

Brevet principal

Complement technique

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

24 août 2022

Thématique du brevet principal

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

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.

Compétences et expertises techniques

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 "développeurs" 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: cela sera nécessaire au cours de l'entièreté 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 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*

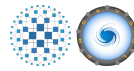


Galera Cluster est une suite logicielle pour Linux permettant la mise en place et la gestion de clusters pour MySQL et MariaDB. C'est une surcouche du moteur de stockage InnoDB pour permettre une sauvegarde flexible des données tout en offrant une sécurité maximale en cas de panne et une haute disponibilité. 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.

