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

15 avril 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

Introduction

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.

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 sytématique: 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 les protocoles SSH et FTP. Comme OASIS, le stockage de masse du CISM est un système de fichier dans un réseau privé (réseau CECI: Universités francophones) en "modèle interne" (entièrement géré et hébergé au CISM).

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.

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.

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.

Produit du projet

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

Contraintes

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

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 février 2023 :** fin du premier brevet

Durée du projet : **10 mois** à partir de la validation du 28 avril 2022

Livraison estimée d'un produit minimum viable (MVP) : **décembre 2022**

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

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

Autres contraintes

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

Déroulement du projet

Planification

Les grandes phases du projet seront les suivantes:

Phase 1 : Initialisation du projet

Fin de rédaction du cahier des charges, choix techniques

Calendrier : **mai 2022**

Phase 2 : Analyse et conception

Conception globale de l'application : analyse fonctionnelle, modélisation

Calendrier : **juin 2022**

Phase 3 : Développement

Evaluation à l'aide du cahier des charges en cours

Calendrier : fin **octobre 2022**

Phase 4 : Tests d'intégration

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

Calendrier : **novembre 2022**

Phase 5 : Documentation et présentation

Documentation de l'application + présentation aux utilisateurs

Calendrier : **février 2023**

Organisation et suivi

La phase d'initialisation du projet est soumise à la validation de la Commission paritaire.
L'ensemble des activités introduites dans la planification des tâches sera discuté et suivi par Thomas Keutgen en tant que coach-évaluateur et par Hugues Goosse en tant que second 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.

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).

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
- échange de données inter-sites via protocole GridFTP (sans doute hors MVP)

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

Analyse et conception du projet

Choix techniques

L'objectif est la création d'une interface de gestion de données sous forme d'une application web service.

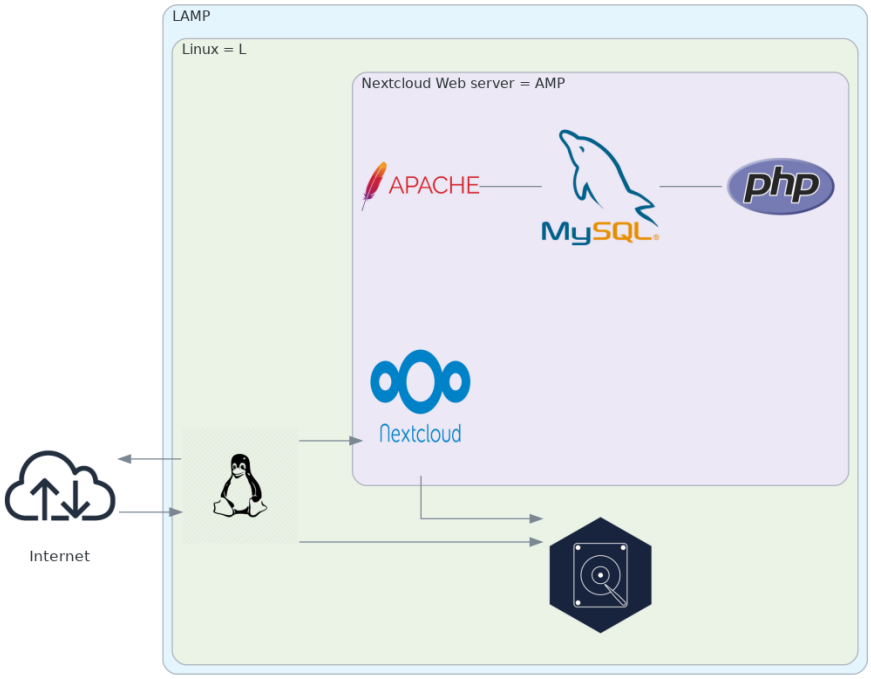


Figure 1: Simple Nextcloud achitecture

Framework

Un framework est un cadre qui permet de structurer le travail de développement grâce à un ensemble d'outils, une structure et des modèles prêts-à-l'emploi. Étant donné l'étendue des développements à effectuer pour concevoir une application web moderne, un framework est indispensable.

Django est un framework Backend Open Source développé en Python. Il a été spécialement créé pour réaliser des sites web puissants et de haut niveau. Il embarque tous les composants utiles, que ce soit la gestion de vues, l'authentification, le mapping objet-relationnel, une documentation détaillée, etc. Python est un avantage car c'est le langage le plus utilisé par les chercheurs en ELIC. En outre, les services IT de l'UCL utilisent également ce framework pour les nouveaux développements web. Une alternative solide serait Ruby on Rails (RoR). Il est le framework libre le plus populaire ces 5

dernières années, et a été conçu pour développer des applications web plus rapidement. Il permet aux développeurs de créer des fonctionnalités avec moins de code. Mais si RoR nécessite peu de configuration, il exige aussi plus de conventions. En outre, le niveau d'expertise pour se lancer est une barrière à l'entrée pour les débutants. Enfin Ruby nécessite des ressources serveur plus importantes que Django et sa technologie comme son utilisation sont en déclin.

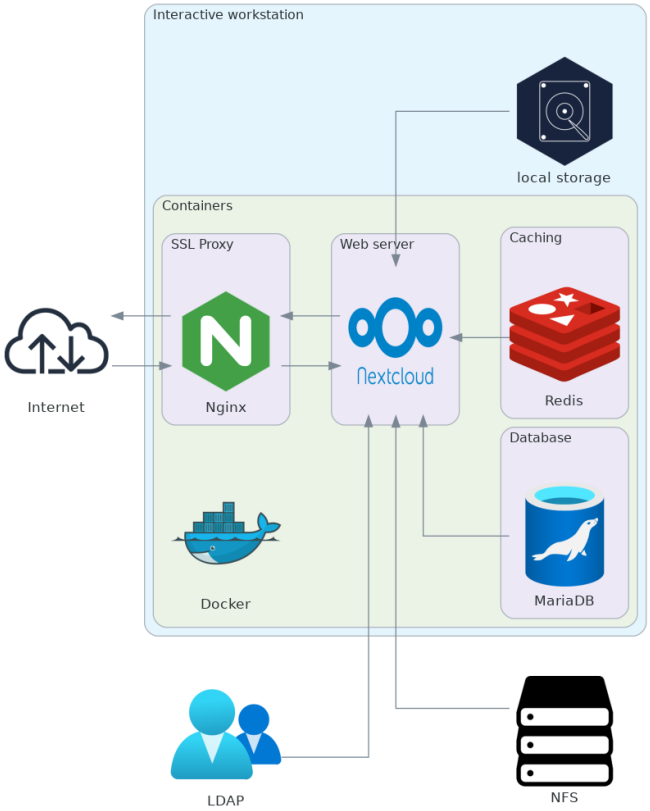


Figure 2: Improved Nextcloud achitecture

Python/Django sera préféré pour la conception du projet. C'est un framework Full-Stack - il est très facile de combiner Django et Angular par exemple - et tout clé en main : modèles, côté serveur, panneau d'administration pour configurer un site sans coder, etc. Il utilise, comme souhaité, le patron

de conception modèle-vue-contrôleur (MVC), c'est à dire que la structure du framework sépare les données (models) qui sont séparées des traitements (controller) qui sont eux-mêmes séparés de la vue (view/template). C'est également un outil idéal pour un projet collaboratif. Django étant très populaire auprès des développeurs web, de nombreux projets sont apparus autour du framework. Par exemple dans notre cas, Ncdjango est un ensemble d'outils de gestion de données et de géotraitement écrits en Python qui fonctionnent sur des données NetCDF.

Environnement

Cette application sera 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.

Outil de développement

PyCharm est un environnement de développement intégré utilisé pour développer en Python ainsi qu'avec Django. Il propose la possibilité de débugger en direct dans un conteneur Docker. Vagrant est un logiciel libre et open-source pour la création et la configuration des environnements de développement virtuel. Il peut être considéré comme un wrapper autour de logiciels de virtualisation comme VirtualBox.

Méthodologie

L'application sera donc standardisée MVC, c'est-à-dire selon une architecture classique à trois couches. La couche vue sera développée très simplement sur base de templates existants à l'UCL. Les couches traitement et modèle présenteront les cas de figure suivants: - données locales: traitement "on the fly" sur DB(s) locales 130.104 - données distantes - à posteriori (DB & protocoles connus) - à priori (infos de

structures à soumettre) - données distantes - indexées: traitement “on the fly” (batch process possible sur DB distante) - non-indexées: traitement différé (DB distante accessible en interactif uniquement)

Scénarios pour les données à posteriori et non-indexées : - téléchargement tiers + demande d’intégration aux DB - téléchargement à travers l’appli + intégration automatique aux DB locales

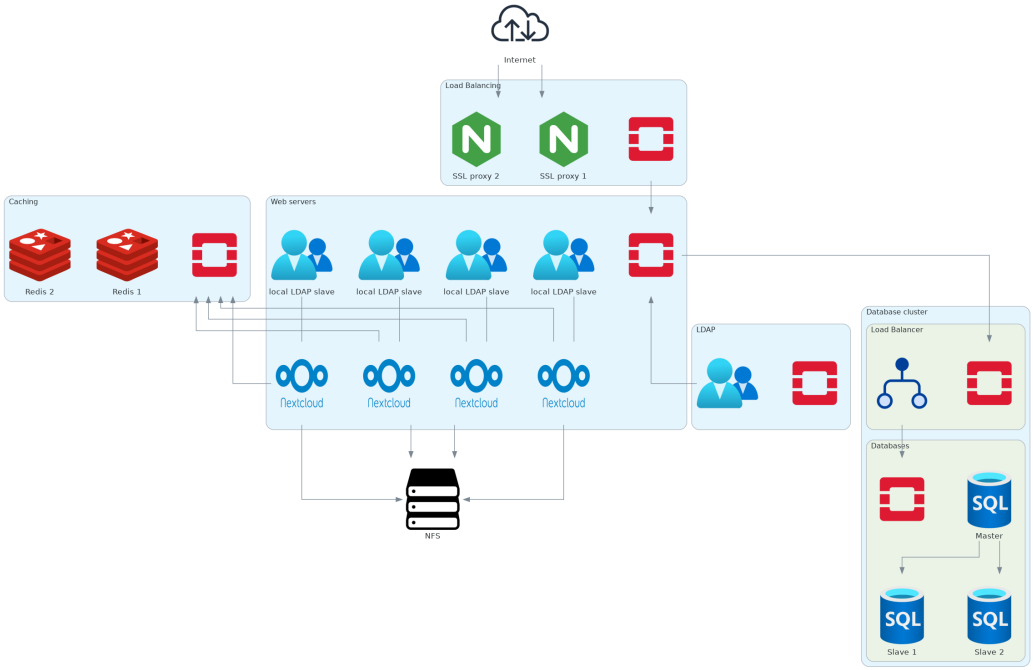


Figure 3: Large and efficient Nextcloud achitecture

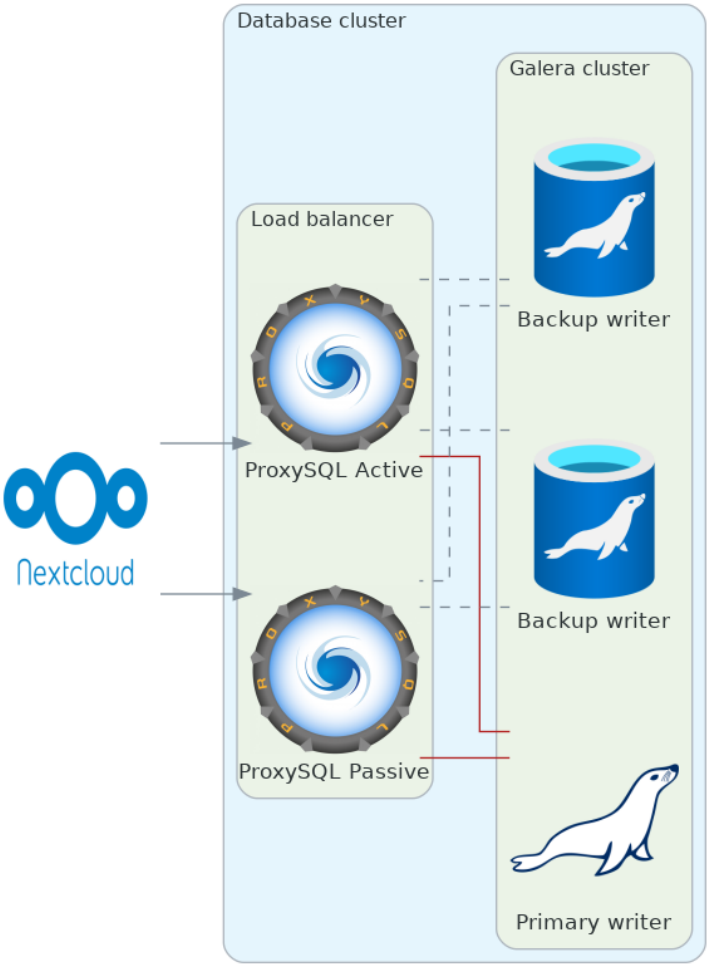


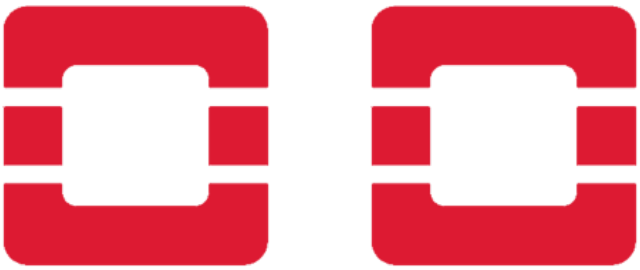
Figure 4: Database cluster

Comme nous souhaitons mettre à disposition des données pour quelles soient utilisées sur d'autres

plateformes et qu'elles puissent intéragir avec d'autres données, une architecture REST ("REpresentational State Transfer") semble appropriée ici. L'architecture REST est plus axée sur un modèle orienté ressources (les données, dans notre cas) que sur un modèle orienté fonctions. Elle imite la façon dont le web lui-même fonctionne dans les échanges entre un client et un serveur. REST constitue donc une méthode d'intégration efficace puisque le service à développer ici concerne surtout la récupération de données. Aussi, plutôt que de définir toute une API (interfaces de programmation d'application) personnalisée mieux vaut utiliser un standard de manipulation des données CRUD (Create, Read, Update, Delete : créer, lire, mettre à jour, supprimer), qui "correspond" aux opérations HTTP (HyperText Transfert Protocol) GET, PUT, POST et DELETE. Ce fonctionnement ne repose pas sur la seule utilisation de ces opérateurs, mais sur une combinaison avec des URI.

Le "Django REST framework" va nous permettre de créer plus facilement une API REST sur notre application Django.

Un interfacage avec Amazon S3 serait un atout supplémentaire.



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.

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum

dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat."

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Figure 5: caption y

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

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

Figure 6: Cool figure!