Synthèse

L'outil OWASP ZAP a mis en lumière des vulnérabilités significatives dans la plateforme. Malgré une résistance aux attaques XSS, d'autres failles liées aux composants obsolètes et versions vulnérables sont préoccupantes. Il est essentiel d'agir rapidement pour corriger ces points et garantir la sécurité continue du site pour tous ses utilisateurs.

Partie 3 : Missions DevOps

C’est quoi le DevOps ?

Présentation de l’esprit DevOps :

Le terme DevOps est une combinaison de deux mots, à savoir Développement et Opérations. DevOps est une pratique qui permet à une seule équipe de gérer l'ensemble du cycle de vie du développement d'une application, c'est-à-dire le développement, les tests, le déploiement et le monitoring.
L'objectif ultime de DevOps est de réduire la durée du cycle de développement d’un système tout en fournissant fréquemment des fonctionnalités, des corrections et des mises à jour en étroite synchronisation avec les objectifs de l'entreprise.
DevOps est une approche de développement de logiciels qui permet de développer des logiciels de qualité supérieure rapidement et avec plus de fiabilité. Elle se compose de différentes étapes telles que le développement continu, l'intégration continue, les tests continus, le déploiement continu et le monitoring continue.

Institut de Science Financière et d’Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr
ISFA Institut de Science Financière et d’Assurances
GRANDE ÉCOLE D’ACTUARIAT & GESTION DES RISQUES

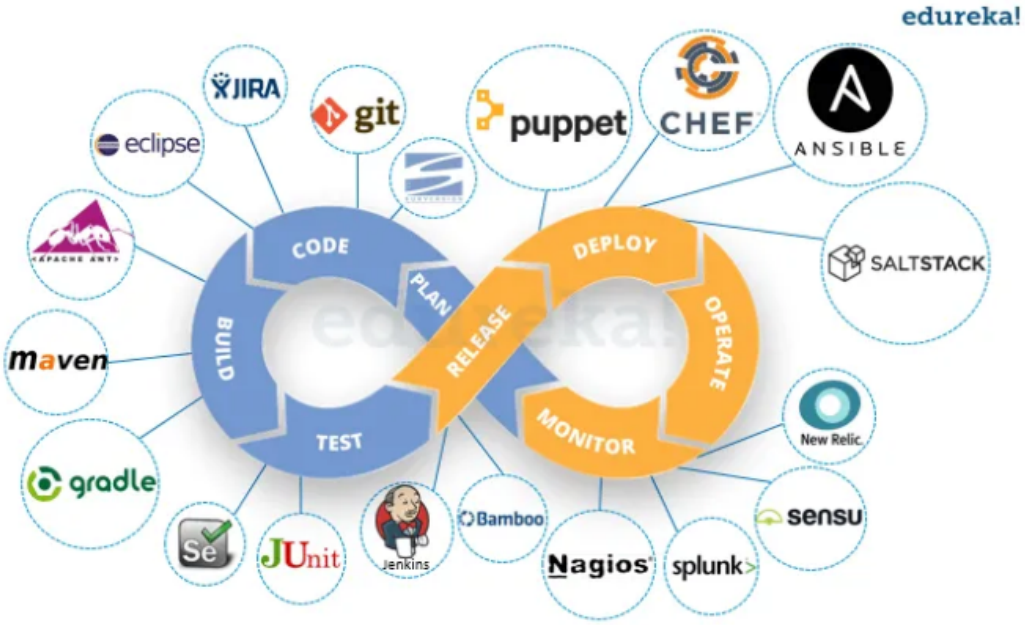


Figure 6 L'ensemble du cycle de vie DevOps

Pourquoi DevOps :

Avant l'émergence du mouvement DevOps, le domaine du développement logiciel était principalement caractérisé par deux méthodologies distinctes : le modèle en cascade et la méthodologie agile.

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

.

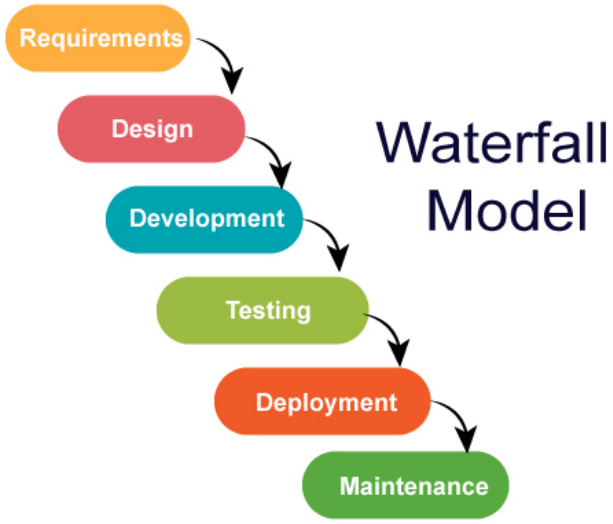


Figure 7 Modèle en cascade

Model en cascade :

Le modèle en cascade est un modèle de développement de logiciels assez simple et linéaire. Ce modèle suit une approche descendante.
Ce modèle comporte plusieurs étapes, à commencer la collecte et l'analyse des besoins. Il s'agit de la phase au cours de laquelle on obtient les exigences du client pour le développement d'une application puis on analyse ces dernières.
La phase suivante est la phase de conception, au cours de laquelle on prépare un plan du logiciel. Il s'agit de réfléchir à l'aspect que prendra le logiciel.
Une fois que la conception est prête, on passe à la phase de mise en œuvre ou l'implémentation de l'application. L'équipe de développeurs travaille ensemble sur les différents composants de l'application.
Une fois le développement de l'application terminé, elle est testée au cours de la phase de vérification. Divers tests sont effectués sur l'application, tels que les tests unitaires, les tests d'intégration, les tests de performance, etc.
Une fois tous les tests de l'application terminés, celle-ci est déployée sur les serveurs de production.

Enfin, vient la phase de maintenance. Au cours de cette phase, les performances de l'application sont contrôlées. Tous les problèmes liés à la performance de l'application sont résolus au cours de cette phase.

- Avantage :
  - Simple à comprendre
  - Facilite l'analyse et les tests
  - Permet d'économiser beaucoup de temps
  - Adapté aux petits projets dont le cahier des charges est clair et explicite

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

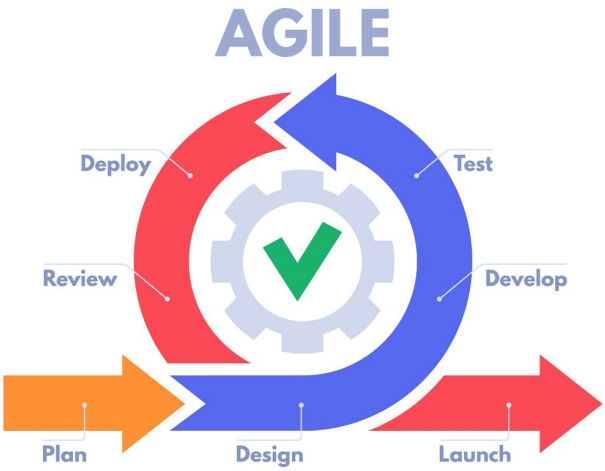
- Inconvénients :
  - Risquée et incertaine
  - Progression des projets difficiles à évaluer
  - S'avère inadaptée lorsque les exigences ont tendance à changer régulièrement
  - Toute modification s'avère compliquée une fois le produit en phase de test.
  - Le produit final n'est révélé qu'à la fin du cycle.

Ces défis rendent cette approche peu recommandée pour des projets de grande envergure ou complexes.

Modèle agile :

Dans la méthode Agile, une entreprise lance initialement l'application avec des fonctionnalités prioritaires lors de la première itération. Après ce lancement, les utilisateurs finaux ou les clients fournissent des retours sur ses performances. Suite à ces retours, des modifications nécessaires sont apportées à l'application, accompagnées de nouvelles fonctionnalités, puis une nouvelle version est lancée, marquant la deuxième itération. Ce processus est répété continuellement jusqu'à obtenir une qualité logicielle optimale.
La spécificité de la méthode agile est de s'ouvrir à l'imprévu et faciliter le suivi des projets. Il est possible, en cours de projet, de prendre en compte les dernières innovations, une modification imprévue du budget ou de nouvelles demandes clients.
Cette approche itérative est par ailleurs idéale pour mettre le client au centre du projet. En effet, ce dernier est présent à chaque étape. Par conséquent, au cours du projet, il est possible de s'adapter à ses nouvelles exigences voire aux modifications du marché.

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr
ISFA Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES



- Avantages :

  - Une réponse adaptative favorable aux changements de besoins rendant le développement plus flexible.
  - De plus, en rectifiant les erreurs dès les premières étapes, le processus devient nettement plus rentable.
  - Ce souci du détail précoce se traduit par une amélioration significative de la qualité du produit, le rendant largement exempt d'erreurs.
  - L'Agile favorise également une communication directe entre tous les acteurs d'un projet logiciel, ce qui renforce la collaboration et la clarté.
  - Idéalement adapté aux projets vastes et à long terme, nécessitant un minimum de ressources tout en étant simple à gérer.
  - L'approche agile est parfaite si vous souhaitez accroître la résilience de vos équipes et renforcer l'esprit d'équipe.

- Inconvénients :
  - La méthode Agile repose énormément sur la clarté des exigences du client, et pour des projets de grande envergure, il peut parfois être ardu de prévoir le temps et les efforts nécessaires.
  - Bien que l'Agile soit adapté à des projets de grande taille, sa nature flexible peut le rendre moins approprié pour des projets aux complexités extrêmes.
  - L'un des défis majeurs reste la documentation, qui, dans le cadre de l'Agile, peut parfois être insuffisante, augmentant ainsi les risques liés à la maintenabilité du logiciel à long terme.

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr
ISFA Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES

Comme indiqué précédemment, les différentes étapes telles que le développement continu, l'intégration continu, les tests continus, le déploiement continu et le monitoring continue constituent le cycle de vie DevOps. Examinons maintenant une à une chacune de ces étapes.

Le cycle de vie DevOps et les outils utilisés :

- Étape - 1 : Développement continu

  Outils utilisés : Git, SVN, Mercurial, CVS

  Il s'agit de la phase qui implique la "planification" et l'écriture du code du logiciel qui sera maintenu a l'aide d'outils de contrôle de version comme GIT. Ce processus de maintenance du code est connu sous le nom de gestion de « Source Code Management » ou gestion du code source.

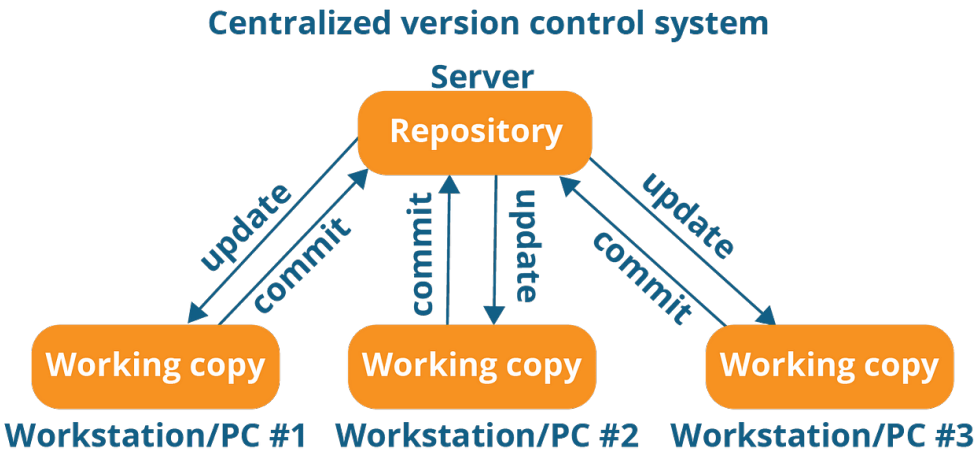


Figure 8 Gestion du code source

- Étape - 2 : Intégration continue

  Outils : GitLab CI , Jenkins, TeamCity, Travis

  Cette étape est au cœur de l'ensemble du cycle de vie DevOps en effet après une éventuelle modification ou intégration d'une nouvelle fonctionnalité cette

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr
ISFA Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES

dernières sont compilés ce qui permet de détecter rapidement les problèmes éventuels. La construction du code n'implique pas seulement la compilation, mais aussi la révision du code, les tests unitaires, les tests d'intégration et l'emballage.

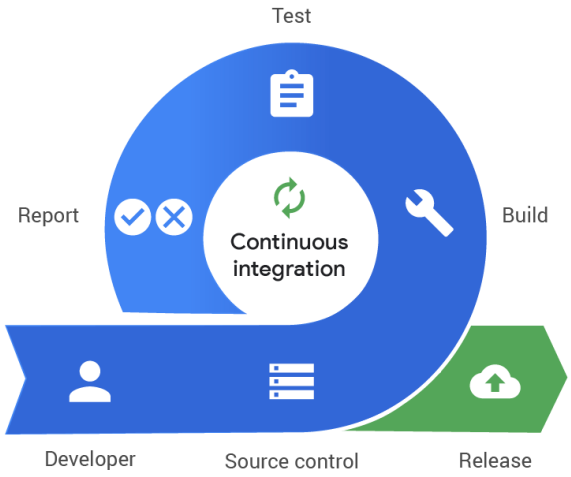


Figure 9 Integration continu

Le code supportant les nouvelles fonctionnalités est continuellement intégré au code existant. Ce qui permet la mise à jour de façon permanente et en douceur afin de de refléter les changements aux utilisateurs finaux.

- Étape - 3 : Tests continus

Outils : Jenkis Selenmium

C'est à ce stade que l'on teste en permanence le logiciel développé pour détecter les bogues à l'aide d'outils d'automatisation des tests. Ces outils permettent aux responsables de la qualité de tester minutieusement plusieurs bases de code en parallèle afin de s'assurer que les fonctionnalités sont opérationnelles . Dans cette phase on peut utiliser des conteneurs dockers pour simuler l'environnent de déploiement.
En outre, des outils tel que Selenium sont utilisés pour les tests d'automatisation et les rapports sont générés par TestNG. On peut automatiser l'ensemble de cette phase de test à l'aide d'un outil d'intégration continue Jenkins.

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr
ISFA Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES

- Étape 4 : déploiement continu

  Outils : Docker, Vagrant , Ansible , Kubernetes

  C'est l'étape où on déploit le code sur les serveurs de production ou communément appelé Continuous deployement CD et ou on fait la gestion de la configuration.
  La gestion de la configuration consiste à établir et à maintenir la cohérence des exigences fonctionnelles et des performances d'une application. En d'autres termes, il s'agit d'effectuer des déploiements sur les serveurs, de programmer des mises à jour sur tous les serveurs et, surtout, de maintenir la cohérence des configurations sur l'ensemble des serveurs.
  Les outils de conteneurisation jouent également un rôle crucial dans cette phase, en effet ces outils aident à assurer la cohérence entre les environnements de développement, de test, de mise en production.
  En outre, ils permettent également d'augmenter et de réduire rapidement la taille des instances.

- Étape - 5 : Contrôle continu

  Outils utilisés : Splunk, ELK Stack, Nagios, New Relic , Kubernetes Dashboard

  Il s'agit d'une étape très critique du cycle de vie DevOps au cours de laquelle on surveille en permanence les performances de l'application. Ici on enregistre les informations vitales sur l'utilisation du logiciel. On traite ensuite ces informations pour vérifier le bon fonctionnement de l'application. C'est au cours de cette phase que l'on résolve les erreurs système telles que le manque de mémoire, le fait que le serveur ne soit pas joignable, etc.

Le sujet de mon stage touchait principalement aux parties 4 (CD) et 2 (CI) et dans une moindre mesure la partie 5.
Ces parties seront traitées de manière détaillée.

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

Première étape des tâches DevOps : Focus sur déploiement continue (CD)

Tout d'abord il est crucial de parler de l'importance de la conteneurisation et de son impact en donnant une brève histoire du déploiement des applications.

Au tout début de l'informatique pour pouvoir déployer une application il fallait dédier un serveur **physique** entier pour chaque service de cette dernière ainsi on perdait énormément de ressources on avait alors un serveur dédié à la base de donne, un autre a la messagerie et puis un serveur web. Dans le cas ou notre application n'utilisait que 30 % des ressources de la machine on perdait 70 % du potentiel de celle-ci ce qui fait que cette opération est très couteuse. Il a fallu alors innover !

C'est là ou est apparu l'idée d'utiliser des machines virtuelles. Ceci a partiellement diminué les couts du déploiement. En effet au lieu de dédier des serveurs physiques on divisait les ressources de ces derniers pour avoir plusieurs serveurs virtuels sur chaque serveur physique. Dans ce cas, au lieu de dédier des serveurs physiques à chaque service on leurs dédiait des serveurs virtuels.

Machine virtuelle

Une machine virtuelle est une abstraction d'une machine physique (matériel). Ainsi, quelques machines virtuelles peuvent être exécutées sur une même machine physique. Comment cela fonctionne-t-il ?

Tout d'abord, l'hyperviseur est installé sur le serveur haute performance (machine physique). Un hyperviseur est un logiciel utilisé pour créer et gérer des machines virtuelles. Il existe de nombreux hyperviseurs tels que VirtualBox, VMware, Hyper-V, etc.
Après l'installation de l'hyperviseur, des machines virtuelles distinctes sont installées au-dessus de l'hyperviseur. Ces machines virtuelles sont configurées en fonction des besoins du serveur. Par exemple, considérons que le serveur d'application exploite 20 % de sa capacité, le serveur de base de données 20 % également, et le serveur web 10 %. Il est alors possible d'ajuster les ressources selon ces besoins spécifiques. Une fois le système d'exploitation installé, la machine virtuelle est prête à l'emploi. Après cela, on installe l'application sur le système, créant ainsi un environnement virtualisé. Si

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

cette approche diminue le gaspillage de ressources, elle a ses limites. La création d'une nouvelle machine virtuelle a des coûts associés, comme les frais de licence du système d'exploitation et le temps d'installation. En outre, garder le système à jour avec des patches nécessite une attention constante. Il est aussi à noter que le démarrage d'une machine virtuelle peut être lent. Et, même si l'idée est d'optimiser les ressources, chaque machine virtuelle consomme une part significative des ressources matérielles, que ce soit le CPU ou la mémoire.
Des outils comme Vagrant peuvent être utilise pour automatiser la configuration des machines virtuels.

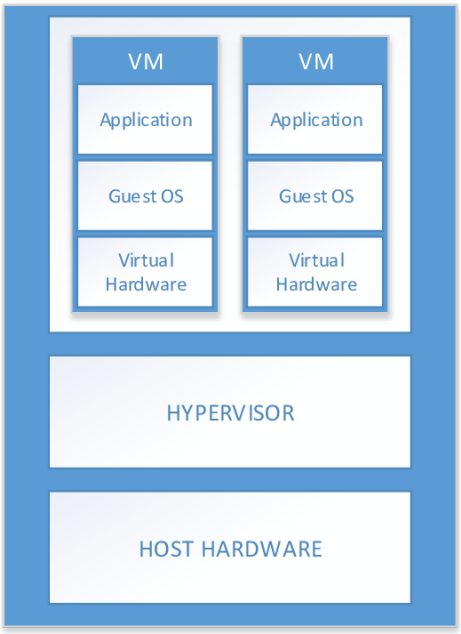


Figure 10 Schéma explicant la conteneurisation

Conteneurisation

Face aux contraintes des machines virtuelles, la conteneurisation s'est imposée comme une réponse adaptée. Elle permet aux applications de fonctionner dans des espaces isolés, appelés conteneurs. Ces derniers se distinguent principalement par leur légèreté. Là où les machines virtuelles nécessitent un système d'exploitation dédié, les conteneurs exploitent celui de la machine hôte, optimisant de fait l'utilisation des ressources. Cette centralisation génère des économies en matière de licences, de mises

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

à jour et de correctifs.
De plus, les conteneurs bénéficient d'un démarrage rapide, s'appuyant sur un système d'exploitation déjà actif. Ils se démarquent également par leur flexibilité, allouant des ressources selon la demande sans fixations prédéfinies.

Docker est notamment un outil de choix pour mettre en œuvre ces conteneurs.

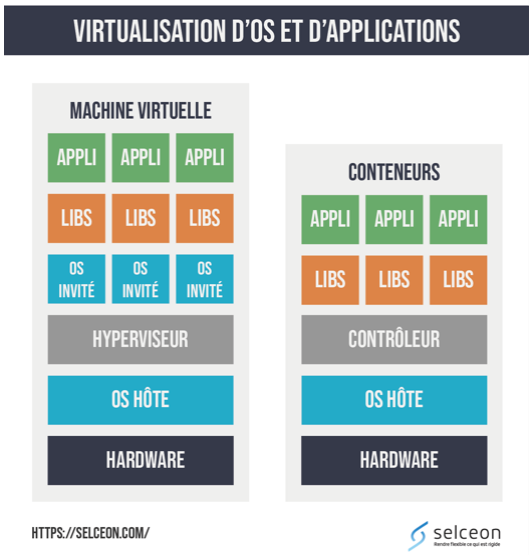


Figure 11 Différence entre conteneur et machine virtuelle

Orchestration :

Avec l'essor de la conteneurisation et des applications basées sur des microservices, est apparue une nécessité impérative : gérer, surveiller et orchestrer ces conteneurs de manière efficace. C'est dans ce contexte qu'est né le concept d'orchestration de conteneurs. Kubernetes, souvent désigné par son diminutif "K8s", s'est rapidement imposé comme le

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

leader incontesté dans ce domaine. Il offre un cadre permettant de déployer, mettre à l'échelle et gérer les conteneurs de manière automatisée. Au lieu de traiter chaque conteneur individuellement, Kubernetes envisage votre infrastructure dans son ensemble comme un système unifié. Cette révolution dans la gestion des conteneurs a par ailleurs favorisé l'émergence d'une nouvelle architecture, appelée architecture en micro-services, redéfinissant ainsi la manière dont les applications sont conçues et déployées.

De l'Unicité du Monolithe :

Durant une grande partie de l'histoire du développement logiciel, tout tournait autour de l'architecture monolithique. Dans cette configuration, toutes les fonctionnalités d'une application se regroupaient dans une unique et vaste base de code. Bien qu'initialement séduisante de par sa simplicité, cette approche se heurtait à de nombreux obstacles lorsqu'il s'agissait de faire évoluer, entretenir ou déployer, surtout quand l'application devenait plus complexe. C'est en réponse à ces contraintes et stimulé par la philosophie DevOps, axée sur

l'efficacité et la robustesse, que le monde du développement a commencé à se tourner vers une nouvelle méthode.

Vers la Modularité des Micro-Services :

Dans ce contexte, plutôt que de s'appuyer sur une vaste base de code, chaque fonctionnalité de l'application est envisagée comme un service à part entière, doté de ses propres responsabilités et interfaces API. Bien que ces services soient indépendants, ils travaillent ensemble pour créer une application fluide et harmonieuse. Par rapport au modèle monolithique, qui peut s'alourdir et devenir complexe à gérer avec le temps, les micro-services, malgré une mise en œuvre initiale plus exigeante, offrent à long terme une meilleure capacité d'adaptation, davantage de flexibilité, une maintenance allégée et une plus grande rapidité d'ajustement.

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr

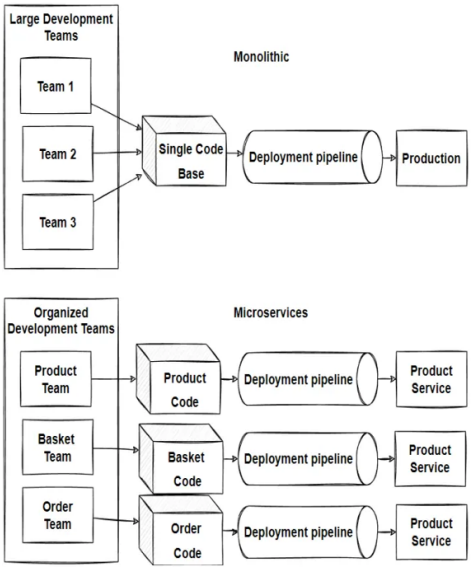


Figure 12 Comparaison monolithique et micro service

Dans le cadre de notre projet, l'application que nous envisageons de conteneuriser est basée sur cette moderne et agile architecture en micro-services. Ce choix garantit que, même si la mise en place initiale nécessite un investissement en termes de configuration et d'adaptation, les bénéfices à long terme en matière de performance, d'évolutivité et de maintenance en valent largement la peine.

Architecture de l'application :

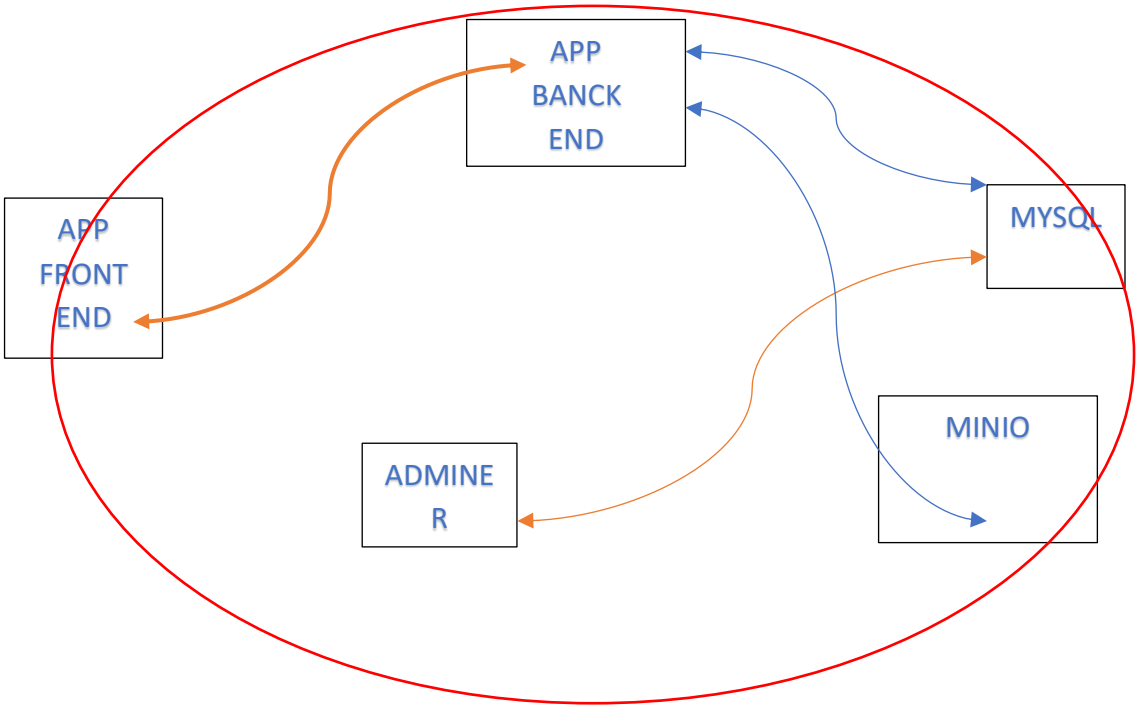
L'entreprise est actuellement en pleine phase de migration vers Kubernetes, une plateforme qui offre des avantages indéniables en matière de déploiement, d'orchestration et de gestion des conteneurs. Dans ce contexte, il m'a été confié la mission de conteneuriser deux applications critiques pour notre organisation. La première, nommée "Clycky", est un gestionnaire de restaurants. Elle offre une suite d'outils complète pour gérer les différentes facettes d'un établissement de restauration. La seconde application est "SIRH", qui est notre solution de gestion des ressources humaines. Elle est essentielle pour gérer et suivre les salariés de l'entreprise, de leur intégration à leur éventuelle sortie, en passant par toutes les étapes intermédiaires de leur carrière.

Toutefois, pour les besoins de ce projet et afin de garantir une approche méthodique, nous nous concentrerons uniquement sur l'application "Clycky".

Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1
50 avenue Tony Garnier 69366 Lyon cedex 7 - Tél : 04 37 28 74 40 - isfa.univ-lyon1.fr
ISFA Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES

Pour rappel, cette application est architecturée autour d'un backend conçu avec Laravel, et d'un frontend développé à l'aide d'Angular et de NodeJs. Le backend de "Clycky" s'appuie sur plusieurs services essentiels. On trouve en premier lieu MYSQL, qui sert de base de données principale pour l'ensemble des informations de l'application. Ensuite, ADMINER offre une interface d'administration pour cette base de données, permettant des opérations de maintenance et de gestion plus aisées. Enfin, MINIO est notre solution de choix en matière de gestion de stockage, indispensable pour traiter et conserver les nombreux fichiers et données associés à l'exploitation d'un restaurant.

La conteneurisation de "Clycky" présente plusieurs défis, mais aussi de nombreuses opportunités. En utilisant des conteneurs, nous pourrons garantir une meilleure isolation des services, une mise à l'échelle plus flexible et une portabilité accrue de l'application sur différents environnements.



Conteneurisation Partie Backend

Conteneurisation et génération de l'image APP-Backend

La première étape consiste à créer un fichier Dockerfile. Ce fichier détaillera toutes les étapes nécessaires pour la génération de l'image de l'api APP Backend et toutes ses dépendances. Une fois cela fait, nous obtiendrons une image Docker prête à l'emploi.