

# INTRODUCTION AU COURS COMPLET SUR TERRAFORM

## Introduction

---

Bonjour à tous, vous êtes sûrement fan comme moi de l'automatisation ? Vous avez sûrement déjà entendu parler de l'Infrastructure as Code (IaC) et vous souhaitez en savoir davantage sur ce sujet ? Voire mieux encore pratiquer ? Alors vous pouvez être certain que vous êtes sur le bon cours. Sans plus attendre commençons !

## Public visé

Cette série d'articles est conçue pour les débutants ayant besoin de **comprendre l'Infrastructure as Code** et plus précisément **découvrir l'utilisation de l'outil Terraform à partir de zéro**. Ce tutoriel vous donnera une compréhension suffisante de la technologie, qui vous permettra plus tard d'atteindre des niveaux d'expertise beaucoup plus élevés.

## Prérequis

Ce cours ne demande pas forcément de prérequis que vous devriez au minimum avoir, car tout sera au maximum expliqué. Cependant si vous maîtrisez Docker, les services de base du cloud Provider AWS et que vous avez une compréhension générale de la virtualisation, alors il vous sera très facile de comprendre les différents concepts de Terraform et d'avancer rapidement sur votre piste d'apprentissage, mais ce n'est pas non plus indispensable !

Êtes-vous prêt pour découvrir des nouveaux horizons ? alors c'est parti !

## **Infrastructure as code (IaC)**

---

### **Naissance du besoin**

Traditionnellement, la gestion du cycle de vie des infrastructures était un processus manuel qui a souvent entraîné des incohérences environnementales. Par exemple, les environnements de test qui ne sont pas entièrement compatibles avec l'environnement de production sont un problème très courant chez les entreprises traditionnelles. Ce type d'incohérences environnementales crée souvent beaucoup de frictions pour la livraison des applications et des solutions, ce qui ralentit ou parfois bloque le rythme de l'innovation dans une organisation numérique.

D'ailleurs, on peut sûrement se le dire, l'automatisation de bout en bout d'un processus de développement d'applications est devenue courante. Bien que certaines parties de ce processus de tests étaient automatisées depuis un certain temps maintenant. Ce n'est pas forcément le cas pour la gestion et le provisionnement d'une infrastructure informatique qui était entre autres le seul domaine qui était un peu ignoré à l'ère de l'automatisation.

Certes, il existe de nombreux outils avec des tableaux de bord sophistiqués qui vous fournissent des mises à jour d'intégrité et effectuent la partie d'approvisionnement de l'infrastructure. Mais ce n'est qu'une partie de l'automatisation de l'infrastructure, ce qui signifie simplement répliquer les étapes plusieurs fois et reproduire plusieurs fois les sorties déjà produites sur plusieurs serveurs auparavant.

En réalité, ce n'est qu'une réplification pour réduire l'intervention humaine et le temps nécessaire pour le faire. Vous le savez sûrement mais le type d'application

produit de nos jours et les environnements informatiques de notre nouvelle ère, sont plus que jamais dynamiques qu'antérieurement. De ce fait, vous avez donc besoin de quelque chose qui non seulement évoluera plus rapidement mais qui maintiendra également la cohérence de vos infrastructures, et l'Infrastructure as Code pourrait bien être cette chose.

## Qu'est-ce que l'Infrastructure as Code (IaC) ?

Infrastructure as Code (IaC) consiste à remplacer les processus manuels et les procédures d'exploitation standard pour configurer les périphériques matériels et les systèmes d'exploitation par du code qui **gèrera et fournira automatiquement votre infrastructure**. Dans l'IaC, vous pouvez configurer et déployer ces composants d'infrastructure plus rapidement avec cohérence en les traitant comme si c'était une application.

Ainsi, chaque fois que vous devez configurer une infrastructure, vous n'avez pas besoin d'aller voir les administrateurs système, de formuler une demande, de créer un ticket et d'attendre qu'ils y participent. Au lieu de cela, vos développeurs et vos équipes d'exploitation peuvent facilement le faire en utilisant votre code.

## Pourquoi IaC?

Comme tout processus d'automatisation, les avantages évidents ici sont le **coût**, la **simplicité** et la **rapidité**. Parce qu'avec IaC, vos administrateurs système et vos développeurs peuvent **travailler plus efficacement** sur des tâches plus prioritaires. Vous pouvez répondre à leurs besoins en infrastructure de manière plus dynamique que jamais et vous pouvez littéralement faire tourner une configuration entière en exécutant un seul morceau de code. Augmentant ainsi la vitesse de vos processus

globaux, réduisant ainsi les efforts manuels et le temps nécessaire pour répondre aux besoins en infrastructure.

Avec l'laC, vos développeurs peuvent travailler de manière plus productive en raison de l'efficacité accrue et de la dépendance réduite vis-à-vis d'autres équipes. Ils peuvent lancer leurs propres environnements en fonction de leurs besoins. Avec un seul code, ils peuvent prendre en charge un certain nombre d'étapes qui seraient là sans l'automatisation, ainsi vous contribuez à garder vos environnements propres et organisés.

De plus, il s'intègre parfaitement à la philosophie DevOps, qui je vous rappelle est un changement culturel qui se concentre immédiatement sur la maximisation de la valeur commerciale en optant pour une meilleure communication, collaboration et rétroaction au sein et entre les équipes de développement et d'exploitation. Ainsi l'Infrastructure as Code (laC) est un élément-clé de la philosophie DevOps avec des avantages pour les équipes de développement et d'exploitation. Dans l'laC, le cycle de vie complet de l'infrastructure, y compris l'orchestration, l'approvisionnement, la configuration, la surveillance, l'auto-réparation peut être gérée de manière automatisée.

Un autre avantage est sur le plan de la sécurité, avec l'laC vous pouvez documenter et **suivre les modifications apportées** à votre code avec un système de gestion de code source tel que git, et par la même occasion vous versionnez également les modifications apportées à votre infrastructure globale. Sans oublier que cela vous aidera à **maintenir la cohérence** entre les procédures de déploiement. De plus, avec l'laC, l'intervention humaine est minimale, ce qui signifie une réduction des erreurs humaines et une uniformité et une standardisation accrues. En conséquence, les problèmes tels que la compatibilité et son fonctionnement sur ma machine sont

réduits au minimum.

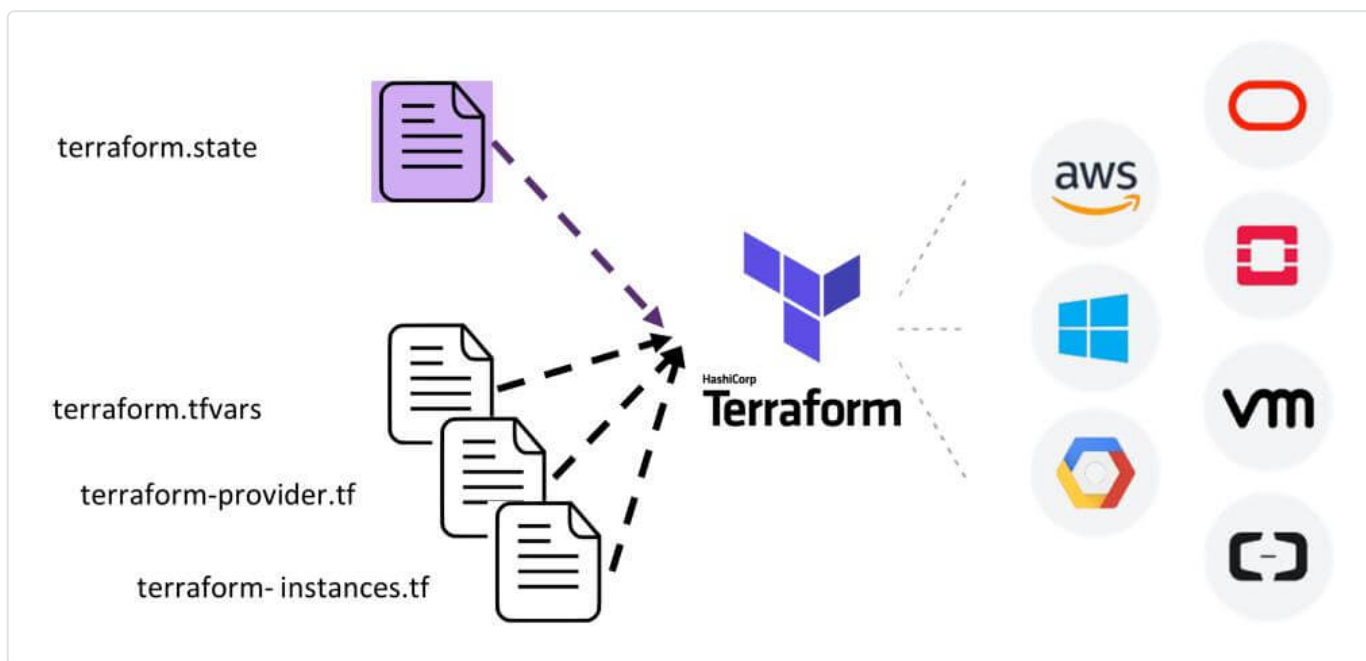
# Terraform

---

## C'est quoi ?

Vous l'aurez compris nous étudierons tout au long de ce cours l'outil Terraform, qui est un outil permettant de créer, modifier et versionner une infrastructure de manière sûre et efficace. Terraform peut **gérer différents fournisseurs d'infrastructure**, allant des Cloud Providers (AWS, Azure, GcP, Alibaba Cloud, etc ...) jusqu'aux solutions internes personnalisées (VmWare, Kubernetes, etc ...).

## Comment ça fonctionne ?



Vous codez vos fichiers de configuration qui décrivent à Terraform les composants nécessaires pour exécuter une seule ressource comme par exemple la création d'une machine virtuelle ou l'ensemble de votre datacenter. Ensuite, il générera plan d'exécution décrivant ce qu'il fera pour atteindre l'état que vous souhaitez, puis

l'exécute pour construire l'infrastructure décrite. Au fur et à mesure que la configuration change, il sera en mesure de déterminer ce qui a changé et de créer des plans d'exécution incrémentiels qui peuvent être appliqués.

L'infrastructure que Terraform peut gérer comprend des composants de bas niveau tel que les instances de calcul, le stockage et la mise en réseau, ainsi que des composants de haut niveau tels que les entrées DNS, les fonctionnalités SaaS, etc. Nous aurons l'occasion d'étudier tous ces aspects dans les futurs chapitres.

## Conclusion

---

Maintenant que vous connaissez les avantages associés à l'Infrastructure as Code, et que je vous ai introduit et présenté l'outil Terraform, vous voudrez peut-être commencer à l'utiliser dans votre organisation ou pour vos projets personnels. C'est pour cela, que sur le prochain chapitre nous commencerons directement par la configuration notre environnement Terraform sur Linux et Windows.