes des participants aux réunions**

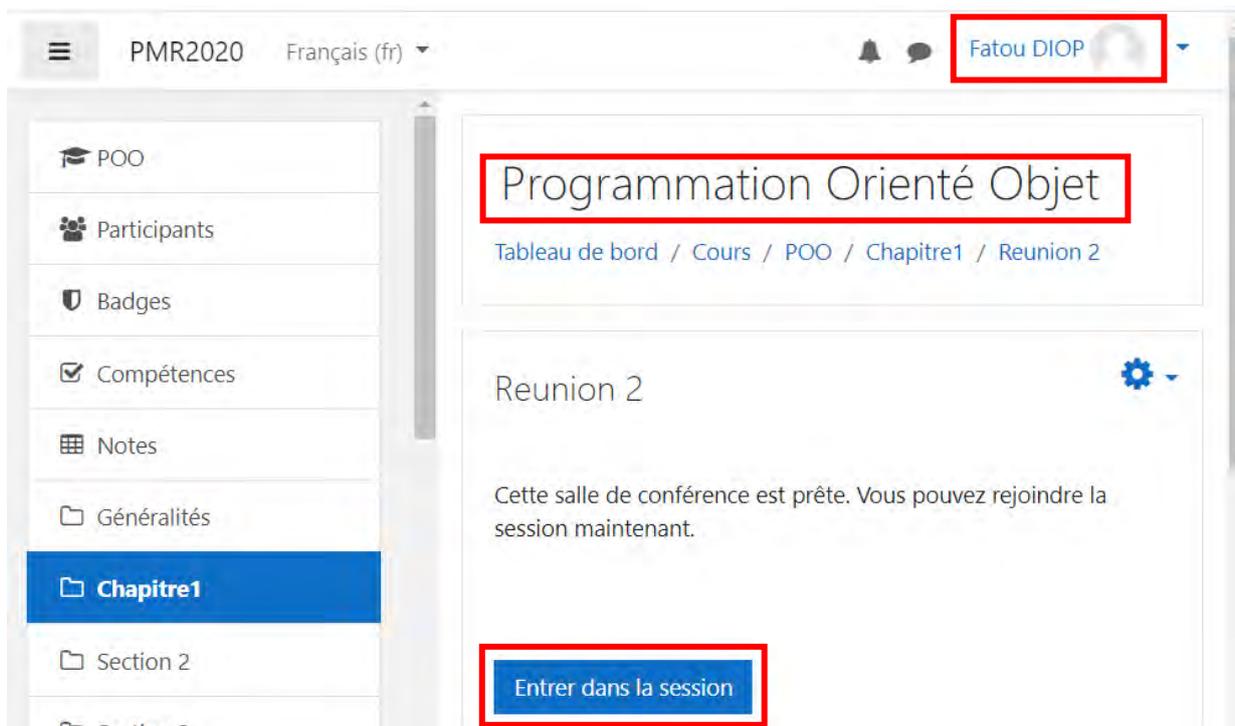
Utilisateurs	Réunions
Fatou DIOP	Réunion_2
Oumar DIOP	Réunion_3
Kadiatou SY	Réunion_1
Moussa Banda NDIAYE	Réunion_4

Avant de commencer les tests nous allons voir d'abord l'état initial de Scalelite :

```
root@scalelite: ~/scalelite-run
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled online    0         0         0         0
ad.org
secureconf-uc  enabled online    0         0         0         0
ad.myfad.org
webconf3.ucad  enabled online    0         0         0         0
.sn
root@scalelite:~/scalelite-run#
```

**Figure 81: Etat initial de Scalelite**

Maintenant nous allons voir dans quel serveur BigBlueButton sera placé la première réunion. L'utilisateur Fatou DIOP va se connecter et entrer dans la réunion (classe virtuelle), voir la figure ci-dessous :



**Figure 82: Première classe virtuelle**

Nous allons voir que l'utilisateur se trouve bien dans réunion, voir la figure ci-dessous :

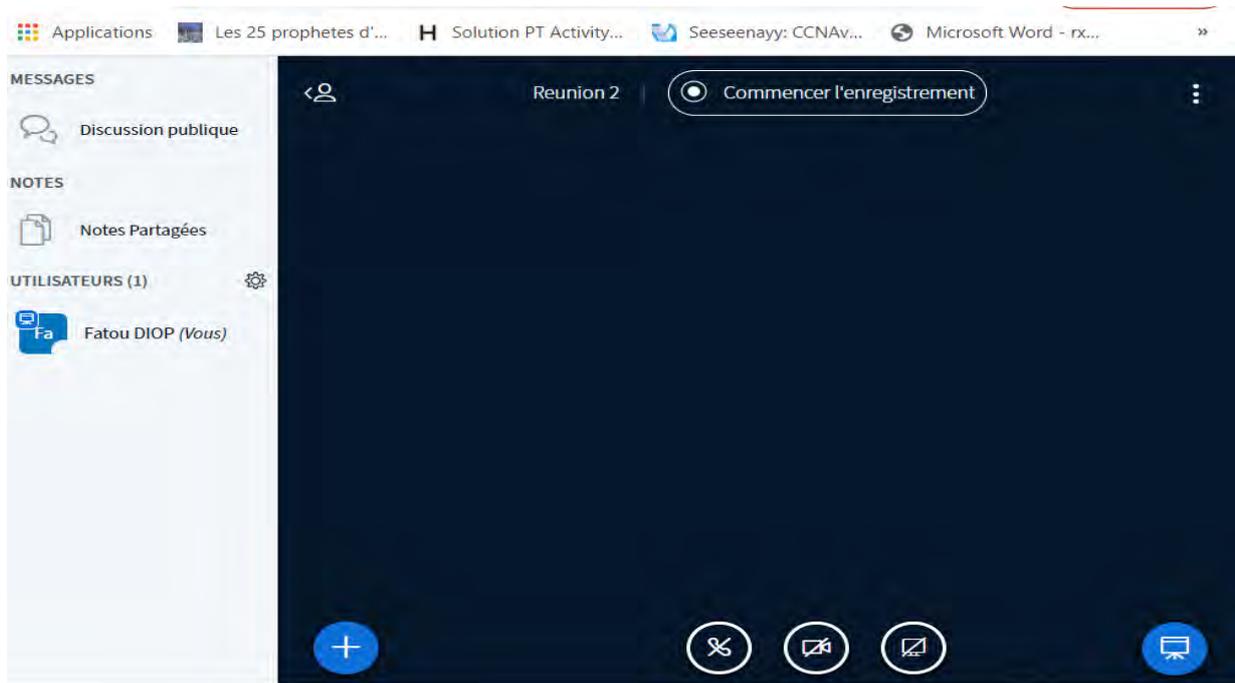


Figure 83: Première classe virtuelle (1)

Maintenant voyons l'état du serveur Scalelite pour voir dans quel serveur il a placé la réunion, voir la figure ci-dessous :

```
root@scalelite: ~/scalelite-run
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled online   0         0         0         0
ad.org
secureconf-uc enabled online   0         0         0         0
ad.myfad.org
webconf3.ucad  enabled online   0         0         0         0
.sn
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled online   0         0         0         0
ad.org
secureconf-uc enabled online   0         0         0         0
ad.mvfad.org
webconf3.ucad  enabled online   1         1         1         0
.sn
root@scalelite:~/scalelite-run#
```

Figure 84: Etat Scalelite (1)

Nous remarquons que la première réunion est placée dans le serveur 3.

Maintenant connectons l'utilisateur Oumar DIOP :

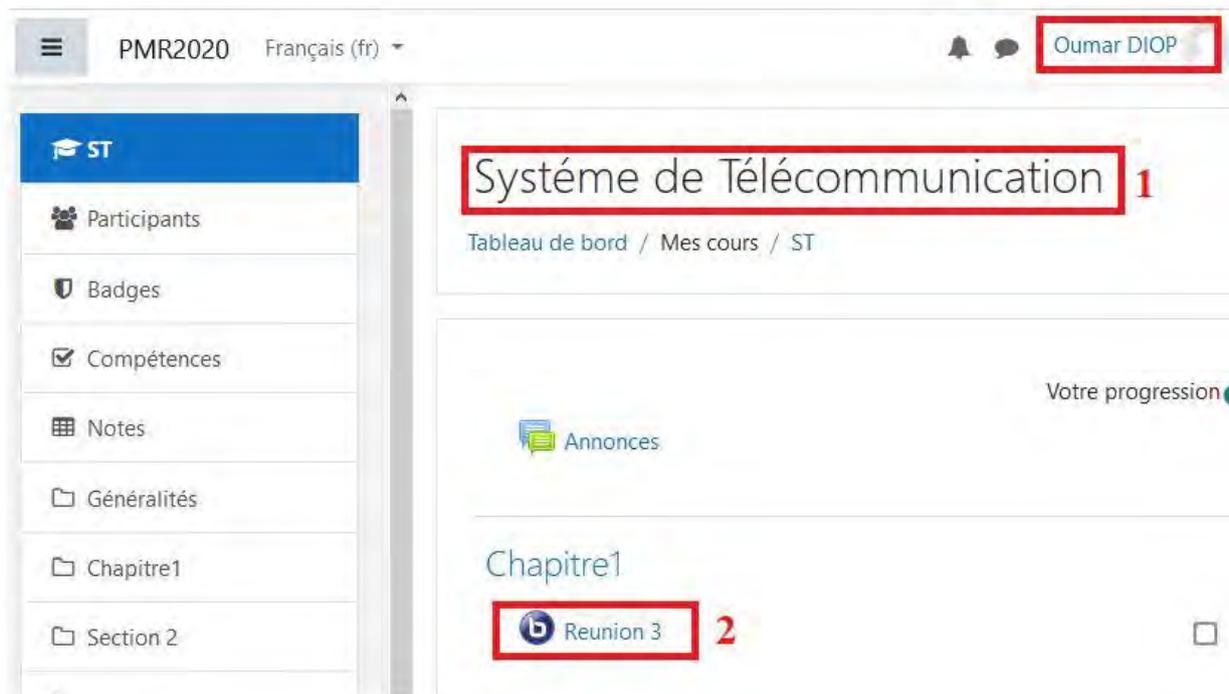


Figure 85: Deuxième classe virtuelle

Regardons à nouveau le comportement du serveur Scalelite :

```
.sn
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled online    1      1          1      0
ad.org
secureconf-uc enabled online    0      0          0      0
ad.myfad.org
webconf3.ucad  enabled online    1      1          1      0
.sn
```

Test 2

Figure 86: Etat Scalelite (2)

Nous constatons que la deuxième classe virtuelle est placée dans le premier serveur.

Maintenant connectons l'utilisateur Kadiatou SY :

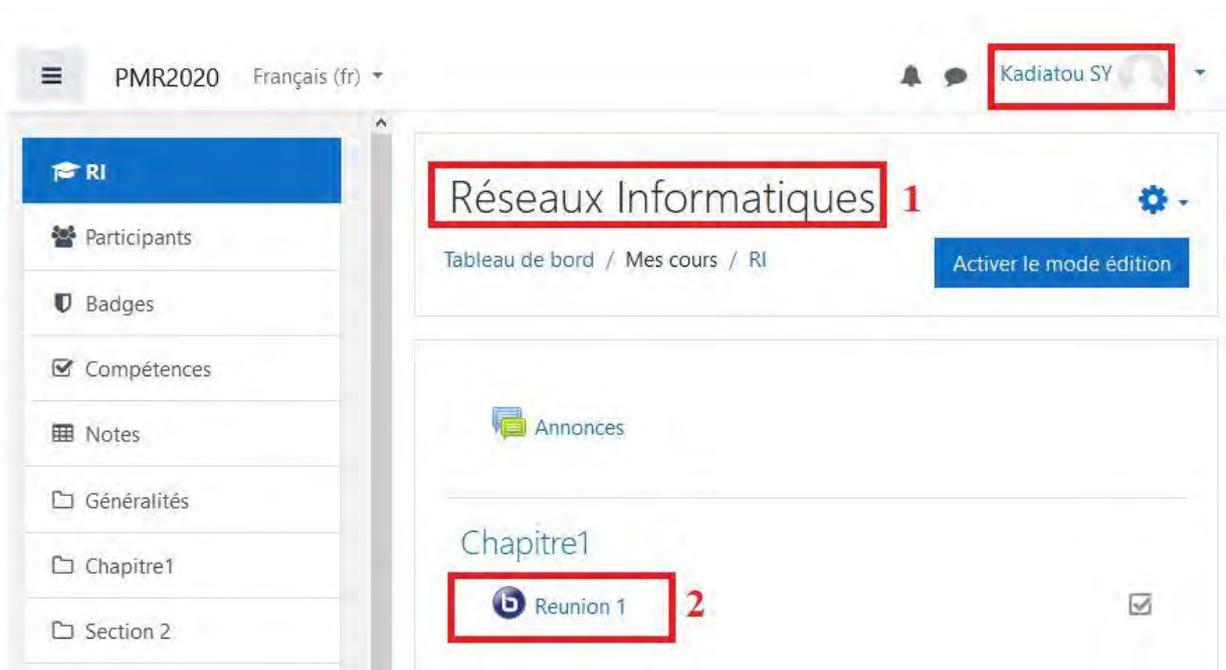


Figure 87: Troisième classe virtuelle

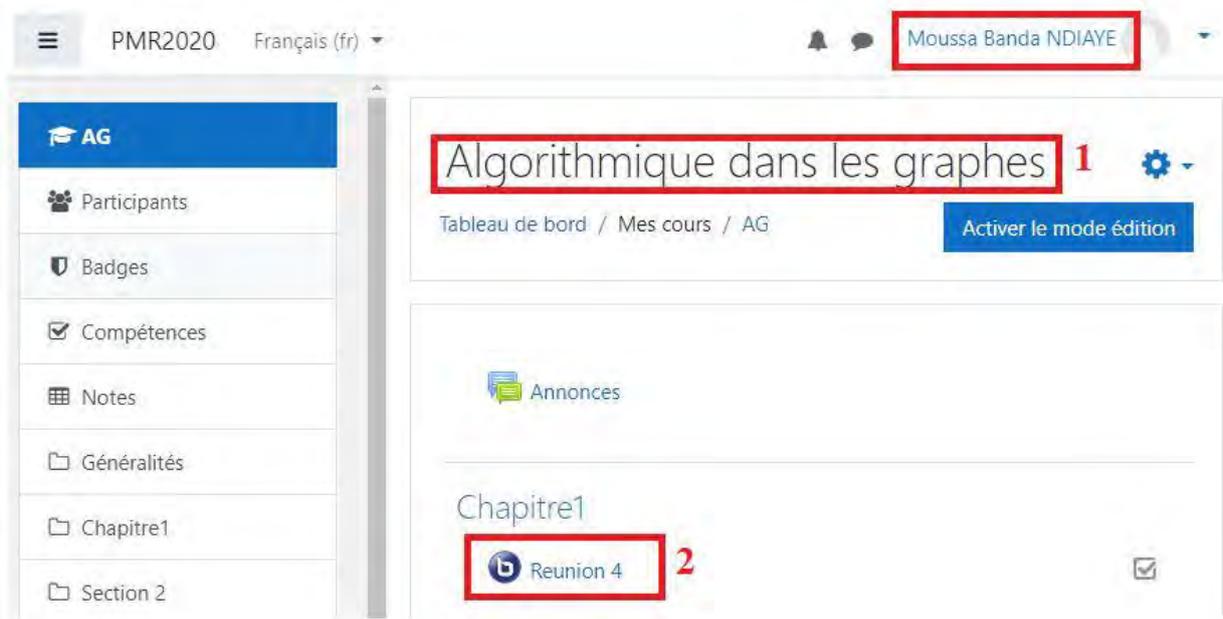
Regardons l'état du serveur Scalelite à nouveau :

```
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME    STATE    STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled  online    1         1         1         0
ad.org
secureconf-uc  enabled  online    0         0         0         0 Test 2
ad.myfad.org
webconf3.ucad  enabled  online    1         1         1         0
.sn
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME    STATE    STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled  online    1         1         1         0
ad.org
secureconf-uc  enabled  online    1         1         1         0 Test 3
ad.myfad.org
webconf3.ucad  enabled  online    1         1         1         0
sn
```

Figure 88: Etat Scalelite (3)

Nous voyons que la troisième classe virtuelle est placée dans le deuxième serveur.

Alors pour mieux voir le fonctionnement du serveur Scalelite nous allons connecter un dernier utilisateur dans une classe virtuelle et voir dans quel serveur sera placé cette dernière :



**Figure 89: Quatrième classe virtuelle**

Voyons maintenant dans quel serveur est placé la réunion de ce dernier utilisateur :

```

root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled online    1      1          1      0
ad.org
secureconf-uc enabled online    1      1          1      0
ad.myfad.org
webconf3.ucad enabled online    1      1          1      0
.sn
root@scalelite:~/scalelite-run# docker exec -i scalelite-api bundle exec rake status
  HOSTNAME      STATE  STATUS  MEETINGS  USERS  LARGEST MEETI  VIDEOS
                                     NG
retel-bbb.myf  enabled online    1      1          1      0
ad.org
secureconf-uc enabled online    2      2          1      0
ad.myfad.org
webconf3.ucad enabled online    1      1          1      0
.sn
  
```

**Figure 90: Etat Scalelite (4)**

Superbe nous voyons bien que notre serveur Scalelite fonctionne bien.

Après une étude sur quelques systèmes du Cloud, sur quelques solutions et les techniques de virtualisation mais aussi sur quelques plateformes d'enseignements à distance, nous avons installé les différents composants de l'architecture proposée, nous avons tester son fonctionnement, donc nous avons atteints les objectifs spécifiques que nous avons fixés.

# CONCLUSION GENERALE ET PERSPECTIVES

Les facultés de l'université Cheikh Anta Diop De Dakar disposent chacune une plateforme Moodle de cours en ligne connectée à un serveur de conférence web BigBluButton se trouvant tous dans le Cloud. Cependant, avec cette solution, des limites se présentent dues au nombre d'étudiants, aux besoins en nombre de connexions simultanées, mais aussi, cette solution ne garantit ni une redondance des plateformes Moodle, ni des serveurs BigBluButton. Par conséquent, nos objectifs étaient de proposer une architecture qui va non seulement garantir une haute disponibilité, mais aussi une scalabilité des plateformes dans un environnement virtuel.

Après avoir fait un état de l'art sur quelques systèmes Cloud, une étude comparative de quelques solutions de virtualisation et de quelques plateformes LMS, nous avons ensuite fait un rappel sur les services de gestion de la haute disponibilité et en fin proposer une architecture hautement disponible. L'architecture de test a été déployée et nous avons effectué des tests afin de le valider.

Malgré les difficultés rencontrées pour la réalisation de ce travail, nous avons atteint nos objectifs. Le travail a été instructif, il nous permet de comprendre la notion de Cloud, de virtualisation, les algorithmes d'équilibrage de charge mais aussi de faire connaissance sur l'existence des autres plateformes de cours à distance.

Pour la continuité de ce travail, nous envisageons d'améliorer notre solution en mutualisant les serveurs Moodle et BigBluButton de chaque faculté pour ensuite l'adopter au sein de l'université ou d'autres établissements.

# BIBLIOGRAPHIE

- [1] **Peter Mell and Timothy Grance**, “National Institute of Standards and Technology Special”. Publication 800-145.7 pages (September 2011)
- [2] **Dinh, H. T., Lee, C., Niyato, D., & Wang, P.** (2013). “A survey of mobile cloud computing: architecture, applications, and approaches”. *Wireless communications and mobile computing*, 13(18), pp 1587-1611.
- [3] **Dr. F.Z. Filali**, « Chapitre 3 : Machines Virtuelles et conteneurs ». Université Abdelhamid Ibn Badis de Mostaganem. Master1 Réseaux et Systèmes. Matière : Virtualisation des Systèmes, Année universitaire : 2019/2020
- [4] **Simon M.C. Cheng** “Introduce, design, and implement high availability clusters using Proxmox”, First published: October 2014
- [5] **Mohammad Nabil Almunawar** (Universiti Brunei Darussalam, Brunei) and **Hasan Jawwad Almunawar** (P. T. Tegar Kupas Mediatama, Indonesia) “From Information Systems Outsourcing to Cloud Computing”. Published in the United States of America by IGI Global Chapter 95 (pages 1101-1115)
- [6] **Akrivi Krouska, Christos Troussas, Maria Virvou**, “Comparing LMS and CMS Platforms Supporting Social e-Learning in Higher Education”. Published in: 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA). Date of Conference: 27-30 Aug. 2017
- [7] **Amal Ganesh, M. Sandhya and Sharmila Sankar**, «A study on fault tolerance methods in Cloud Computing». Published in: 2014 IEEE International Advance Computing Conference (IACC). Date of Conference: 21-22 Feb. 2014
- [8] **Elizabeth Sherly, Suji Gopinath** “A Comprehensive Survey on Data Replication Techniques in Cloud Storage Systems”. *International Journal of Applied Engineering Research* ISSN 0973-4562 Volume 13, Number 22 (2018) pp. 15926-15932
- [9] **Sidra Aslam, Munam Ali Shah** “Load Balancing Algorithms in Cloud Computing: A Survey of Modern Techniques”. Published in: 2015 National Software Engineering Conference (NSEC). Date of Conference: 17-17 Dec. 2015

# WEBOGRAPHIE

- [10] <https://www.futura-sciences.com/tech/definitions/informatique-cloud-computing-11573/> ; consulter le 04/10/2020
- [11] <https://www.uniprint.net/fr/7-types-cloud-computing-structures/> (service cloud figure) ; consulter le 04/10/2020
- [12] <https://devanshagarwal121.medium.com/virtualization-and-hypervisors-9c4c8f4ab27d> ; consulter le 10/10/2020
- [13] <https://www.institut-numerique.org/chapitre-2-les-technologies-du-cloud-computing-51c0279ca9f6f> ; consulter le 10/10/2020
- [14] <https://systeme.developpez.com/tutoriels/virtualisation/livre-blanc-lucas-bonnet/> ; consulter le 10/10/2020
- [15] <https://fr.wikipedia.org/wiki/Hyper-V> ; consulter le 28/10/2020
- [16] <https://www.tomshardware.fr/comparatif-virtualisation-les-solutions-gratuites/6> ; consulter le 28/10/2020

**Installation du cluster Galera** : consulter le 10/04/2021

<https://clouding.io/hc/es/articles/360010622579-CI%C3%BAster-de-MariaDB-con-Galera-y-HAProxy-en-Ubuntu-18-04>

<https://sourcecode.web.id/artikel/galera-cluster-high-availability-dengan-haproxy-di-ubuntu-18-04/>

<https://severalnines.com/database-blog/deploying-mysql-galera-cluster-40-amazon-aws-ec2>

<https://severalnines.com/database-blog/mariadb-cluster-offline-installation-centos>

**Installation du ProxySQL** : consulter le 13/04/2021

<https://severalnines.com/blog/how-set-read-write-split-galera-cluster-using-proxysql>

<https://community.centminmod.com/threads/proxysql-for-mariadb-galera-cluster.19040/>

<https://community.hostek.com/t/high-availability-with-a-mariadb-galera-mysql-cluster-with-proxysql-tutorial/407>

<https://linuxglobe.com/centos/centos-7/proxysql-for-mariadb-galera-clustering-centos-7.html>

**Installation du HAProxy** : consulter le 15/04/2021

<https://www.tuxtips.net/how-to-install-and-configure-haproxy-on-centos-7>

**Installation de Moodle** : consulter le 15/04/2021

<https://websiteforstudents.com/setup-moodle-cms-on-ubuntu-18-04-lts-beta-with-apache2-mariadb-and-php-7-1-support/>

<https://benisnous.com/installing-moodle-3-9-lms-on-centos-7-how-to/>

**Installation de BigBlueButton** : consulter le 16/04/2021

<https://docs.bigbluebutton.org/2.2/install.html>

**Installation de Scalelite** : consulter le 16/04/2021

<https://github.com/blindsidenetworks/scalelite/blob/master/README.md>