UCLouvain
Earth and Life Institute (ELI)
Georges Lemaître Centre for Earth and Climate Research

Projet de premier brevet

Examen d'avancement au grade d'informaticien-expert

Pierre-Yves Barriat

FGS : 01108821

Institut ELI - pôle ELIC

Dénomination de la fonction : informaticien de recherche

Fonction exercée depuis le : 5 novembre 2007

Grade et barème actuels : informaticien - 12/2

Grade et barème sollicités : informaticien-expert - 13/3

05 décembre 2022

LOUVAIN-LA-NEUVE | BRUXELLES | MONS | TOURNAI | CHARLEROI | NAMUR
Place Louis Pasteur, 3 bte L4.03.08, 1348 Louvain-la-Neuve, Belgique
Tél. +32 (0)10 47 33 02 – fax +32 (0)10 47 47 22 – pierre-yves.barriat@uclouvain.be

1 Introduction

1.1 Contexte

Les chercheurs sont de plus en plus confrontés à la gestion d'énormes quantités de données scientifiques. Ces dernières sont soit produites localement soit téléchargées depuis les machines d'autres centres de recherche ou depuis des dépôts centralisés. Elles sont ensuite utilisées, analysées, étudiées, modifiées, localement ou à distance sur d'autres machines de calculs.

La gestion de cette masse de données constitue un réel défi pour de nombreux pôles de recherche de notre institution. En effet, chaque chercheur ou groupe de chercheurs est soumis à une phase de formation et d'adaptation afin de maîtriser les différentes méthodologies à appliquer pour pouvoir obtenir et travailler sur telles ou telles données, et afin de pouvoir les stocker et les partager, les diffuser ou les réutiliser.

Le stockage et l'accessibilité des données constituent l'enjeu de ce brevet.

1.2 Solutions de stockage

Il existe 3 principales solutions de stockage des données à l'UCLouvain, hormis les solutions d'archivage et de backup. L'utilisation de l'une ou l'autre de ces solutions dépend du type des données (volumes, finalité, etc) mais aussi de l'environnement de l'utilisateur (système d'exploitation, logiciels, etc) et de l'environnement pour les données elles-mêmes (origine, logiciels pour les exploiter, etc).

Le système de fichiers OASIS[1] est un espace de stockage centralisé qu'il est possible d'intégrer à n'importe quel environnement de travail (multiplateformes via différents protocoles) au sein du réseau institutionnel. Il est utilisé pour des données dont la taille reste de l'ordre du Mo jusqu'à plusieurs Go et offre un backup journalier. OASIS représente un système de fichier dans un réseau privé (réseau UCLouvain) en "modèle interne" (entièrement géré et hébergé localement).
C'est un système de stockage hiérarchique qui fournit un accès partagé aux données. Les utilisateurs peuvent créer, supprimer, modifier, lire et écrire des fichiers et peuvent les organiser logiquement dans des arborescences de répertoires (accès intuitif).

SharePoint est une solution de stockage dans un cloud publique en modèle SaaS (Software as a Service ou Logiciel en tant que Service) : entièrement géré et hébergé par Microsoft. Cette solution est parfaitement intégrée à la suite de logiciels bureautique MS Office 365. Il s'agit d'un espace de travail collaboratif partagé, mais exclusivement disponible pour les utilisateurs de l'UCLouvain. L'utilisation de Sharepoint se fait en ligne via un navigateur. Il est possible de l'intégrer davantage (synchronisation, édition locale, etc) à l'environnement de travail via un client OneDrive (mais pas pour un environnement

[1] https://intranet.uclouvain.be/fr/myucl/services-informatiques/service-fichier-personnel-en-detail.html

GNU/Linux).
Pour une utilisation plus individuelle, OneDrive est plus approprié[2]. OneDrive (via un compte UCLouvain) permet de stocker et sauvegarder de grande quantité de données en toute sécurité dans l'UE (respectant les recommandations GDPR). Mais les données ne sont pérennes que pour un utilisateur de l'UCLouvain: si ce dernier quitte l'institution, les données disparaissent.

Les solutions de stockage de Microsoft ne sont en revanche pas ou peu adaptées pour des données sous environnement GNU/Linux. En outre, collaborer (SharePoint) ou partager des données (SharePoint, OneDrive) avec des utilisateurs extérieurs n'est pas automatique: il est nécessaire d'être authentifié avec un compte Microsoft (UClouvain, personnel ou d'une autre organisation).

Pour une partie du parc de machines individuelles de l'UCLouvain utilisant un environnement de travail GNU/Linux (majoritaires par exemple dans les pôles de recherche ELIC, TFL, MODL, ELEN, INMA, etc) mais aussi pour les clusters HPC et les serveurs interactifs partagés, il n'est donc pas aisé de lier efficacement l'un de ces services de stockage au système de fichiers local (du poste de travail ou de la machine partagée). Pour cela, il existe le service du stockage de masse proposé par la plateforme technologique du CISM. Cet espace de stockage offre une très grande capacité aux utilisateurs et de hautes performances. En revanche, celui-ci est essentiellement adapté aux environnements Unix (MacOs ou GNU/Linux) car accessible uniquement via le protocole SSH. Comme OASIS, le stockage de masse du CISM est un système de fichier dans un réseau privé en "modèle interne" (entièrement géré et hébergé au CISM).

2 Description du projet

Comment offrir une solution de stockage multiplateformes combinant grande capacité de stockage, gestion des grands groupes de données (datasets >10Go) et hébergée localement ?

Nextcloud est un logiciel libre qui propose de combiner cela. Il s'agit d'un logiciel de site d'hébergement de fichiers mais aussi d'une plateforme de collaboration. C'est-à-dire qu'il peut s'utiliser comme un espace de stockage dans le nuage publique à la manière de OneDrive ou DropBox, mais en "modèle interne" (entièrement géré et hébergé localement).
Il propose en outre une panoplie de fonctionnalités afin d'offrir des services à la manière d'Office 365 ou des services de Google: gestion des agendas (CalDAV), des contacts (CardDAV), des tâches, des notes, espace de collaboration (suite bureautique en ligne basée sur LibreOffice), gestion de version des fichiers, partage multiple, etc. Il est également possible de lui ajouter des extensions afin de prendre en charge des espaces de stockage externes par protocoles (FTP, SSH, NFS, WebDAV ou SMB/CIFS comme OASIS) mais aussi par services (comme OneDrive ou DropBox). Il peut également prendre en charge le stockage objet (comme Amazon S3 ou OpenStack).

[2] https://intranet.uclouvain.be/fr/myucl/services-informatiques/applications-disponibles.html

Enfin, Nextcloud propose un logiciel client multiplateformes pour une intégration totale avec tous les environnements (Windows, MacOs, GNU/Linux, Android, iOS), dispose d'un système d'authentification à deux facteurs, et respecte les recommandations GDPR.

Dans le contexte de la gestion des grandes quantités de données scientifiques, je propose une collaboration avec le CISM afin de pérenniser un service Nextcloud, c'est-à-dire le rendre performant et disponible à long terme.
L'objectif premier est d'offrir aux chercheurs un accès efficace à leurs données non bureautiques, c'est-à-dire non exploitables par des logiciels comme ceux fournis par la suite MS Office et nécessitant un traitement spécifique. La mise en place de ce service offre également l'opportunité d'intégrer les solutions de stockage existantes de l'Institution au sein d'une plateforme commune.

2.1 Objectifs

Dans le cadre de ce premier brevet, les objectifs poursuivis sont:

- d'installer une instance de Nextcloud dans l'infrastructure OpenStack du CISM
- d'optimiser le déploiement du service en termes de performances, de robustesse, d'accessibilité et de maintenance
- de créer des liens efficaces entre Nextcloud et les différents espaces de stockage: stockage de masse, OneDrive, OASIS, stockage partagé CECI, etc.
- d'offrir via le service Nextcloud une solution intégrée de gestion de données volumineuses à l'ensemble des chercheurs de l'institut ELI ainsi qu'à tous les pôles de recherche intéressés

Une phase de test du service Nextcloud a déjà été réalisée en amont de ce brevet: j'ai en effet pu installer une instance locale à petite échelle : déploiement simple (LAMP : acronyme pour Linux, Apache, MySQL, PHP) dans un conteneur virtuel Docker, sur une machine interactive financée par ELIC et installée dans l'infrastructure du CISM depuis 2017. Cette machine arrive cependant en fin de vie en juillet 2022.
Cette phase de test est arrivée à son terme et le but du présent brevet est de pérenniser ce service.

À l'heure actuelle, 700 téraoctets de données sont stockés localement par le pôle ELIC. Ces informations doivent être gérées et accessibles, ce pour quoi Nextcloud offre une solution pratique. Il permet un accès à la demande à partir de plusieurs appareils connectés et facilite le déploiement rapide et l'archivage des données.

Ce projet se positionne au niveau d'un institut de recherche (ELI) en collaboration avec une plateforme technologique sectorielle (CISM). Les services de l'Institution en relation avec le projet ou impacté sont le CISM et le SGSI.

2.2 Produit

Les délivrables de ce projet prendront la forme d'une instance Nextcloud entièrement opérationnelle:

- côté serveur, dans l'environnement CISM pour l'infrastructure backend (service et stockage cloud)
- côté client, disponible pour les utilisateurs via une interface web ou via les applications clientes disponibles pour Windows, MacOS, GNU/Linux (de nombreuses distributions), iOS et Android.

Les spécifications et fonctionnalités de cette instance seront les suivantes:

- authentification via l'utilisation d'un compte CISM préalablement créé
- gestion des versions de fichiers (la fréquence de sauvegarde et la fréquence de conservation sont définies par l'administrateur)
- partage des fichiers au niveau utilisateurs (les fichiers ou dossiers individuels peuvent être partagés avec des personnes sélectionnées sur les comptes Nextcloud, ou avec n'importe qui via un simple lien URL, l'expéditeur ayant un grand contrôle sur le processus. Ils peuvent, par exemple, définir une date d'expiration pour le lien, exiger un mot de passe pour ouvrir le fichier envoyé, joindre une note, etc.)
- collaboration (Collabora Online est une suite bureautique en ligne basée sur LibreOffice qui prend en charge tous les principaux formats de documents, de feuilles de calcul et de fichiers de présentation.)
- montage de stockages externes par protocoles SSH ou CIFS
- montage de stockages externes des services OneDrive, DropBox et GoogleDrive
- authentification à deux facteurs (via des codes de sauvegarde ou une application d'authentification TOTP)
- conforme au RGPD

3 Contraintes

Les contraintes de ce projet s'expriment essentiellement en termes de délais et de ressources.

3.1 Délais

Voici une proposition pour l'ensemble du calendrier:

- **25 janvier 2022 :** soumission d'acte de candidature à l'examen
- **15 avril 2022 :** soumission du présent projet pour le premier brevet

- **28 avril 2022 :** analyse de l'acte de candidature à l'examen et du présent projet par la Commission paritaire
- **mai 2022 :** actualisation et initialisation du projet
- **fin mars 2023 :** fin du premier brevet

Durée du projet : **11 mois** à partir de la validation du 28 avril 2022

Livraison estimée d'un produit minimum viable (MVP) : **janvier 2023**

Les échéances intermédiaires sont détaillées dans la section de planification du projet.

3.2 Ressources

Voici certaines ressources à prendre en compte:

- accessibilité à une infrastructure de développement et de validation (test) puis de production. Prise en compte des points suivants :
  - localisation et hébergement des serveurs : qualité des performances, possibilité d'accès, coût, sécurité, etc.
  - réseau (type, vitesse et performance des liaisons, disponibilité, support)
  - outils de sécurisation des transactions et des serveurs (certificats, authentification par clés, mots de passe, firewall, etc.)
  - procédures, outils et ressources pour assurer la gestion et la maintenance: du réseau, du matériel, des logiciels, des accès, de l'usage, des coûts, du support utilisateur, de la performance, etc.
- accessibilité aux données : démarches administratives, protocoles de connexion, coûts éventuels, confidentialité, etc.
- localisation des applications et des bases de données (répartition des processus applicatifs entre serveurs, etc.)
- outils utilisés issus de logiciels libres
- environnement de travail
- architecture et fonctionnalités de l'application:
  - interface web (proxy, load balancing)
  - middleware (serveur web, php-fpm)
  - bases de données (stockage, réplication, performance)
- outils de sécurisation des transactions et des serveurs (certificats, authentification par clés, mots de passe, firewall, etc.)

- performances que le système doit supporter dans 90% des cas: temps de réponse utilisateur, outils de mesure des performances, disponibilité requise, etc.)
- reproductibilité et persistance de l'application
- évolutivité de la solution (possibilités)
- maintenance du service

3.3 Autres contraintes

- documentation requise
- méthode d'analyse (performance, risques)
- évolutivité de la solution (coût)
- plan de formation des utilisateurs et des gestionnaires
- support utilisateurs

4 Déroulement du projet

4.1 Planification

Les grandes phases du projet seront les suivantes:

4.1.1 Phase 1 : Initialisation du projet

Fin de rédaction du cahier des charges, choix techniques

Calendrier : **mai 2022**

4.1.2 Phase 2 : Analyse et conception

Conception globale de l'application : analyse fonctionnelle, modélisation

Calendrier : **juillet 2022**

4.1.3 Phase 3 : Développement

Evaluation à l'aide du cahier des charges en cours

Calendrier : fin **novembre 2022**

4.1.4 Phase 4 : Tests d'intégration

Intégration de l'ensemble des développements dans l'environnement de test

Calendrier : **décembre 2022**

4.1.5 Phase 5 : Documentation et présentation

Documentation de l'application + présentation aux utilisateurs

Calendrier : **mars 2023**

4.2 Organisation et suivi

La phase d'initialisation du projet a été soumise et validée par la commission paritaire en date du 28/04/2022.
L'ensemble des activités introduites dans la planification des tâches sera discuté et suivi par Thomas Keutgen en tant que coach et par Freddy Gridelet en tant qu'évaluateur. RHUM sera régulièrement informé du suivi global du travail.
Le développement du projet nécessitera des interactions avec de nombreux groupes au sein de l'UCL comme détaillé dans les spécifications techniques.

4.3 Evaluation

La méthode d'analyse et les critères d'évaluation du projet sont soumis au règlement des examens d'avancement au grade d'informaticien-expert (document du 30 janvier 2006).

4.4 Spécifications techniques

Quelques pistes pour les points encore à définir :

- connexion réseau GB, agrégation de liens, etc. (contact : CISM, SRI)
- localisation serveurs (contacts: CISM pour DCIII)
- serveur(s) de stockage (contacts : ELI, ELIC, CISM, SGSI)
- serveur(s) d'application (contacts : ELI, ELIC, CISM)
- environnement de travail sous distribution virtualisée (OpenStack)
- infrastructure en tant que code (IaC) : reproductibilité aisée de la configuration (Ansible)
- persistance via système de gestion de versions (Git): utilisation de la forge GitLab de l'UCLouvain pour le suivi

- duplication middleware (serveur Web) pour charges élevées
- couche accès aux données locales puis via DB distribuée
- base de données SQL (MySQL, Postgresql ?) et réplication
- base de données haute performance Redis pour la mise en cache des requêtes de base de données (via la RAM)
- stockage Ceph pour des charges élevées, et possibilité d'utiliser le stockage d'objets compatible S3

4.5 Spécifications de réalisation

Quelques points encore à détailler :

- maquette ou démonstration fonctionnelle : objectifs, représentativité par rapport au projet complet, configuration, plan de travail, ressources, critères d'acceptation avant de poursuivre les travaux ;
- détails du calendrier des prestations : début, fin, phases, check-points ;
- planning de disponibilité des ressources mises à disposition (quantité, qualification, dates, lieux)
- méthodologie, plan et outils requis pour effectuer les tests :
  - fonctionnels, de performance et de qualité
  - de montée en charge du réseau et des applications, d'ergonomie
  - des fonctions de sauvegarde et de reprise

5 Analyse et conception du projet

5.1 Choix techniques

Une configuration Nextcloud simple est aisément déployable en utilisant une pile LAMP classique. LAMP est l'acronyme de Linux, Apache, MySQL et PHP. Ensemble, ils constituent un ensemble de logiciels pour la création d'applications Web, comme Nextcloud.

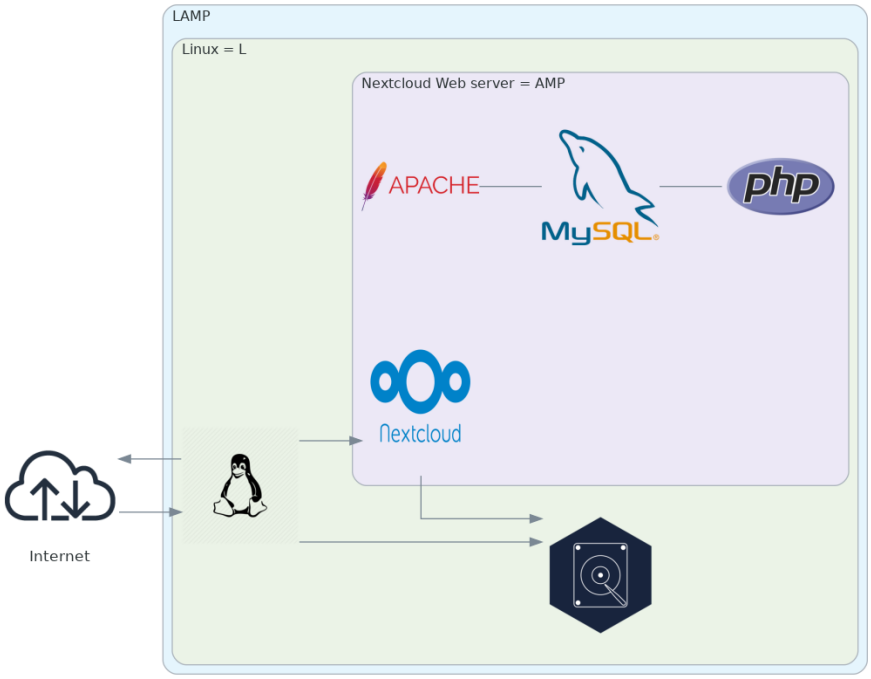


Figure 1: Achitecture Nextcloud basique

Ce type de déploiement permet de mettre en oeuvre un test du service rapidement.

5.1.1 Environnement

Dans le cadre d'un déploiement initial, l'application a été conteneurisée. La conteneurisation logicielle permet une gestion simplifiée des dépendances: une application et toutes ses dépendances sont placées dans une seule unité. Le système hôte ne doit pas se soucier de ces dépendances. L'application conteneurisée est donc indépendante de l'architecture ou des ressources de l'hôte. Elle est donc plus flexible et plus facilement distribuable.
Si cette conteneurisation apporte son lot d'avantages en développement et pour les tests de validation, son utilisation reste plus discutable dans le contexte d'une mise en production. Nous en rediscuterons

plus en avant dans ce projet.

Docker est la solution de conteneurisation la plus utilisée aujourd'hui. C'est un logiciel libre qui utilise une interface de programmation "Libcontainer" pour démarrer, gérer et arrêter les conteneurs. Il est basée sur le fonctionnement de LXC et y ajoute des capacités de niveau supérieur. Les conteneurs Docker peuvent servir d'images à d'autres conteneurs et le partage de conteneurs en public est possible via un service en ligne appelé Docker Hub. Il contient des images de conteneurs, ce qui permet aux utilisateurs de faire des échanges. Cela rend l'installation d'un conteneur extrêmement facile.

Le déploiement de Nextcloud dans un environmment conteneurisé est schématisé à la figure 2.

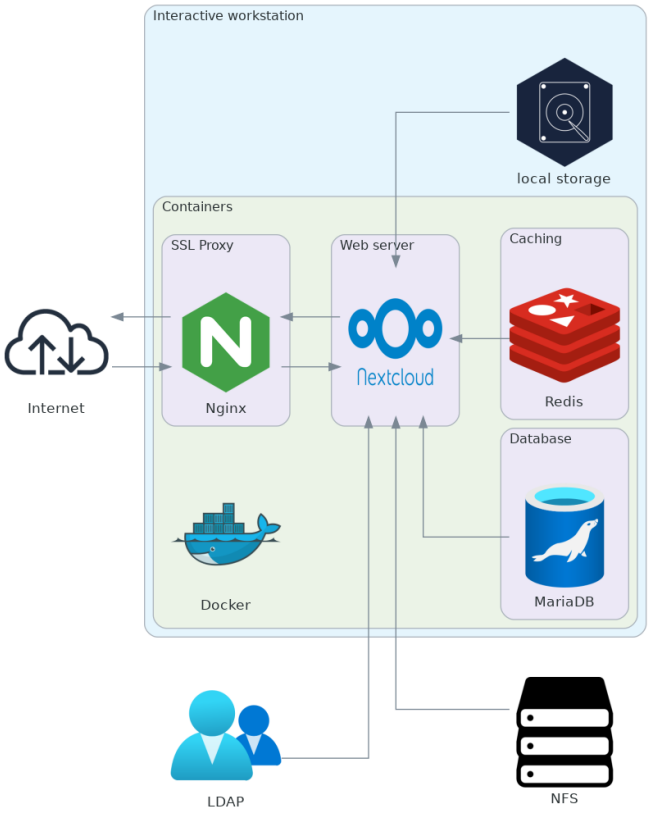


Figure 2: Achitecture Nextcloud améliorée

A présent le but est de déployer l'application dans un environnement virtualisé centralisé. Le détail de ce déploiement est développé dans la section "méthodologie" ci-après (5.2).

5.1.2 Compétences et expertises

La réalisation de ce projet nécessite l'acquisition d'une nouvelle expérience dans le développement de plates-formes et services de bout en bout, assumant la responsabilité non seulement de la conception de la solution, mais également de son bon fonctionnement en production.

Dans cette gamme de compétences et d'expertises techniques spécifiques, certaines me sont déjà familières :

- système d'exploitation Unix/Linux (administration et maintenance)
- langages de programmation tels que PHP et Python
- langages d'interrogation de base de données tels que MySQL
- serveurs Web tels que Apache
- conteneurs tels que Docker et le framework d'orchestration Docker Compose

Comprendre techniquement les services offerts par le CISM est une condition préalable pour démarrer le projet. Cet apprentissage de certaines de leurs méthodes et compétences fait essentiellement référence aux opérations de développement DevOps.
DevOps est la contraction des mots "developpers" et "ops". Il s'agit de l'un des frameworks les plus populaires visant à l'unification du développement logiciel (dev) et de l'administration des infrastructures informatiques (ops) afin de faciliter le développement, le test et la livraison des logiciels.
Une chaîne d'outils DevOps est une combinaison d'outils participant au développement, à la distribution et plus généralement à la gestion d'applications logicielles tout au long de leur cycle de développement.
Les logiciels virtuels et l'informatique sans serveur minimisent la dépendance matérielle et facilitent le processus de développement et d'évolutivité. Ayant une expertise pratique dans les conteneurs, je souhaite compléter mes connaissances dans la bonne utilisation des environnements virtualisés.

Je souhaite souligner une différence d'apprentissage technique en définissant ici les deux terminologies pour mon projet: la *compétence* et l'expertise.
L'acquisition d'une nouvelle compétence consiste en une bonne compréhension de la technologie sous-jacente en vue de son utilisation appropriée dans le cadre du travail. L'expertise implique un développement et une connaissance plus poussée de la compétence acquise, que ce soit pour son importance d'utilisation ou pour son rôle clé dans la conception du produit.

Je vais donc d'abord acquérir de nouvelles *compétences* et expertises techniques via l'apprentissage de différents outils utilisés en DevOps.

Les serveurs d'approvisionnement

- *Cobbler*

L'approvisionnement est le processus de préparation de l'équipement. Cobbler est un serveur d'installation Linux permettant une configuration rapide des environnements d'installation: DNS, DHCP, mises à jour de paquet, gestion de l'alimentation, orchestration de la gestion de la configuration, etc.
Cobbler est une *compétence* utile à acquérir ici afin d'intégrer efficacement un déploiement possible du produit hors d'un environnement virtualisé.

Les frameworks d'orchestration

- *Vagrant*
- *OpenStack*

Dans le contexte de la virtualisation, l'orchestration est le résultat de l'automatisation et la gestion de déploiement de systèmes et de services.

Capable de déployer, gérer et faire évoluer automatiquement des environnements de développement virtuels, Vagrant peut-être qualifié de structure ("framework") d'orchestration. C'est un logiciel libre et open-source pour la création et la configuration de machines virtuelles. Cet outil orchestre l'utilisation des ressources, la gestion des défaillances, la disponibilité, la configuration, l'état souhaité et l'évolutivité de machines autour de logiciels de virtualisation comme VirtualBox.
Vagrant est une *compétence* à acquérir, nécessaire durant tout le processus de développement et de test.

OpenStack est une plate-forme logicielle libre et open source permettant de déployer des infrastructures. Des serveurs virtuels ou autres ressources sont mis à disposition des utilisateurs par ce biais. La technologie possède une architecture modulaire avec plusieurs composants reliés les uns aux autres qui contrôlent divers ensembles matériels pour déployer les différentes ressources des machines virtuelles telles que la puissance de calcul, le stockage ou encore le réseau. Un tableau de bord est disponible, donnant aux administrateurs le contrôle tout en permettant à leurs utilisateurs de provisionner des ressources via une interface Web.
Au-delà de la fonctionnalité standard d'infrastructure en tant que service (IaaS), des composants supplémentaires assurent l'orchestration, la gestion des pannes et la gestion des services, entre autres services, pour garantir une haute disponibilité des applications utilisateur.

Le produit de ce projet sera déployé dans l'instance Openstack du CISM. Il s'agit d'acquérir une *compétence* simple dans son utilisation en mode utilisateur uniquement.

L'intégration d'outil de gestion de code source

- *GitLab* (via la forge UCLouvain)

La gestion de versions (Git) et l'utilisation d'une plateforme d'hébergement de projets est une pratique de base du DevOps. Gitlab, open source et collaboratif, permet d'héberger et de gérer des dépôts Git et ainsi de mieux appréhender la gestion des versions des codes sources.

L'intégration continue et le déploiement continu (CI/CD)

- *Gitlab CI*

L'approche CI/CD (Continuous Integration/Continuous Delivery-Deployment) permet d'augmenter la fréquence de déploiement des applications grâce à l'utilisation de l'automatisation au niveau des étapes de développement.
L'intégration continue consiste à vérifier à chaque modification de code source que le résultat des modifications ne provoque pas l'émergence de nouveaux problèmes dans le logiciel développé. Cela permet d'automatiser l'exécution des suites de tests.
Le déploiement continu est une approche visant à produire des logiciels dans des cycles courts. Le but est de développer, tester et déployer un logiciel plus rapidement en adoptant une approche incrémentale, simple et répétable des modifications en production.
Une *compétence* GitLab CI va me permettre d'automatiser les builds, les tests et les déploiements du produit.

La mise en place de l'IaC

- Ansible
- Salt

La mise en place de l'IaC (Infrastructure en tant que Code) dans ce projet sera réalisée par l'automatisation et la gestion de configuration avec l'acquisition d'une expertise dans les outils Ansible et Salt.
Ce sont deux outils dont l'objectif est de définir un état, et de le conserver, des ressources d'une infrastructure informatique : les serveurs, les réseaux, les utilisateurs/groupes, les logiciels et la sécurité. Ansible et Salt peuvent être complémentaires: le premier avec une approche plus impérative (c'est-à-dire procédurale ou comment l'infrastructure devrait être modifiée: méthode "push"), le

second avec une approche davantage déclarative (c'est-à-dire fonctionnelle ou ce que la configuration devrait être: méthode "pull").

L'équilibrage des charges du cloud

- *HAproxy*

Un équilibreur de charge permet de répartir le trafic sur plusieurs serveurs, ce qui facilite la gestion. Il permet également au réseau d'être plus résilient.
HAProxy est un équilibreur de charge open source, capable d'équilibrer n'importe quel service basé sur TCP. Il est couramment utilisé pour équilibrer le flux des requêtes HTTP et peut aider à résoudre les problèmes de trafic. Acquérir cette *compétence* dans la mise en oeuvre de ce proxy est donc indispensable.

Les outils de monitoring et alerting

- *Zabbix*
- Prometheus et le service Grafana

Le CISM a déployé Zabbix comme logiciel de monitoring pour l'ensemble de son infrastructure. Je souhaite donc acquérir une nouvelle *compétence* avec cet outil afin de pouvoir le comparer dans un second temps avec un concurrent: Prometheus. Ce dernier, également libre et open source, est plus récent et est recommandé pour le monitoring cloud et OpenStack alors que Zabbix est plus orienté infrastructure matérielle. Développer une expertise Prometheus (petit déploiement et configuration pour ce projet) serait un atout.
Prometheus est intimement lié à Grafana qui permet la visualisation: cet outil de restitution va récupérer les métriques pour construire des tableaux de bord graphiques sur une période donnée.

D'autres sujets doivent être abordés comme la gestion d'architecture de bases de données relationnelles: supervision (monitoring), sauvegarde/restauration et haute disponibilité. La connaissance d'un système de stockage de données en mémoire de type NoSQL est également particulièrement importante pour ce projet.
Cela nécessite donc l'acquisition des technologies ci-dessous.

La création de cluster de bases de données

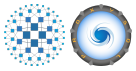
- *Galera Cluster*

Galera Cluster est une suite logicielle pour Linux permettant la mise en place et la gestion de clusters pour MySQL et MariaDB. C'est une surcouche du moteur de stockage InnoDB pour permettre une

sauvegarde flexible des données tout en offrant une sécurité maximale en cas de panne et une haute disponibilité. Une bonne configuration de la grappe permet de minimiser la latence occasionnée par la réplication.
Je souhaite acquérir une *compétence* de cette technologie en vue de son utilisation dans la stratégie de déploiement Nextcloud.

La gestion du monitoring et du load balancing

- *HAproxy*
- ProxySQL



ProxySQL est un serveur proxy compatible SQL qui peut être positionné entre une application et sa base de données. Il offre de nombreuses fonctionnalités: l'équilibrage de la charge entre plusieurs serveurs MySQL, la mise en cache des requêtes, etc.
Une *compétence* HAproxy est déjà nécessaire pour optimiser le trafic HTTP. HAproxy peut également être utilisé pour contrôler la charge de la grappe de bases de données. Mais développer une expertise avec ProxySQL peut-être très intéressant dans ce cadre: il est en effet particulièrement orienté SQL.

La mise en place d'un cluster de cache NoSQL

- *Redis et Sentinel*

Redis est un système de stockage de données en mémoire de type NoSQL (enregistrement en clé-valeur) rapide et persistant. Il s'exécute sur la mémoire vive (comme un système de cache utilisateur) et propose de bonnes performances pour des sites à trafic important, en évitant les accès disques, particulièrement coûteux.
Redis offre une haute disponibilité via le système distribué Redis Sentinel (surveillance de la grappe Redis, détection des défaillances, etc) augmentant ainsi la résilience et les performances.

L'utilisation d'un système de stockage objet S3

- *Ceph*

Il est impératif d'acquérir une *compétence* simple dans l'utilisation du stockage objet. Un stockage centralisé est adéquat pour profiter d'un système de haute disponibilité: gain sur l'espace de backup, scalabilité de l'espace de stockage, déduplication des données, etc.
Le CISM a déployé une solution de stockage Ceph dans leur centre de calcul. Ceph est un système de stockage distribué qui délivre à la fois des services de stockage en mode bloc (par exemple pour le stockage de VM), des services de stockage en mode objet (compatibles S3 et Swift) et des services en

mode fichiers (via CephFS). Ceph dispose de nombreux points d'intégration avec OpenStack qui en font la technologie de stockage la plus déployée dans le cloud libre.
Je compte donc utiliser la grappe CISM comme stockage objet primaire pour le service Nextcloud.

L'évaluation

- *Sitespeed.io*
- *Locust*

Un objectif supplémentaire du projet est la mesure et les tests de performances du produit. Il s'agit d'évaluer les performances du service Nextcloud dans les différents environnements (conteneurs Docker et machines virtuelles) en termes de performances de débit de mémoire, de performances de stockage en lecture/écriture, de test de charge et de mesure de la vitesse de fonctionnement.
Sitespeed.io est un ensemble d'outils open source qui facilite le suivi et la mesure des performances d'un site web. Ces outils sont utilisés en intégration continue (CI) pour détecter les régressions de performances Web, mais aussi pour suivre les performances en production (monitoring).
Locust est un outil de test de charge open source en Python scriptable et évolutif. Le test de charge est un type de test logiciel qui est effectué pour vérifier la tolérance et le comportement du système sous une charge attendue spécifique.

Une comparaison des performances avec le cloud institutionnel (SharePoint) sera également mis en oeuvre.

La sécurité

Enfin, la cybersécurité est un sujet transversal qui doit être partie intégrante du projet dès sa conception. Le produit n'échappe pas à cette règle et les compétences techniques en sécurité doivent être acquises: les contrôles fondamentaux tels que l'authentification multifacteur, la classification et la protection des données sont importants.

5.2 Méthodologie

5.2.1 Architecture de déploiement

Une architecture de déploiement est créée en mappant les blocs fonctionnels logiques d'une application sur un environnement informatique physique (ou virtuel). L'un des aspects de cette conception architecturale est le dimensionnement de l'environnement physique (en déterminant le nombre de machines) pour satisfaire les performances, la disponibilité, la sécurité et les autres exigences de qualité de service. Un noeud fait référence à n'importe quelle machine (machine physique, machine virtuelle, conteneurs, etc.).

L'objectif est de définir un déploiement de production avec une topologie multi-noeuds compatible haute disponibilité. Cela implique une installation de serveurs applicatifs/web Nextcloud sur plusieurs noeuds derrière un équilibreur de charge. Cela permet de s'assurer que chaque composant est entièrement redondant et peut tomber en panne sans interruption de service. De même les sauvegardes s'opèrent sans interruption de service.

Une telle exigence de service peut se traduire par une architecture de déploiement comme illustré à la figure suivante.

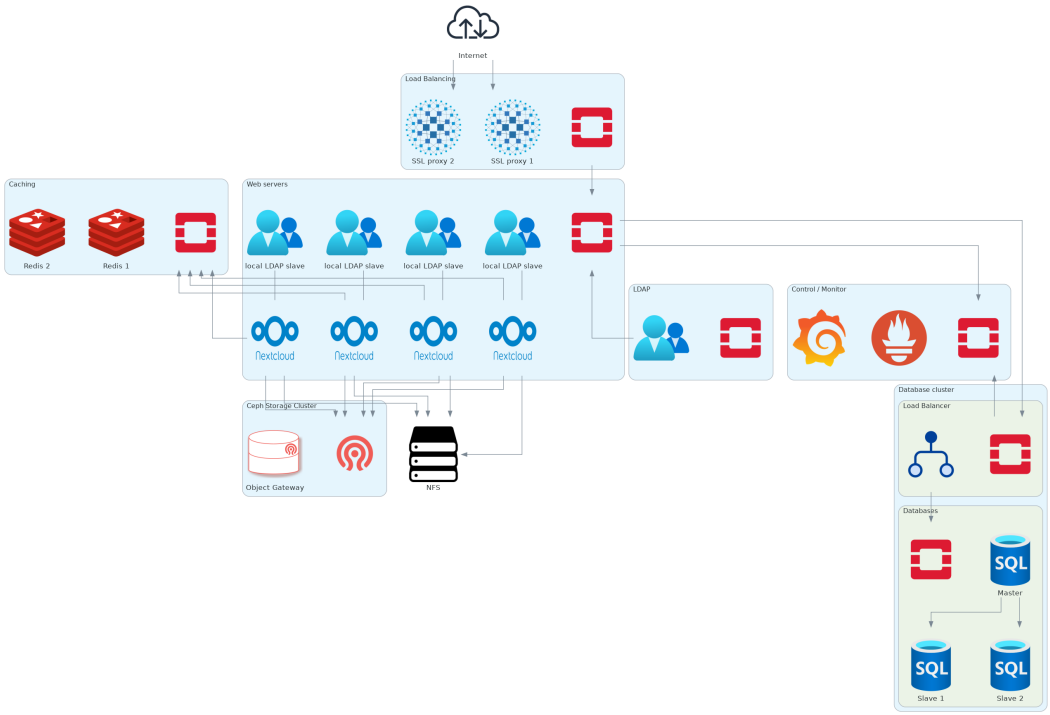


Figure 3: Achitecture Nextcloud haute disponibilité

Cette architecture à haut niveau de disponibilité doit assurer de bonnes performances au-delà du millier d'utilisateurs. La taille du stockage n'est a priori pas limitée puisque dépendante de l'évolutivité de la grappe Ceph via le magasin d'objets compatible S3. Au niveau applicatifs web, 4 à 10 serveurs pourraient être envisagé en fonction du résultat des tests d'intégration et de performance.

Concernant la grappe de base de données, nous pouvons envisager trois serveurs ou plus derrière un équilibreur de charge comme illustré à la figure ci-dessous.

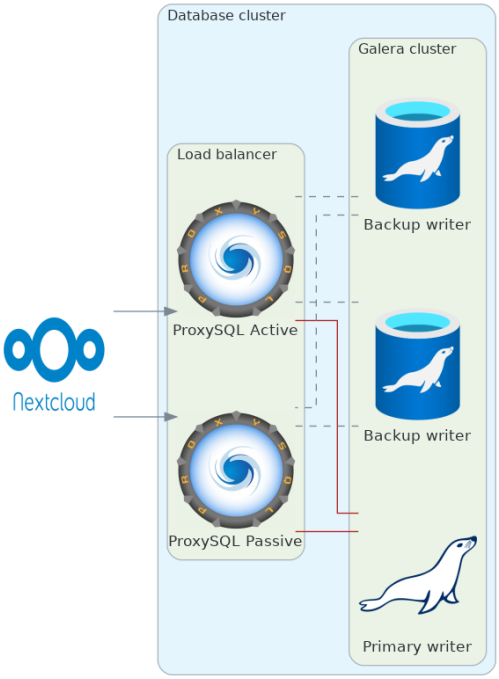


Figure 4: Grappe de bases de données équilibrée

5.2.2 Environnement de développement

La mise en place de l'achitecture ci-dessus sera opéré avec Vagrant dans un premier temps avec un déploiement simplifié. Cette phase de développement permettra la création des roles Ansible (IaC) suivant:

- déploiement de l'instance Nextcloud
  - support de Centos 7, openSUSE Leap 15.4, Ubuntu 20.04, Rocky Linux 9.0
  - intégration de Nginx ou Apache
- déploiement d'un serveur Redis simple (Centos 7, Rocky Linux 9.0)
- déploiement d'une serveur mariadb simple (Centos 7, Rocky Linux 9.0)

- déploiement d'une instance HAProxy simple (Centos 7, Ubuntu 20.04)
- déploiement d'une instance Prometheus
  - Prometheus: event monitoring and alerting (Centos 7, Ubuntu 20.04)
  - Node exporter: monitoring host metrics (Centos 7, Ubuntu 20.04)
  - Grafana: interactive visualization (Centos 7, Ubuntu 20.04)

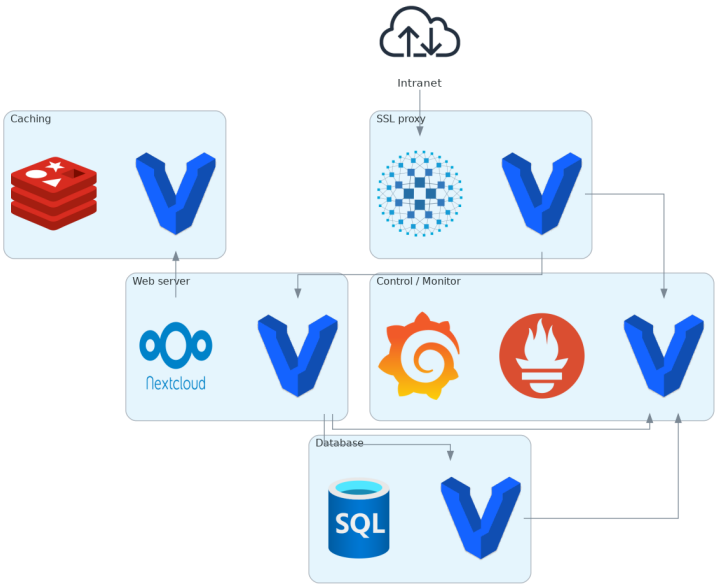


Figure 5: Environnement de développement simplifié

> Ressources : https://gogs.elic.ucl.ac.be/pbarriat/Brevet/src/master/dev/provisioning/ansible

Dans un second temps, les rôles Ansible ci-dessus seront enrichis:

- déploiement de multiples instances Nextcloud
- déploiement d'un cluster Redis
- déploiement d'une cluster Galera avec LoadBalancing
- déploiement de HAProxy avec LoadBalancing

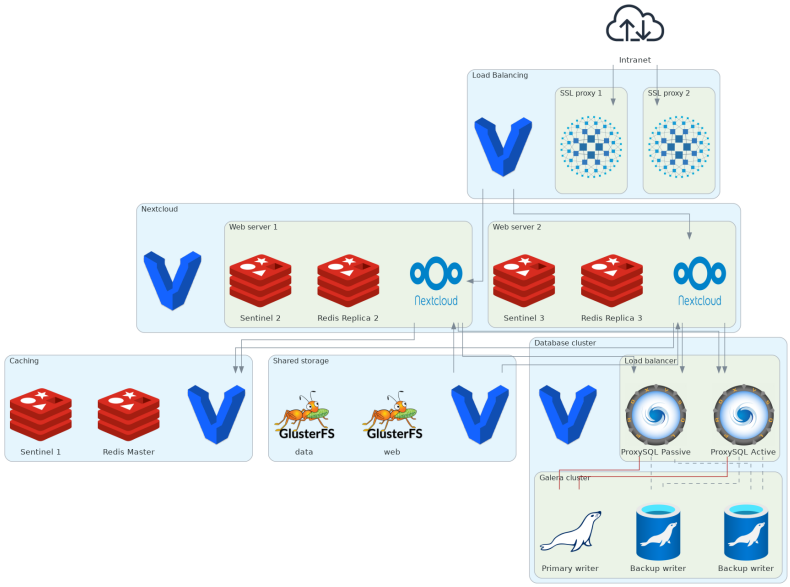


Figure 6: Environnement de développement avancé

Nextcloud:

- https://www.slideshare.net/Severalnines/tips-to-drive-maria-db-cluster-performance-for-nextcloud

Redis:

- https://redis.io/docs/management/sentinel/
- https://severalnines.com/blog/redis-high-availability-architecture-sentinel/
- https://nicoll.io/posts/ha-cache/

Galera & ProxySQL:

- https://www.digitalocean.com/community/tutorials/how-to-optimize-mysql-queries-with-proxysql-caching-on-ubuntu-16-04-fr
- https://mydbops.wordpress.com/2018/08/20/proxysql-series-percona-cluster-mariadb-cluster-galera-read-write-split/
- https://mysqldb-info.blogspot.com/2019/05/load-balancing-pxc-with-proxysql.html
- https://mysqldb-info.blogspot.com/2019/08/make-proxysql-for-high-availability.html

- https://mydbops.wordpress.com/2018/02/19/proxysql-series-mysql-replication-read-write-split-up/

5.2.3 Environnement de test

… la même chose mais dans **OpenStack** avec :

- intégration “LDAP slave” dans les instances Nextcloud
- intégration “Ceph” dans les instances Nextcloud
- intégration des montages NFS

5.2.4 Evaluation du déploiement

5.2.4.1 Tests d’intégration

- Synchronisation de base et conflits
- Suppression simultanée du répertoire pendant l’ajout de fichiers
- Repartage
- Partage d’annuaire entre utilisateurs
- Partage de fichiers entre utilisateurs
- Partage de fichiers entre utilisateurs et groupes
- Partage de fichiers par lien
- Simulation avec différentes autorisations
- Tests ETag de propagation de partage entre groupes d’utilisateurs
- Synchronisation des montages partagés

5.2.4.2 Tests de performance

- Upload/Download de petits/gros fichiers
- Performances de montage partagées
- Tests de charge

5.2.4.3 Outils

- Test de vitesse TCP/UDP, génération de paquets
  - pktgen
  - iperf/jperf
  - scapy

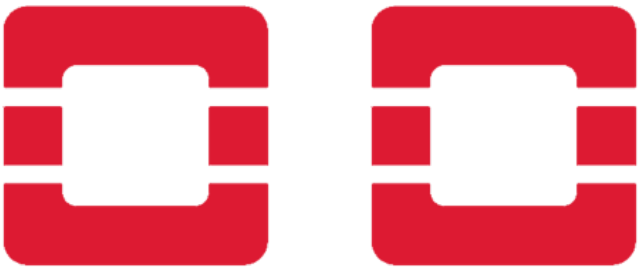
- Test de charges HTTP
  - owncloud/smashbox
  - owncloud/performance-tests
  - JMeter
  - Siege
  - load-test.io
  - flood.io
- Etat de l'art:
  - web-performance-testing
  - nextcloud features
  - deployment_recommendations

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a ante in mi ornare volutpat sed sit amet diam. Nullam interdum erat a augue faucibus, nec tempus tortor sagittis. Aenean imperdiet imperdiet dignissim. Nam aliquam blandit ex, sed molestie nibh feugiat ac. Morbi feugiat convallis semper. Ut et consequat purus. Fusce convallis vehicula enim in vulputate. Curabitur a augue arcu. Mauris laoreet lectus arcu, sed elementum turpis scelerisque id. Etiam porta turpis quis ipsum dictum vulputate. In ut convallis urna, at imperdiet nunc. Cras laoreet, massa lobortis gravida egestas, lacus est pellentesque arcu, imperdiet efficitur nibh dolor vel sapien. Sed accumsan condimentum diam non pellentesque.

Zabbix stocke les données recueillies dans une base de données relationnelle. Prometheus utilise sa propre base de données intégrée dans le processus backend (base de données non relationnelle). Zabbix utilise son propre protocole de communication basé sur TCP entre les agents et un serveur. Prometheus utilise HTTP avec des tampons de protocole (+ format texte pour une facilité d'utilisation avec curl).

Zabbix propose sa propre interface Web pour la visualisation. Prometheus offre un outil de base pour explorer les données recueillies et les visualiser dans des graphiques simples sur son serveur natif et offre également un constructeur de tableau de bord minimal, mais il est conçu pour être supporté par des outils de visualisation modernes comme Grafana.

Zabbix "pense" en termes de machines. Prometheus n'a pas cette restriction et conçoit de multiples entités: machines, services, centres de données, etc.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a ante in mi ornare volutpat sed sit amet diam. Nullam interdum erat a augue faucibus, nec tempus tortor sagittis. Aenean imperdiet imperdiet dignissim. Nam aliquam blandit ex, sed molestie nibh feugiat ac. Morbi feugiat convallis semper. Ut et consequat purus. Fusce convallis vehicula enim in vulputate. Curabitur a augue arcu. Mauris laoreet lectus arcu, sed elementum turpis scelerisque id. Etiam porta turpis quis ipsum dictum vulputate. In ut convallis urna, at imperdiet nunc. Cras laoreet, massa lobortis gravida egestas, lacus est pellentesque arcu, imperdiet efficitur nibh dolor vel sapien. Sed accumsan condimentum diam non pellentesque.

Vestibulum cursus nisi risus, sit amet consectetur massa suscipit nec. Sed condimentum, est id iaculis ornare, purus risus finibus felis, posuere congue est nibh eget dui. Maecenas orci erat, commodo auctor justo quis, vestibulum mollis ex. Vivamus sed bibendum turpis.

(a) caption a **(b)** caption b **(c)** caption c

Figure 7: Cool figure!