



### Nouveau buzzword ou tendance de fond ?

Le FinOps as Code apporte, comme toute nouvelle approche innovante, son lot de **bénéfices potentiels** en même temps que des **craintes de trop « simplifier » un domaine complexe**.



On s'intéresse ici à :

- 1) **définir** le combo FinOps avec les pratiques de développement logiciel telles que l'Infrastructure-as-Code (IaC),
- 2) **se pencher sur les bénéfices** liés à cette méthode visant à automatiser et à optimiser la gestion des coûts en utilisant des scripts et des outils pour surveiller, analyser et ajuster les dépenses cloud en temps réel,
- 3) **décortiquer de manière pragmatique les étapes de mise en œuvre** en s'appuyant sur des exemples clients concrets et récents.

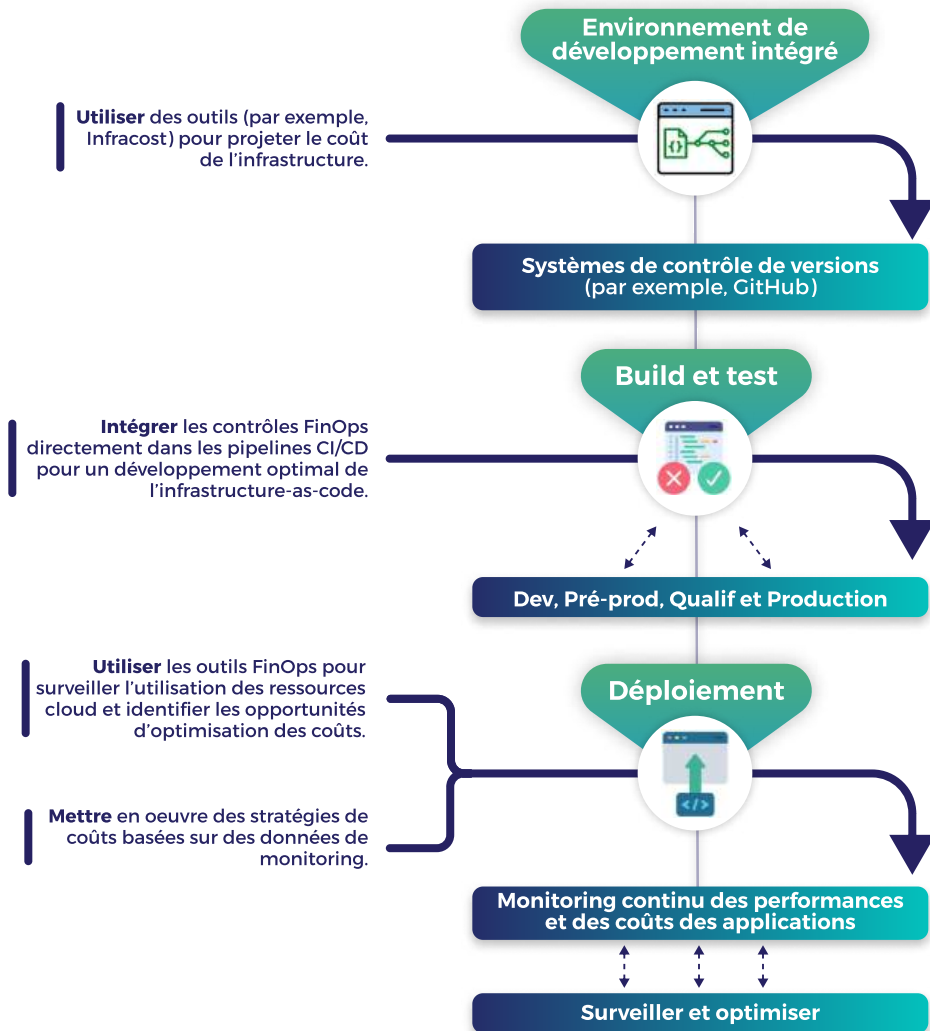
## 1 Qu'est-ce que le FinOps as Code ?

FinOps est la concaténation de **Fin** & **DevOps**. De facto, il est donc tout à fait naturel que nombre de bonnes pratiques qui nous sont familières dans le DevOps depuis toutes ces années, permettent de pleinement bénéficier au FinOps. Parmi les piliers du DevOps, on retrouve notamment : l'**Automatisation**, le « **Shift Left** », le **release management**, le **versioning**, l'utilisation de **pipelines d'intégration continue CI/CD...**

À lire sur le sujet : [Dossier Presse CIO : "FinOps breaks out the cloud"](#) publié le 12 Mars 2025

Et si l'on appliquait cette approche au FinOps ? C'est ce qui a conduit McKinsey, dans son article **"Tout est mieux en code : utiliser FinOps pour gérer les coûts du cloud"** (3 Février 2025), à schématiser cette approche :

**FinOps as code permet de coder les normes et instructions (guidelines) FinOps directement dans les pipelines de provisioning d'infrastructures.**



\*CI/CD : Continuous Integration/Continuous Deployment  
Intégration continue et déploiement continu.

Source : inspirée de [McKinsey & Company](#)

Quel est le constat de départ ? Les **\$675 milliards** dépensés dans le Cloud en 2024, vont – plus vite encore qu'annoncé par les analystes – très prochainement franchir la barre symbolique des **\$1 000 milliards**. Si l'on reprend notre ratio de gaspillage, il y a un consensus autour de **30%** en moyenne soit **\$200 Milliards**.



Dans la plupart des contextes clients, le plafond de verre arrive en général à **15 points de Savings réalisés** en « **Sanity FinOps** ». En effet, lorsque les mises en œuvre des recommandations **unused resources decommissioning, right-sizing** en cas d'under-usage, **scale up/scale down, start/stop, réservations (RIs, SPs, Spot), storage, sauvegardes (back-up policies...), négociations contractuelles EA, MACC, MCA...**, la mobilisation des équipes, la fréquence d'exécution des recommandations, le passage Walk, Crawl, Run sont les sujets largement couverts par la Fondation FinOps, les témoignages et articles, études des cabinets de conseils. On atteindrait ainsi, dans les cas de Maturity Run, jusqu'à **\$100 milliards** de Potential Savings, ce qui est déjà tout à fait remarquable.

Force est néanmoins de constater ainsi qu'au moins la moitié donc, les **\$100 milliards restants** (les 15 points restant, là où McKinsey évalue entre 10 et 20%\*, ce qui est cohérent) est relatif à la fois à **l'automatisation de l'exécution** et à **l'over-spending** afférent au **code applicatif**, qui nécessite **re-platforming, re-engineering**.

*\*cf. méthodologie de McKinsey : « Nous avons procédé à une étude complète des factures détaillées du cloud à l'aide d'un ensemble d'algorithmes permettant d'analyser les habitudes de consommation et les dépenses afin d'identifier les opportunités d'économies. Les résultats ont ensuite été associés à d'autres données auxiliaires, à des entretiens avec les principales parties prenantes du cloud et, si elles étaient disponibles, à des données détaillées sur la consommation de ressources issues d'un logiciel de surveillance et d'observabilité. Les données quantitatives et qualitatives ont fourni une image la plus fiable et holistique possible du portefeuille cloud. »*

## 2

# Quels sont les bénéfices du FinOps as Code pour les ingénieurs ?

## ► Principaux bénéfices du FinOps as Code :

**Automatisation et efficacité** Réduction des tâches manuelles grâce à l'automatisation des processus financiers « **Automate what can be automated** »

**Visibilité accrue** Meilleure surveillance des dépenses cloud, permettant des décisions plus éclairées « **If you can not measure it, you can not improve it** »

**Optimisation des coûts** Identification et réduction des dépenses inutiles ou inefficaces. « **Stop reserving Waste** »

**Agilité financière** Capacité à adapter rapidement les budgets et les ressources en fonction des besoins changeants. « **It is not necessary to do extraordinary things to get extraordinary results** »

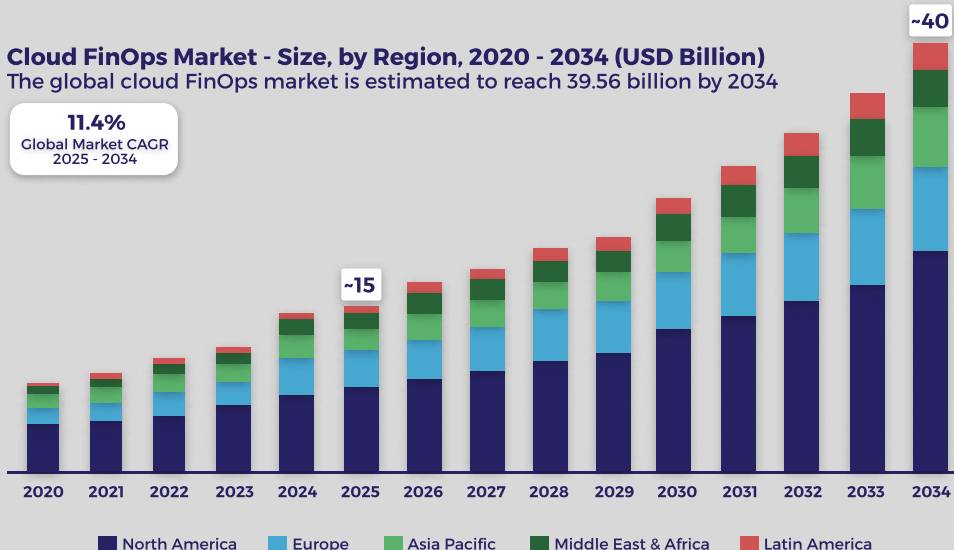


Le marché du Cloud FinOps évalué à ~\$15 Milliards pour 2025 a vocation à atteindre ~\$40 Milliards ces 10 prochaines années.

### Cloud FinOps Market - Size, by Region, 2020 - 2034 (USD Billion)

The global cloud FinOps market is estimated to reach 39.56 billion by 2034

**11.4%**  
Global Market CAGR  
2025 - 2034



FinOps as Code permettrait ainsi, selon le point de vue de Shanal Aggarwal de TechAhead, de fusionner les **opérations financières** avec l'**Infrastructure-as-Code (IaC)** spécifiquement et le **FinOps** (avec sa teinte « Fin + DevOps ») de manière plus globale, créant ainsi une **approche performante de la gestion des coûts cloud**. Cette intégration **rationaliserait les opérations, améliorerait le contrôle et favoriserait la collaboration**. Voici pourquoi elle pourrait changer la donne :



Source : [Techaheadcorp](#)

Par rapport à une approche traditionnelle et plus manuelle de FinOps, FaC présente plusieurs avantages pour les équipes d'ingénierie :



Les Cloud Providers introduisent en permanence de **nouveaux services** qui peuvent être plus rentables et plus performants (cf. [la newsletter sur Kubernetes](#)). FaC **peut alléger** la charge de la mise en œuvre de ces changements pour les ingénieurs et **utiliser du code** pour intégrer automatiquement les modifications dans leurs workflows en mode CI/CD pipeline automatisé. Exemple client : **cost savings de 6%** sur le total Spend des environnements non-productifs hors applications critiques grâce au **start/stop automatisé** en mode pipeline.

Autre exemple, **AWS** a introduit une **offre de stockage optimisée** qui était moins chère et plus compatible avec les machines virtuelles (VM) (source AWS 13 Janvier 2025 : « comparaison des types de volumes Amazon EBS gp2/gp3 »).

Une fois que les équipes FinOps ont traduit cette offre en code et l'ont déployée, les **modèles de stockage hérités** peuvent être **automatiquement mis à niveau** vers les nouvelles machines virtuelles.



## Offrir aux ingénieurs une visibilité et une responsabilisation des coûts en temps réel

Ce retour d'information en temps réel favorise une **culture de responsabilité financière** en mode « Shift Left ». Dans le cadre des processus de déploiement, les développeurs ont **plus conscience des impacts financiers** et accompagner l'entreprise dans les mises en place de contraintes budgétaires fortes **sans entrave sur le Business et la productivité**.

Les outils FaC offrent aux ingénieurs une **visibilité immédiate** sur les implications financières de leurs conceptions dans un environnement de développement **intégré**, afin qu'ils puissent prendre des **décisions plus rentables** avant de déployer le code. Des outils, de type Infracost, se sont multipliés, pour permettre de **sensibiliser les développeurs, AVANT** déploiement de leurs applications sur l'**impact financier** de leur Cloud Spend, renforçant leur **Accountability**.



Source : [Infracost](#)



## Réduire les interruptions et les tâches de « maintenance »

les interruptions et permettant aux ingénieurs de gagner du temps pour des tâches plus importantes.

L'approche traditionnelle du « **nettoyage annuel de printemps** » (la pratique consistant à effectuer périodiquement des activités de maintenance et de nettoyage sur une base de code) oblige souvent les entreprises à mettre des **ressources hors ligne** ou à **apporter des modifications importantes aux configurations**. Cela perturbe le travail et peut souvent affecter négativement les clients. L'utilisation du temps des ingénieurs est très « challengeable ». Au lieu de cela, les **équipes d'ingénierie pourraient utiliser FaC pour optimiser en permanence les ressources et les configurations**, réduisant ainsi



## Allocation et planification plus efficaces des ressources cloud

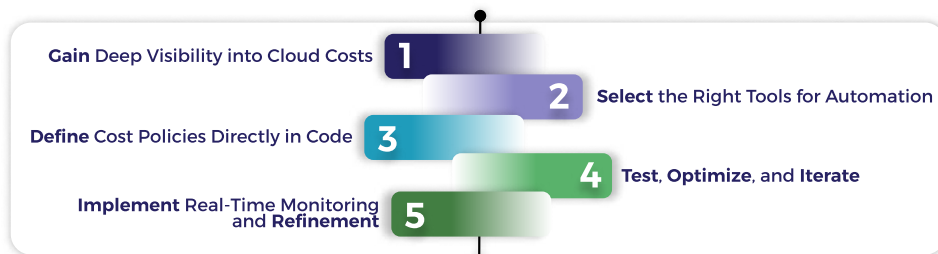
Les équipes d'ingénierie peuvent utiliser FaC pour identifier en temps réel les domaines dans lesquels les ressources cloud sont **utilisées de manière inefficace** (sous-utilisation, non utilisation, utilisation dévoyée...) et **automatiser la correction de ces inefficacités**. L'optimisation de ce processus libère des budgets d'exploitation afin que les fonds puissent être réinvestis dans des innovations et des développements de fonctionnalités supplémentaires.



## Identification des infrastructures oubliées ou inutilisées.

Que ce soient en raison du développement et de l'expansion rapides des applications ou dans le cadre des processus de mise hors service des systèmes, les composants d'infrastructure tels que les **adresses IP non allouées**, les **interfaces réseau**, les **sauvegardes** et les **snapshots**, les **disques « unattached »**, les **ressources orphelines**... peuvent continuer à entraîner des coûts pour l'organisation. FaC peut aider à identifier et, grâce à des **processus de correction automatisés**, à supprimer ces composants. Cela peut également résoudre les **écarts de refacturation** et les **problèmes de visibilité** sur les coûts associés aux actifs et aux applications qui auraient dû être mis hors service.

### 3 Quelles sont les étapes de mise en œuvre pour gagner en maturité ?

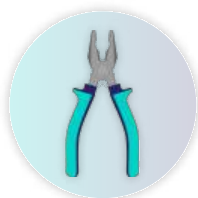


#### ► La boîte à outils FaC

Pour qu'il soit efficace et pérenne dans la durée, le déploiement du FaC ne doit pas **nécessairement être trop contraignant** malgré sa complexité apparente. Il nécessite que les entreprises comprennent clairement ce qui est nécessaire et s'engagent à intégrer la FaC dans la production. Pour faire du FaC une réalité, toujours selon McKinsey, les organisations doivent se concentrer sur quatre éléments :

#### ● Outils spécialisés

Une mise en œuvre efficace du FaC nécessite les bons outils pour appliquer les **Politiques** (par exemple, Open Policy Agent) qui valident les **scripts d'infrastructure en tant que code (IaC)** par rapport aux politiques prédéfinies écrites dans un langage déclaratif optimisé pour la définition de Politiques. D'autres outils, tels que ceux qui offrent des « **guardrails** » IaC similaires à la **Policy as Code (PaC)**, ciblent différents domaines de la gestion du cloud.



#### ● Catégorisation des Politiques

Les organisations peuvent établir une **catégorisation simple des Politiques** selon leurs fonctions, rôles et périmètres. Les **Politiques** d'information, d'avertissement et de blocage en sont un bon exemple. Il s'agit d'un ensemble de mesures de contrôle d'accès qui gèrent l'accès des utilisateurs aux ressources selon des conditions prédéfinies.



Les **Politiques d'information** fournissent à l'équipe des recommandations pertinentes, telles que l'exploitation d'un service de base de données natif dans le cloud au lieu d'une approche DIY. Les **Politiques d'avertissement** alertent les équipes, sans empêcher le provisioning, des bonnes pratiques qui ne sont pas respectées (par exemple, un groupe d'auto-scaling qui n'est pas défini). Les **Politiques de blocage** empêchent tout déploiement si certaines conditions ne sont pas remplies, comme le provisioning d'une infrastructure d'une valeur supérieure à \$100 000 pour un environnement de développement. Les bonnes pratiques incluent **l'exécution de ces Politiques** en Infrastructure-as-Code (IaC) à chaque pull request avant le passage en production.



### ● Policy Scripts

Les scripts de policy sont l'épine dorsale du FaC. La **GenAI** peut simplifier la création de Politiques en utilisant des prompts en langage naturel, en s'entraînant sur un large jeu de données, data set de contrôles et de Politiques FinOps et en mettant en œuvre la génération augmentée de récupération (RAG). Au début de cette transition vers le FaC, les entreprises peuvent **démarrer avec 10 à 15**

Politiques. Pour déterminer les **Politiques à créer**, elles peuvent examiner les principales sources de gaspillage, telles que la **maintenance d'environnements non productifs** (Dev, Test, Qualif...) en dehors des heures de travail (HNO et week-end, en tenant compte des fuseaux horaires pour les applications et infrastructures concernées) ou la définition de périodes de **conservation des logs inutilement longues**. Les équipes peuvent ensuite identifier et tester un cas d'utilisation cible qui peut être mis en œuvre en mode PoC sur un périmètre de ressources limitées.



### ● Gestion du changement

La mise en œuvre réussie du FaC nécessite une **gestion efficace du changement** (au sens DevOps, ITIL du terme, avec un outillage ITSM de type ServiceNow, JIRA d'Atlassian...). Au fur et à mesure que les Politiques sont déployées, il est essentiel d'**informer toutes les principales parties prenantes** des nouvelles Politiques et de s'assurer que les ingénieurs **comprennent ce qu'est le FaC, quelle est sa valeur et comment elle affecte leurs opérations quotidiennes**. Cela garantit une **adoption en douceur** et **minimise la résistance**, bien naturelle et humaine aux changements, en particulier lorsqu'il s'agit de la mise en place de **nouveaux processus**.

# Synthèse ●

## ● FinOps as Code (FaC)



**En conclusion**, comme l'explique bien Victor Boissière, le Head of SRE de Qonto, dans son approche FinOps en environnement Kubernetes, « **capturer les éléments de télémétrie et allouer les coûts de manière précise et fiable** aux équipes et aux applications dans une approche « **Kubernetes-native** » permet « d'empowerer » et de **responsabiliser les équipes** et en particulier les Dev dans la **compréhension de leur structure de coûts** et d'**optimiser le spending** de manière **pro-active** ».

Comme pour tout projet, bien définir l'objectif, extrêmement bien communiquer en transverse sur les **bénéfices pour chaque équipe**, à **chaque niveau de l'entreprise** et **mettre en place en mode « agile »** sur des **périmètres bien définis** et **en itératif** font partie des clés de succès des pionniers qui sont en train de mettre en œuvre FinOps as Code (FaC) et ont réussi ces premiers jalons, avant un déploiement à l'échelle. Gageons que **l'IA** accompagnée du bon niveau d'**automatisation**, d'**infra-as-code** va permettre de dynamiser et accélérer un **FinOps gagnant en maturité**.

   **Cloud FinOps** est un cabinet **100% dédié FinOps indépendant** intervenant en France et à l'international qui aide ses clients à « **booster la Valeur du Cloud** » et à vraiment « **contrôler leurs dépenses Cloud** ».



[Cloud FinOps](#)



[cloud-finops.io](https://cloud-finops.io)