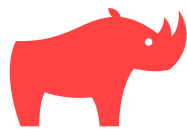


LIVRE BLANC

Les fondamentaux de la conteneurisation et sa gestion sur AWS





Ce livre blanc explore la conteneurisation, une technologie essentielle pour moderniser et simplifier vos applications. Il présente les bases de ce système, les solutions comme Docker et Kubernetes, ainsi que les services d’AWS – ECS, EKS et Lambda – pour gérer efficacement vos conteneurs. Vous y trouverez des conseils pratiques pour choisir la solution adaptée et tirer parti des atouts d’AWS en termes de performance, sécurité et coûts.

Sommaire



- 04** Introduction
- 05** Les fondamentaux de la conteneurisation
- 09** Les solutions de gestion de conteneurs sur AWS : ECS, EKS et Lambda
- 22** Choisir sa solution de conteneurisation
- 27** Conclusion

Introduction

Contexte et enjeux de la conteneurisation

Depuis quelques années, nous sommes passés du développement d'applications lourdes à des applications plus légères. Ces applications sont basées sur un domaine ou une fonction spécifique, ce qui permet de fluidifier les mises à jour, d'utiliser les technologies adéquates et d'appliquer les principes d'élasticité là où cela est nécessaire.

Naturellement, la conteneurisation a gagné en popularité, car elle permet d'encapsuler les applications et leurs dépendances avec la possibilité de porter facilement ces dernières sur différents environnements, par exemple l'on-premise ou le cloud.

Les conteneurs assurent également la cohérence des environnements de développement, de test et de production, réduisant ainsi les problèmes de compatibilité.

Comme toutes technologies, de nouveaux défis se présentent, tels que la gestion des applications à l'échelle, la sécurité, l'automatisation et les compétences nécessaires pour supporter les applications.

La gestion des conteneurs avec AWS

Amazon Web Services (AWS) est l'un des leaders du marché du cloud computing, offrant une multitude de services et de solutions pour la gestion des conteneurs. Le choix de la bonne solution de gestion de conteneurisation sur AWS peut avoir un impact significatif sur la performance, la sécurité, les coûts et l'évolutivité des applications.

Avec plusieurs options disponibles, telles qu'Amazon ECS, EKS et Lambda, il est crucial pour les entreprises de sélectionner une solution qui correspond à leurs besoins spécifiques et à leurs objectifs stratégiques. Une évaluation approfondie des fonctionnalités, des avantages et des inconvénients de chaque option est nécessaire pour s'assurer que l'infrastructure choisie est adaptée à la stratégie de développement et de déploiement de l'entreprise. En fin de compte, une décision éclairée peut non seulement améliorer l'efficacité opérationnelle, mais aussi favoriser l'innovation et la réactivité face aux besoins du marché.

AWS dispose d'autres solutions pouvant exécuter des conteneurs. Toutefois, dans ce livre blanc, nous allons nous concentrer sur les solutions les plus courantes.



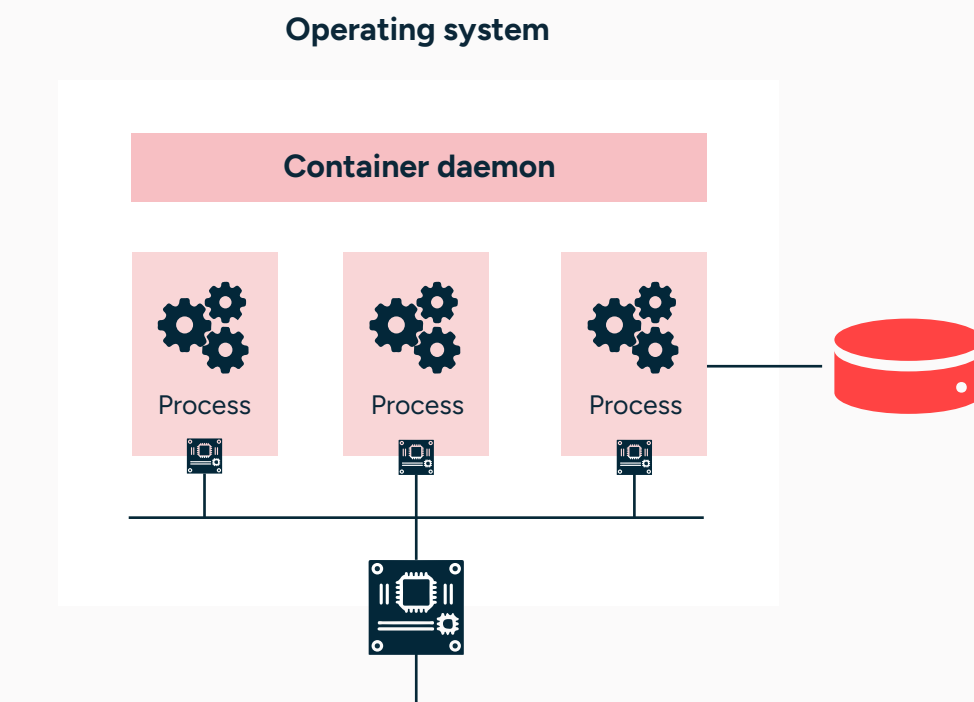
1. Les fondamentaux de la conteneurisation

Qu'est-ce que la conteneurisation ?

La conteneurisation est une méthode d'isolation d'une application ou de processus, avec les dépendances (bibliothèques, exécutables et fichiers) et permet une certaine indépendance avec le système d'exploitation de l'hôte.

Contrairement aux machines virtuelles, qui nécessitent un hyperviseur et incluent un système d'exploitation complet, les conteneurs partagent le même noyau du système d'exploitation hôte, ce qui les rend beaucoup plus légers et rapides à déployer.

Les conteneurs sont généralement gérés par des outils tels que Docker ou containerd, qui facilitent la création, le déploiement et la gestion des conteneurs. Grâce à leur portabilité, les conteneurs peuvent être exécutés sur différentes plateformes, qu'il s'agisse de serveurs physiques, de machines virtuelles ou d'environnements cloud.



Pourquoi utiliser des conteneurs ?

Avantage 1

Portabilité

Les conteneurs peuvent être exécutés sur n'importe quel environnement compatible, ce qui facilite le transfert d'applications entre le développement, le test et la production.

Avantage 2

Scalabilité

Les conteneurs peuvent être rapidement démarrés ou arrêtés, ce qui permet aux entreprises d'ajuster facilement leur infrastructure en fonction de la demande.

Avantage 3

Isolation

Chaque conteneur fonctionne de manière isolée, minimisant ainsi les conflits entre applications et permettant une gestion plus facile des dépendances.

Avantage 4

Efficacité des ressources

Les conteneurs consomment moins de ressources que les machines virtuelles, car ils partagent le noyau de l'hôte, ce qui permet de maximiser l'utilisation des ressources.

Avantage 5

Rapidité de déploiement

Les conteneurs peuvent être créés, déployés et mis à jour rapidement, ce qui accélère le cycle de développement et améliore le time-to-market.

L'écosystème de conteneurs

L'écosystème de conteneurs comprend un ensemble d'outils, de plateformes et de pratiques qui facilitent la gestion et l'orchestration des conteneurs.

Parmi les composants clés de cet écosystème, on trouve :

Docker : L'outil le plus populaire pour créer et gérer des conteneurs. Il fournit des commandes simples pour empaqueter des applications et les exécuter de manière isolée.

Kubernetes : Une plateforme d'orchestration de conteneurs qui automatise le déploiement, la mise à l'échelle et la gestion des applications conteneurisées. Kubernetes facilite la gestion des clusters de conteneurs et permet aux entreprises de gérer des centaines de conteneurs en même temps.

Réseau et stockage : Les solutions de réseau et de stockage adaptées aux conteneurs sont essentielles pour assurer la communication entre les conteneurs et la persistance des données.

CI/CD : L'intégration continue et le déploiement continu (CI/CD) sont des pratiques essentielles dans l'écosystème des conteneurs, permettant des mises à jour fréquentes et fiables des applications. L'écosystème de conteneurs continue d'évoluer, apportant des innovations qui améliorent l'efficacité et la sécurité des applications conteneurisées.

Pourquoi utiliser un orchestrateur de conteneur ?

L'utilisation d'une solution d'orchestration de conteneur permet de faciliter le passage à l'échelle lorsque le nombre d'applications croît et lorsqu'il est nécessaire de gérer la scalabilité d'une application.

Les solutions d'orchestration apportent également des capacités natives ou via des plugins pour faciliter l'intégration avec des services tiers (load balancer, vault, stockage...)



2. Les solutions de gestion de conteneurs sur AWS : EKS, ECS et Lambda

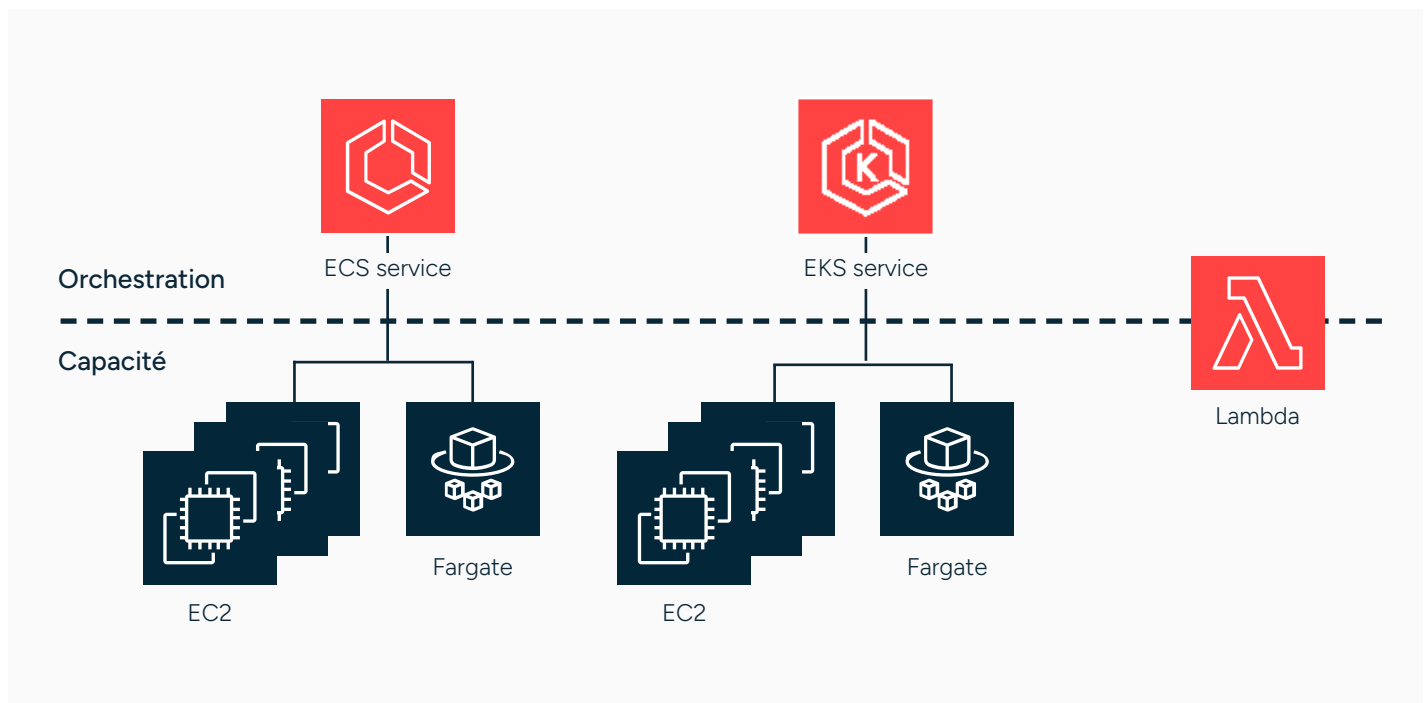
La gestion des conteneurs sur AWS

Les solutions connues lorsqu'on parle de conteneur sur AWS sont ECS et EKS, pour les applications effectuant des traitements longs et Lambda pour du traitement par événement.

ECS et EKS permettent d'orchestrer les charges de travail sur des EC2 ou sur des instances Fargate, alors que pour Lambda les instances sous-jacentes sont entièrement gérées par AWS, seules les ressources nécessaires (RAM, runtime...) sont configurables.

La partie data plane liée à la capacité dans le cadre de ECS et EKS nécessite de définir le type de compute :

- ↻ **EC2** qui est le service de gestion des machines virtuelles déployées avec un système d'exploitation et des agents pour autoriser le management par l'orchestrateur.
- ↻ **Fargate** qui permet d'exécuter des conteneurs sans provisionner ni gérer de serveurs.



ECS, EKS, Lambda :

avantages et inconvénients



Scalabilité dynamique et élastique

L'un des avantages clés de ces orchestrateurs est la capacité à ajuster dynamiquement les ressources en fonction de la charge.

Les services **ECS** et **EKS** permettent de configurer des règles de mise à l'échelle basées sur des métriques telles que l'utilisation du CPU, mémoire ou nombre de requêtes permettant d'adapter sa capacité de traitement et d'optimiser les coûts.

Lambda, de son côté, prend en charge cette gestion de manière complètement transparente, en allouant les ressources en fonction des événements, mais est limité à des exécutions de 15 minutes maximum. Il existe également un quota par compte AWS

qui limite le nombre d'instances lambda exécutées en parallèle qu'il est possible de lever via une demande au support.

Intégration avec l'écosystème AWS

ECS et Lambda s'intègrent naturellement avec un large éventail de services AWS, facilitant la gestion et la surveillance des applications conteneurisées. Parmi ces services :

- 🔗 **CloudWatch** pour la surveillance en temps réel et l'alerte proactive,
- 🔗 **IAM** pour la gestion fine des accès et des autorisations,
- 🔗 **Secrets Manager** pour la gestion sécurisée des informations sensibles,
- 🔗 **Elastic Load Balancing** pour la répartition du trafic réseau.

Dans le cadre d'EKS, il est nécessaire d'utiliser des contrôleurs spécifiques pour étendre la capacité d'intégration, ceci étant dû au fait que EKS repose sur Kubernetes qui est un projet Open Source agnostique à tout cloud provider.

Ces intégrations permettent de construire des infrastructures cloud résilientes et sécurisées, tout en centralisant la gestion des ressources.

Haute disponibilité et multi-région

Les orchestrateurs AWS offrent des fonctionnalités de haute disponibilité en répartissant automatiquement les charges de travail sur plusieurs Zones de Disponibilité (AZ). Ils permettent également la création de clusters ou de services multi-régions, assurant ainsi une tolérance aux pannes à grande échelle.

Sécurité renforcée et conformité

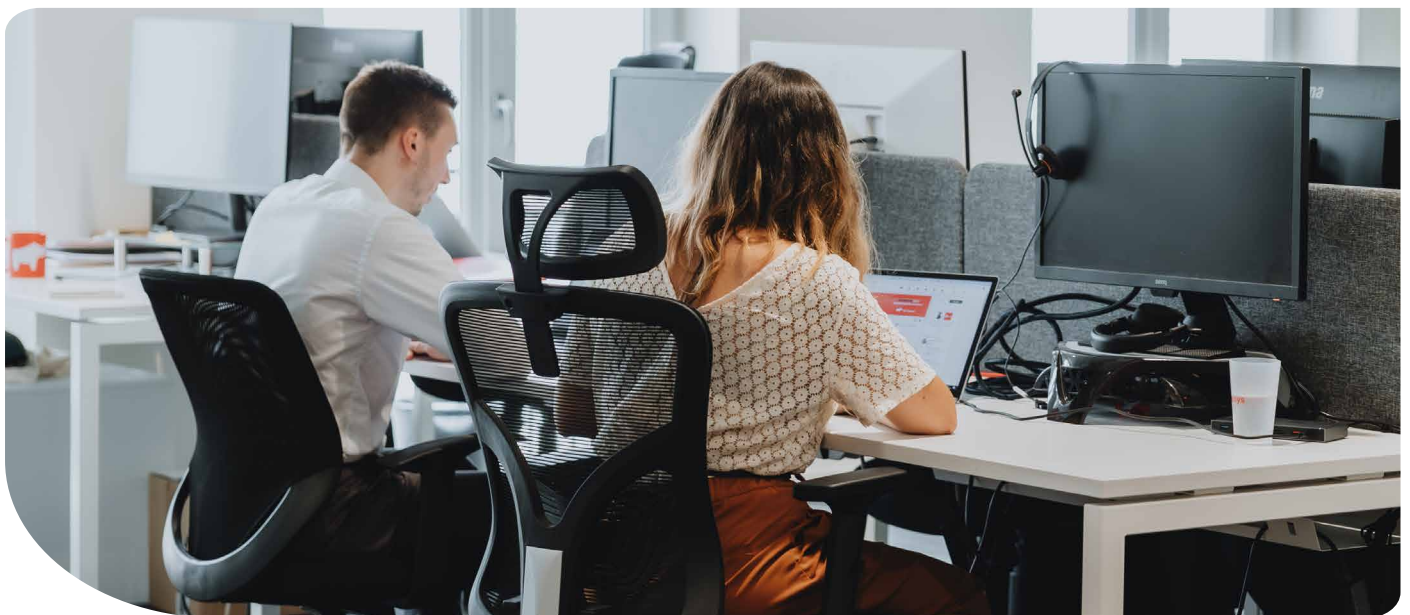
ECS, EKS et Lambda bénéficient des normes de sécurité robustes d’AWS, incluant l’isolation des tâches, le chiffrement des données et l’intégration avec IAM pour une gestion granulaire des permissions.

La conformité avec les normes de sécurité les plus strictes, telles que SOC, ISO et PCI DSS, est native dans ces orchestrateurs, facilitant les audits et garantissant que les applications déployées respectent les exigences réglementaires.

Optimisation des coûts

Un autre avantage non négligeable est l’optimisation des coûts. Avec Fargate, les utilisateurs paient uniquement pour les ressources consommées par les conteneurs actifs, ce qui élimine les coûts de surprovisionnement associés à la gestion de serveurs dédiés.

Pour ECS et EKS, la gestion des instances EC2 peut être optimisée en utilisant des Spot Instances, offrant des économies importantes pour les tâches de calcul élastiques.



Amazon ECS (Elastic Container Service)

Présentation

Amazon ECS (Elastic Container Service) est un service de gestion de conteneurs hautement scalable et performant, conçu pour simplifier le déploiement, la gestion et la mise à l'échelle des applications conteneurisées. ECS permet aux développeurs de lancer des conteneurs Docker dans un environnement entièrement géré, offrant ainsi une orchestration robuste et intégrée au sein de l'écosystème AWS.

ECS prend en charge plusieurs modes d'exécution, notamment EC2 (où les conteneurs s'exécutent sur des instances EC2) et Fargate (où AWS gère

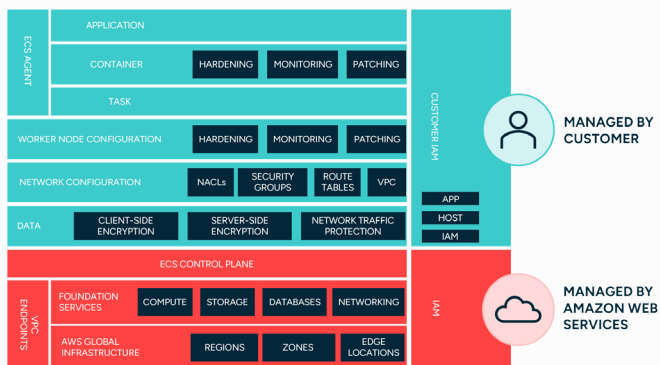
l'infrastructure sous-jacente, permettant aux utilisateurs de se concentrer uniquement sur la gestion des conteneurs). ECS s'intègre également de manière fluide avec d'autres services AWS, tels que Amazon RDS, Amazon S3 et AWS IAM, pour offrir une expérience complète et sécurisée.

Modèle de responsabilité

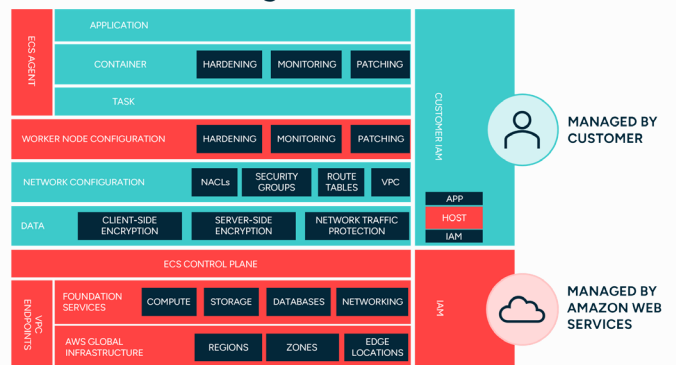
Avec l'utilisation d'EC2 pour héberger les conteneurs, le client est responsable de la couche système d'exploitation et donc du hardening, monitoring et patching.

Avec l'utilisation de Fargate, le client se concentre sur le déploiement, hardening, patching et monitoring de ses applications.

Amazon ECS on EC2



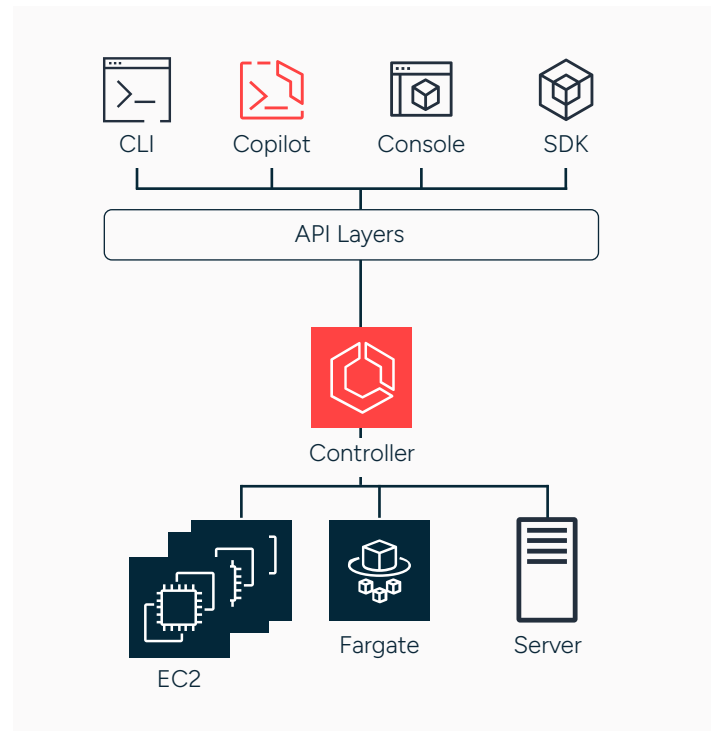
Amazon ECS on Fargate



Architecture

Le service ECS est composé de trois couches :

- 🔄 Les **API** permettant la gestion des clusters et applications
- 🔄 La partie **contrôleur** qui gère le placement des ressources et leur état de fonctionnement
- 🔄 La partie **compute** supportée par les services EC2 et Fargate voire des serveurs on-premises dans le cadre des offres ECS anywhere



Cas d'utilisation

Amazon ECS est particulièrement adapté à divers scénarios d'utilisation, tels que :

Applications web :

Les entreprises peuvent déployer des applications web conteneurisées qui nécessitent une mise à l'échelle rapide en fonction de la demande des utilisateurs.

Microservices :

ECS permet de gérer des architectures basées sur les microservices, facilitant la communication entre les différents services tout en garantissant leur isolation.

Traitement de données :

Les tâches de traitement par lots ou de calcul intensif peuvent être exécutées dans des conteneurs ECS, permettant une gestion efficace des ressources.

CI/CD :

ECS peut être intégré dans des pipelines de CI/CD pour automatiser le déploiement de nouvelles versions d'applications, garantissant ainsi des mises à jour rapides et fiables.

Avantages

Intégration avec AWS :

ECS s'intègre parfaitement avec d'autres services AWS, offrant une solution cohérente et sécurisée pour la gestion des conteneurs.

Simplicité d'utilisation :

Avec son interface utilisateur conviviale et ses API faciles à utiliser, ECS permet une configuration et un déploiement rapides des applications conteneurisées.

Scalabilité automatique :

ECS peut automatiquement ajuster la capacité de votre application en fonction de la demande, optimisant ainsi les coûts et les performances.

Support pour Fargate :

Le mode d'exécution Fargate permet aux utilisateurs de déployer des conteneurs sans avoir à gérer les instances EC2, simplifiant la gestion de l'infrastructure.

Inconvénients

Complexité des configurations avancées :

Bien que simple à utiliser pour des cas d'utilisation basiques, la configuration d'ECS pour des scénarios plus complexes peut nécessiter une compréhension approfondie des différents composants d'AWS.

Dépendance à l'écosystème AWS :

Les entreprises qui choisissent ECS peuvent se retrouver fortement dépendantes de l'écosystème AWS, ce qui peut poser des défis en termes de portabilité et de flexibilité.

Fonctionnalités limitées par rapport à Kubernetes :

ECS, bien qu'efficace, peut ne pas offrir toutes les fonctionnalités avancées disponibles dans Kubernetes, ce qui pourrait limiter certaines utilisations spécifiques.



Amazon EKS (Elastic Kubernetes Service)

Présentation

Amazon EKS (Elastic Kubernetes Service) est un service géré qui facilite le déploiement, la gestion et la mise à l'échelle d'applications conteneurisées utilisant Kubernetes sur AWS. EKS simplifie le processus d'orchestration des conteneurs en prenant en charge la gestion des clusters Kubernetes, y compris les mises à jour, la scalabilité et la disponibilité.

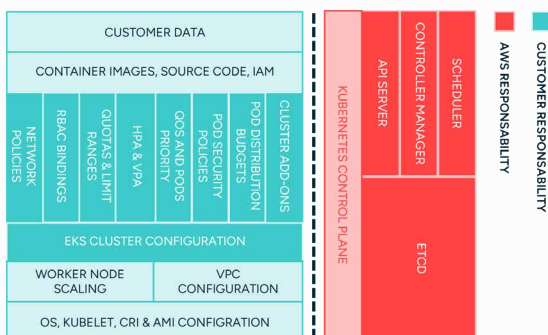
EKS est conçu pour être entièrement compatible avec Kubernetes standard, ce qui permet aux développeurs d'utiliser les mêmes outils et API qu'ils utiliseraient dans un environnement Kubernetes sur site.

Ce service géré permet également d'intégrer facilement d'autres services AWS, offrant ainsi une solution robuste et sécurisée pour les applications conteneurisées.

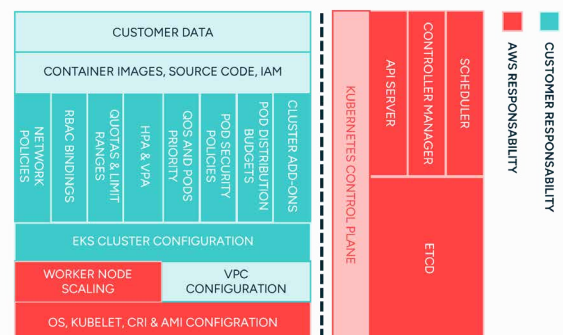
Modèle de responsabilité

Au même titre que ECS, l'utilisation d'EC2 pour les workers nodes augmentent la part de responsabilité du client dans la gestion de l'infrastructure, alors que sur Fargate, le focus est mis sur la gestion applicative.

Self Managed Workers



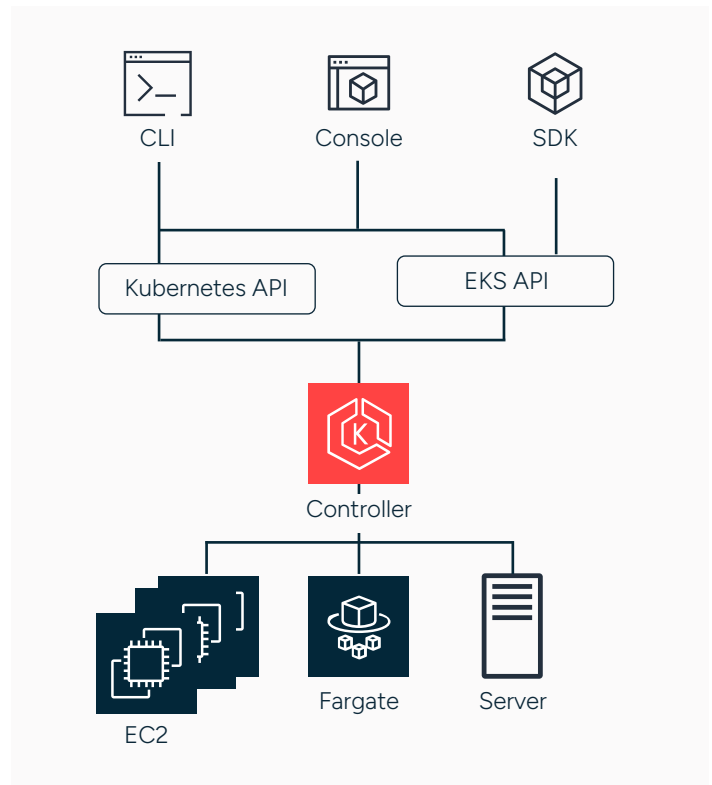
EKS Fargate



Architecture

Le service EKS est composé de trois couches :

- ↪ Les **API de management** de l'infrastructure et celle de Kubernetes.
- ↪ La partie **contrôleur** qui gère le placement des ressources et leur état de fonctionnement.
- ↪ La partie **compute** supportée par les services EC2 et Fargate voir des serveurs on-premises dans le cadre des offres EKS anywhere.



Cas d'utilisation

Amazon EKS est particulièrement adapté pour divers cas d'utilisation, tels que :

Déploiements de micro-services :

Les entreprises peuvent déployer et gérer des architectures basées sur les microservices, en tirant parti des fonctionnalités de Kubernetes pour l'orchestration et la communication inter-services.

Applications nécessitant une haute disponibilité :

EKS offre des options de déploiement multi-AZ (zones de disponibilité) pour assurer une disponibilité continue des applications critiques.

Traitement de données et ML :

EKS est souvent utilisé pour les charges de travail de traitement de données et d'apprentissage automatique, où la scalabilité et l'élasticité sont essentielles.

Environnements hybrides et multi-cloud :

EKS peut être intégré dans des architectures hybrides, permettant aux entreprises d'exécuter des charges de travail sur plusieurs clouds tout en conservant une cohérence dans la gestion des conteneurs.

Avantages

Support complet de Kubernetes :

EKS fournit une gestion complète des clusters Kubernetes, permettant aux utilisateurs d'exploiter les fonctionnalités avancées de Kubernetes tout en bénéficiant d'un service géré.

Scalabilité et flexibilité :

EKS permet de redimensionner les clusters Kubernetes en fonction de la demande, garantissant ainsi que les ressources sont utilisées efficacement.

Sécurité intégrée :

EKS intègre des fonctionnalités de sécurité robustes, telles que le contrôle d'accès basé sur les rôles (RBAC), le chiffrement des données au repos et en transit, ainsi que l'intégration avec AWS IAM.

Écosystème riche :

Les utilisateurs d'EKS peuvent tirer parti de l'écosystème AWS, en intégrant facilement des services comme Amazon RDS, AWS Lambda et Amazon S3 pour enrichir leurs applications.

Inconvénients

Complexité de la gestion de Kubernetes :

Bien qu'EKS gère certaines complexités de Kubernetes, la courbe d'apprentissage pour les utilisateurs qui ne sont pas familiers avec Kubernetes peut être élevée.

Coûts supplémentaires :

EKS entraîne des frais de gestion des clusters, en plus des coûts d'infrastructure associés aux instances EC2 et à d'autres services AWS, ce qui peut augmenter le coût total de possession.

Dépendance à AWS :

Comme pour d'autres services AWS, une forte dépendance à EKS peut rendre la migration vers d'autres environnements plus complexes, ce qui peut poser des défis en termes de portabilité.



Amazon Lambda

Présentation

AWS Lambda, contrairement à ECS et EKS, se concentre sur l'exécution de fonctions serverless en réponse à des événements, sans avoir besoin de gérer l'infrastructure. Cependant, Lambda permet également d'exécuter des conteneurs pour des traitements de courte durée, offrant ainsi une flexibilité similaire aux déploiements conteneurisés tout en limitant les besoins de gestion. En utilisant Lambda avec des conteneurs, les utilisateurs peuvent emballer leurs dépendances et environnements personnalisés dans une image Docker, jusqu'à une taille de 10Go, pour des workloads spécifiques.

Contrairement à ECS et EKS, où les conteneurs sont utilisés pour des applications souvent persistantes, Lambda est particulièrement adapté pour des tâches brèves et scalables déclenchées par des événements, avec une facturation à la milliseconde basée sur le temps d'exécution.

ECS et EKS, notamment avec Fargate, conviennent mieux aux applications à long terme et à des architectures conteneurisées plus complexes, tandis que Lambda est avantageux pour des traitements événementiels nécessitant une grande réactivité sans infrastructure persistante.

Modèle de responsabilité

Deployment mode	Lambda's responsibility	Customer's responsibility
Managed runtime, auto mode	Publish new runtime versions containing the latest patches. Apply runtime patches to existing functions.	Roll back to a previous runtime version in the rare event of a runtime update compatibility issue.
Managed runtime, Function update mode	Publish new runtime versions containing the latest patches.	Update functions regularly to pick up the latest runtime version. Switch a function to Auto mode when you're not regularly updating the function. Roll back to a previous runtime version in the rare event of a runtime update compatibility issue.
Managed runtime, Manual mode	Publish new container images containing the latest patches.	Redeploy functions regularly using the latest container base image to pick up the latest patches.
Container image	Publish new container images containing the latest patches.	Redeploy functions regularly using the latest container base image to pick up the latest patches.

Cas d'utilisation

AWS Lambda est particulièrement adapté à plusieurs cas d'utilisation, notamment :

Applications basées sur des événements :

Lambda est parfait pour exécuter du code en réponse à des événements provenant d'autres services AWS, comme le téléchargement d'un fichier dans S3, une modification de table dans DynamoDB ou un appel d'API via API Gateway.

Traitement de données en temps réel :

Pour des tâches de traitement en continu, comme le filtrage et la transformation de flux de données entrants dans Amazon Kinesis ou la gestion de logs en temps réel, Lambda peut exécuter des fonctions en parallèle et ajuster automatiquement les ressources en fonction du volume.

Automatisation de l'infrastructure :

Lambda permet d'automatiser des tâches d'administration, comme le démarrage ou l'arrêt de ressources, la gestion d'alertes, ou la création de sauvegardes, grâce à des intégrations avec CloudWatch et d'autres services AWS.

Traitements courts en conteneur :

En plus des scripts, Lambda peut exécuter des conteneurs pour des traitements courts, jusqu'à 15 minutes. Cela le rend utile pour des workloads où l'environnement dépend de bibliothèques spécifiques encapsulées dans un conteneur Docker.

Backends sans serveur pour applications mobiles ou web :

Pour des applications nécessitant un backend simple, Lambda peut gérer la logique métier, en étant couplé avec API Gateway et des bases de données comme DynamoDB, sans nécessiter de gestion de serveur ou de conteneur persistants.

Avantages

Gestion ultra-simplifiée :

Lambda ne nécessite aucune gestion d'infrastructure, permettant aux équipes de se concentrer exclusivement sur le développement et la logique métier, avec une gestion automatique de la mise à l'échelle et de la maintenance.

Scalabilité instantanée :

Lambda s'adapte automatiquement à la charge, lançant instantanément des instances pour chaque événement et offrant une mise à l'échelle infinie pour des pics de trafic soudains.

Facturation à la milliseconde :

Les utilisateurs ne paient que pour le temps réel d'exécution, à la milliseconde, ce qui le rend particulièrement économique pour des tâches à exécution courte, comme des traitements déclenchés par des événements.

Simplicité pour des architectures événementielles :

Lambda s'intègre facilement avec des services AWS pour des cas d'utilisation basés sur des événements, comme les notifications, les mises à jour en temps réel, et le traitement de données par lots légers.

Inconvénients

Limitation de durée d'exécution :

Lambda limite la durée d'exécution d'une fonction à 15 minutes, ce qui le rend moins adapté aux traitements de longue durée. Fargate, en revanche, n'a pas cette contrainte de temps et convient mieux aux tâches intensives.

Limites de personnalisation de l'environnement :

Bien que Lambda permette l'exécution de conteneurs, il ne prend en charge que des images Docker jusqu'à 10 Go, avec certaines restrictions sur les bibliothèques et environnements. Fargate offre plus de flexibilité pour des déploiements conteneurisés complexes ou spécifiques.

Facturation à l'exécution par événement :

Pour des workloads continus ou nécessitant des états persistants, Lambda peut devenir coûteux comparé à une facturation horaire plus standard de Fargate pour des tâches longue durée.



3. Choisir sa solution de conteneurisation

Les critères de choix

Le choix d'une solution de conteneurisation appropriée est crucial pour le succès des projets d'application modernes. Plusieurs critères doivent être pris en compte pour garantir que la solution choisie répond aux besoins spécifiques de l'entreprise.

Scalabilité et performance

La scalabilité et la performance sont des facteurs déterminants dans le choix d'une solution de conteneurisation. Une solution doit être capable de s'adapter facilement à la demande, qu'il s'agisse d'une augmentation soudaine du trafic ou d'une charge de travail croissante. Les entreprises doivent évaluer si la solution choisie permet de :

- 🐣 **Scalabilité horizontale** : Ajouter ou supprimer des instances de conteneurs en fonction des besoins.
- 🐣 **Performances optimales** : Assurer que les conteneurs fonctionnent efficacement sans impact sur la latence ou les temps de réponse, même sous charge.
- 🐣 **Équilibrage de charge** : Disposer de mécanismes pour gérer efficacement le trafic entre les conteneurs et garantir une utilisation équilibrée des ressources.



Coûts et budget

Le budget est un élément essentiel dans le choix d'une solution de conteneurisation. Les entreprises doivent tenir compte des coûts associés à la mise en œuvre, à l'exploitation et à la maintenance de la solution. Cela inclut :

- 🐣 **Coûts d'infrastructure** : Évaluer les coûts des ressources cloud, des instances EC2, et des autres composants nécessaires au fonctionnement des conteneurs.
- 🐣 **Modèle de tarification** : Analyser les différentes options de tarification (par exemple, à la demande, réservée ou facturation à l'utilisation) pour choisir celle qui correspond le mieux à leur modèle commercial.
- 🐣 **Coûts opérationnels** : Prendre en compte les coûts liés à la gestion, à la surveillance et à la maintenance des conteneurs.

Facilité d'utilisation et intégration

La sécurité est un aspect primordial dans le choix d'une solution de conteneurisation, en particulier pour les entreprises soumises à des réglementations strictes. Les entreprises doivent évaluer :

- 🐣 **Contrôles de sécurité intégrés** : Vérifier si la solution offre des fonctionnalités de sécurité robustes, telles que le chiffrement des données, l'authentification et les autorisations.
- 🐣 **Conformité réglementaire** : S'assurer que la solution répond aux exigences de conformité pertinentes pour le secteur d'activité, comme le RGPD, PCI-DSS, etc.
- 🐣 **Mécanismes de surveillance et de détection des menaces** : Analyser les capacités de la solution pour surveiller les conteneurs en temps réel et détecter les anomalies ou les menaces potentielles.

Sécurité et conformité

Le support technique et la communauté autour d'une solution de conteneurisation sont des éléments à ne pas négliger. Une bonne assistance peut faire la différence dans la résolution rapide des problèmes. Les critères à considérer incluent :

- 🐣 **Disponibilité du support** : Évaluer les options de support proposées par le fournisseur, y compris les niveaux de service, la réactivité et les heures de disponibilité.
- 🐣 **Documentation et ressources** : Vérifier la qualité de la documentation, des tutoriels et des ressources disponibles pour aider les utilisateurs à se former et à résoudre les problèmes.
- 🐣 **Communauté d'utilisateurs** : Une communauté active peut être une précieuse source d'informations et de solutions, facilitant le partage des meilleures pratiques et des conseils d'utilisation.

Les étapes à suivre pour choisir sa solution

Le choix d'une solution de conteneurisation appropriée nécessite une approche méthodique pour s'assurer qu'elle répond aux besoins de l'entreprise tout en optimisant les ressources et en minimisant les risques. Voici les étapes clés à suivre pour sélectionner la solution la plus adaptée.

1- ÉVALUATION DES BESOINS SPÉCIFIQUES DE L'ENTREPRISE

Avant d'explorer les options de conteneurisation, il est crucial de bien comprendre les besoins et les objectifs spécifiques de l'entreprise. Cette évaluation doit inclure :

- 🔗 **Analyse des processus métier :**
Identifier les applications et services qui bénéficieront le plus de la conteneurisation et comment ils interagissent.

- 🔗 **Détermination des exigences techniques :**
Évaluer les contraintes techniques, telles que la compatibilité avec l'infrastructure existante, les besoins en performances et les exigences de sécurité.

- 🔗 **Prévisions de croissance :**
Anticiper l'évolution des besoins de l'entreprise à court et long terme, en tenant compte des fluctuations de la charge de travail et des exigences de scalabilité.

2- ANALYSE DES SOLUTIONS DISPONIBLES

Une fois les besoins évalués, il est temps d'examiner les différentes solutions de conteneurisation disponibles sur le marché. Cette analyse devrait inclure :

- 🔗 **Recherche des options :**
Dresser un inventaire des solutions de conteneurisation disponibles sur AWS, y compris Amazon ECS, Amazon EKS, AWS Fargate, ROSA, et l'utilisation de Docker sur EC2.

- 🔗 **Comparaison des caractéristiques :**
Évaluer les caractéristiques, les cas d'utilisation, les avantages et inconvénients de chaque solution pour déterminer celles qui répondent le mieux aux besoins identifiés.

- 🔗 **Consultation de la documentation et des avis :**
Consulter la documentation officielle, les études de cas et les retours d'expérience d'autres utilisateurs pour obtenir des insights sur l'efficacité et les limitations des différentes solutions.

3- TEST ET VALIDATION DES SOLUTIONS

Après avoir sélectionné quelques solutions potentielles, il est essentiel de les tester avant de prendre une décision finale.

Les étapes à suivre comprennent :

🐘 **Mise en place d'un environnement de test :**

Créer un environnement de développement ou de test pour déployer et configurer les solutions sélectionnées.

🐘 **Évaluation des performances :**

Tester la performance des solutions en simulant des charges de travail réelles pour observer leur comportement en termes de scalabilité, de latence et de fiabilité.

🐘 **Validation de la sécurité :**

Évaluer les fonctionnalités de sécurité de chaque solution en effectuant des tests de vulnérabilité et en vérifiant la conformité aux exigences réglementaires.

4- MISE EN ŒUVRE ET SUIVI

Une fois la solution de conteneurisation choisie, il est temps de procéder à la mise en œuvre et à la gestion continue de l'environnement. Cette phase comprend :

🐘 **Planification de la mise en œuvre :**

Établir un plan détaillé de mise en œuvre, en définissant les étapes, les ressources nécessaires et les responsabilités des membres de l'équipe.

🐘 **Formation des équipes :**

Assurer que les équipes de développement et d'exploitation reçoivent la formation nécessaire pour utiliser efficacement la nouvelle solution de conteneurisation.

🐘 **Suivi et optimisation :**

Mettre en place des mécanismes de surveillance pour évaluer en continu la performance de la solution et identifier les opportunités d'optimisation et d'amélioration.

Conclusion

La conteneurisation a révolutionné le développement et le déploiement d'applications, permettant aux entreprises d'améliorer leur agilité et leur efficacité opérationnelle. Choisir la bonne solution de gestion de conteneurisation sur AWS est essentiel pour tirer parti de ces avantages tout en répondant aux défis spécifiques à chaque organisation.

Dans ce livre blanc, nous avons abordé plusieurs aspects cruciaux concernant le choix d'une solution de conteneurisation sur AWS :

🔄 Contexte et enjeux :

La conteneurisation est devenue un standard indispensable dans le développement moderne, tout en soulignant l'importance d'une sécurité proactive face aux cybermenaces.

🔄 Compréhension de la conteneurisation :

Nous avons expliqué les concepts de base, les avantages et l'écosystème des conteneurs, ainsi que leur rôle clé dans l'optimisation des processus de développement.

🔄 Solutions de gestion sur AWS :

Une analyse approfondie des principales solutions, notamment Amazon ECS, Amazon EKS, Lambda, Red Hat OpenShift Service on AWS (ROSA), ainsi que l'utilisation de Docker sur EC2 et la création d'un cluster Kubernetes sur EC2.

🔄 Critères de choix :

L'identification des critères essentiels tels que la scalabilité, les coûts, la sécurité, et le support, pour guider les entreprises dans leur processus de sélection.

🔄 Étapes de choix :

Un cadre structuré pour évaluer les besoins, analyser les solutions, tester et valider, et enfin, mettre en œuvre la solution choisie.



Synapsys

A propos de Synapsys

Synapsys est un acteur de référence spécialisé dans la transformation des infrastructures IT. Depuis plus de 12 ans, nous accompagnons nos clients tout au long du cycle de vie des projets d'infrastructure à travers nos expertises en Digital Workplace, Cloud, DevOps, Cybersécurité, Data/IA et Transformation des SI.

Synapsys propose à ses clients un service technologique de qualité, grâce à l'esprit collectif et engagé de ses 180 talents répartis à Paris, Lille, Lyon et Kuala Lumpur.

Nous sommes fiers d'être considérés comme un partenaire de confiance et plébiscités pour la réalisation de projets de transformation structurants. Nos clients grands comptes nous sollicitent pour bâtir des infrastructures agiles et résilientes afin de relever les défis de transformation digitale de demain.

Convaincus que tout projet doit apporter le progrès et toute collaboration, la confiance, nous avons à cœur de proposer une vision de l'entreprise inclusive et équitable. Nous faisons du développement des hommes un véritable modèle d'entreprise qui guide nos orientations stratégiques, notre culture et notre mode de fonctionnement.

Chez Synapsys, c'est la force du collectif, l'engagement, l'équité et l'authenticité qui priment. Nous mettons tout en œuvre pour que chacun ait l'opportunité de se développer professionnellement dans un climat de confiance autour d'un projet commun.

www.synapsys-groupe.com

Crédit images : Synapsys & Adobe Stock

Crédit schémas : Synapsys & AWS