Amazon EKS

Auteur : Dasek Joiakim

LI : 2 sur 2

Ressources :

https://www.udemy.com/course/kubernetes-les-bases-indispensables/learn/lecture/22021750#overview

Introduction

L’orchestration de conteneurs est devenue un pilier essentiel pour les entreprises cherchant à améliorer l'agilité, la scalabilité et la fiabilité de leurs applications. Parmi les différentes solutions d'orchestration de conteneurs, Kubernetes s'est rapidement imposé comme le standard de facto grâce à sa flexibilité et sa robustesse.

Dans cet écosystème en pleine expansion, Amazon Web Services (AWS), un pionnier du cloud computing, a introduit l'Amazon Elastic Container Service (ECS) en 2015. ECS a permis aux clients AWS de déployer et de gérer des conteneurs Docker sur l'infrastructure cloud d'AWS de manière efficace et sécurisée. Cette solution a rencontré un succès considérable, mais les clients ont exprimé un intérêt croissant pour Kubernetes, en raison de sa popularité et de son adoption généralisée dans l'industrie.

En réponse à cette demande croissante, AWS a lancé l'Amazon Elastic Kubernetes Service (EKS) en 2018. EKS offre aux utilisateurs la possibilité de déployer, de gérer et de scaler des applications conteneurisées utilisant Kubernetes sur l'infrastructure cloud d'AWS, tout en bénéficiant de l'évolutivité, de la sécurité et de la fiabilité du cloud AWS.

Le passage d'ECS à EKS pour de nombreux clients AWS s'explique par plusieurs facteurs clés. Tout d'abord, Kubernetes est devenu le standard de facto dans le domaine de l'orchestration de conteneurs, offrant une vaste communauté de développeurs, un écosystème riche en outils et une portabilité des charges de travail entre différents environnements cloud et sur site. Deuxièmement, EKS simplifie considérablement le déploiement et la gestion de clusters Kubernetes, en fournissant une expérience native AWS, une intégration étroite avec d'autres services AWS et une tarification simple et flexible.

L’architecture

AWS EKS repose sur une architecture solide qui combine les capacités d'orchestration de Kubernetes avec l'évolutivité, la fiabilité et la sécurité de l'infrastructure cloud d'AWS. Voici les principaux composants de l'architecture EKS :

- **Cluster EKS :** Un cluster EKS est un groupe de ressources de calcul et de stockage qui exécutent des applications conteneurisées Kubernetes. Il est composé de plusieurs nœuds, dont un nœud maître et plusieurs nœuds de travail.

- **Nœud maître :** AWS EKS gère le nœud maître Kubernetes pour vous. Le nœud maître est responsable de la gestion et de la coordination des nœuds de travail dans le cluster. AWS EKS utilise des services managés tels que Amazon EKS Control Plane et Amazon EKS et AWS Identity and Access Management (IAM) pour gérer le nœud maître, garantissant ainsi une haute disponibilité et une sécurité robuste.

- **Nœuds de travail :** Les nœuds de travail sont des instances EC2 (ou des instances Fargate pour les clusters EKS basés sur Fargate) qui exécutent les conteneurs de votre application. Vous pouvez ajouter ou supprimer des nœuds de travail en fonction des besoins de votre charge de travail, ce qui vous permet de scaler facilement votre infrastructure en fonction de la demande.

Déploiement et gestion des clusters EKS

Déployer et gérer un cluster EKS implique plusieurs étapes :

- **Création du cluster :** Vous pouvez créer un cluster EKS à l'aide de la console AWS Management Console, de l'interface de ligne de commande AWS CLI ou d'une infrastructure en tant que code (IaC) à l'aide d'outils tels que AWS CloudFormation ou AWS CDK.

- **Configuration du cluster :** Une fois le cluster créé, vous pouvez le configurer en définissant des paramètres tels que la taille et le type des instances des nœuds de travail, les politiques de réseau, les stratégies de sécurité, etc.

- **Gestion des nœuds de travail :** Vous pouvez gérer les nœuds de travail de votre cluster EKS à l'aide d'outils de gestion tels que AWS Auto Scaling pour scaler automatiquement les nœuds en fonction de la demande, et AWS Systems Manager pour surveiller et automatiser les tâches administratives.

Intégration avec d'autres services AWS

AWS EKS s'intègre étroitement avec d'autres services AWS pour offrir une expérience d'orchestration Kubernetes native et transparente. Voici quelques exemples d'intégrations :

- **Amazon ECR (Elastic Container Registry) :** Vous pouvez stocker, gérer et déployer des images de conteneurs Docker dans Amazon ECR, puis les utiliser dans vos clusters EKS.

- **AWS IAM (Identity and Access Management) :** Vous pouvez utiliser IAM pour définir des rôles et des politiques d'accès granulaires pour contrôler l'accès aux ressources de votre cluster EKS.

- **AWS CloudWatch :** Vous pouvez surveiller les performances de votre cluster EKS, collecter des métriques et des journaux, et créer des alarmes pour détecter et résoudre les problèmes de manière proactive.

Création d’un compte AWS & bonnes pratiques

La création d'un compte AWS est une étape cruciale pour démarrer votre parcours dans le cloud. Cependant, il est essentiel de mettre en place les bonnes pratiques de sécurité dès le début pour protéger vos données, vos ressources et votre infrastructure contre les menaces potentielles. Voici quelques principes de sécurité à suivre lors de la création de votre compte AWS :

1. Utilisez une adresse e-mail professionnelle :

- Utilisez une adresse e-mail professionnelle pour créer votre compte AWS. Évitez les adresses e-mail publiques ou personnelles pour des raisons de sécurité et de gestion.

2. Activez l'authentification multi-facteurs (MFA) :

- Activez l'authentification multi-facteurs (MFA) pour renforcer la sécurité de votre compte. AWS prend en charge diverses méthodes d'authentification MFA, telles que l'utilisation d'un appareil MFA virtuel ou matériel.

3. Appliquez le principe du moindre privilège :

- Lors de la configuration des utilisateurs et des accès, suivez le principe du moindre privilège. Accordez uniquement les autorisations nécessaires à chaque utilisateur ou rôle pour accomplir leurs tâches spécifiques.

4. Créez des utilisateurs IAM distincts :

- Ne partagez pas les informations d'identification root de votre compte AWS. Au lieu de cela, créez des utilisateurs IAM (Identity and Access Management) distincts pour chaque utilisateur, en leur attribuant des autorisations appropriées.

5. Utilisez des groupes IAM pour gérer les autorisations :

- Utilisez des groupes IAM pour regrouper les utilisateurs ayant des autorisations similaires. Cela simplifie la gestion des autorisations en permettant d'attribuer des autorisations à un groupe plutôt qu'à chaque utilisateur individuellement.

6. Activez les journaux AWS CloudTrail :

- Activez AWS CloudTrail pour enregistrer toutes les activités liées à votre compte AWS. Cela vous permet de suivre les modifications apportées à vos ressources et de répondre aux exigences de conformité en matière d'audit.

7. Configurez des alertes de sécurité :

- Configurez des alertes de sécurité pour surveiller les activités suspectes ou les tentatives d'accès non autorisées à votre compte AWS. Vous pouvez utiliser des services tels que AWS GuardDuty pour détecter les menaces potentielles.

8. Chiffrez vos données sensibles :

- Utilisez le chiffrement pour protéger vos données sensibles lorsqu'elles sont stockées ou en transit. AWS offre des services de chiffrement, tels que AWS Key Management Service (KMS), pour gérer les clés de chiffrement de manière sécurisée.

Pourquoi créer un compte IAM ?

Créer un compte IAM (Identity and Access Management) est une pratique essentielle dans AWS pour plusieurs raisons :

1. **Gestion des Utilisateurs et des Accès :** IAM permet de créer des utilisateurs individuels et de leur attribuer des identifiants d'authentification uniques, ainsi que des autorisations spécifiques pour accéder aux ressources AWS. Cela vous permet de contrôler qui peut accéder à votre compte AWS et quelles actions ils peuvent effectuer.

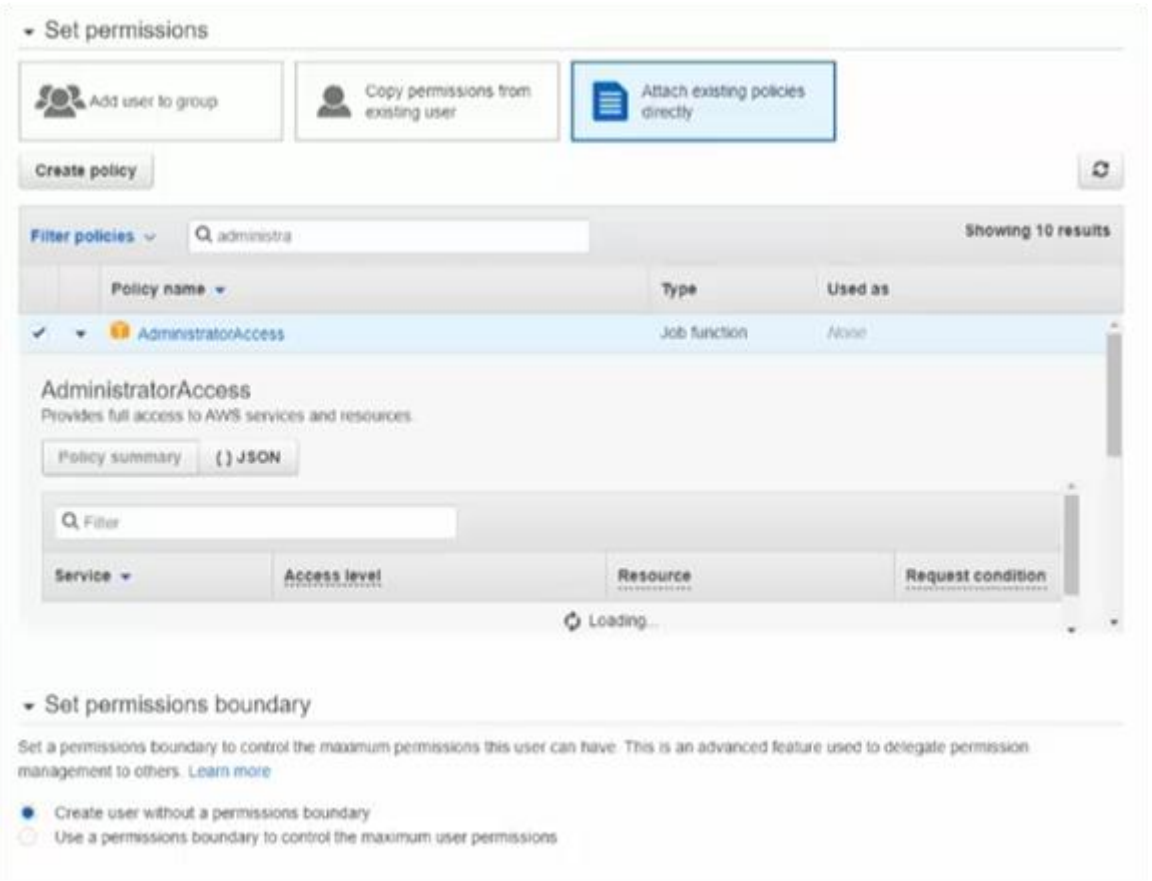
2. **Principe du Moindre Privilège :** IAM vous permet de suivre le principe du moindre privilège en accordant à chaque utilisateur ou rôle uniquement les autorisations nécessaires pour accomplir leurs tâches spécifiques. Cela réduit les risques liés à l'accès excessif ou non autorisé aux ressources AWS.

3. **Sécurité Améliorée :** En utilisant IAM, vous pouvez renforcer la sécurité de votre compte AWS en activant des fonctionnalités telles que l'authentification multi-facteurs (MFA), la rotation régulière des mots de passe, la surveillance des accès et la création de politiques de sécurité personnalisées.

4. **Traçabilité et Audit :** IAM enregistre toutes les actions des utilisateurs, fournissant ainsi une traçabilité complète des activités réalisées dans votre compte AWS. Cela vous permet de répondre aux exigences de conformité et de réaliser des audits de sécurité.

5. **Facilité de Gestion :** En créant des utilisateurs IAM et en les regroupant dans des groupes, vous pouvez simplifier la gestion des autorisations en attribuant des autorisations à un groupe plutôt qu'à chaque utilisateur individuellement. Cela facilite également la gestion des changements d'autorisations lorsque les rôles des utilisateurs évoluent.

Voici la policy à attacher au compte IAM :


Set permissions
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
administra
Showing 10 results
Policy name
Type
Used as
AdministratorAccess
Job function
None
AdministratorAccess
Provides full access to AWS services and resources.
Policy summary
{ } JSON
Filter
Service
Access level
Resource
Request condition
Loading...
Set permissions boundary
Set a permissions boundary to control the maximum permissions this user can have. This is an advanced feature used to delegate permission management to others. Learn more
Create user without a permissions boundary
Use a permissions boundary to control the maximum user permissions

EKSCTL

« **eksctl** » est une interface en ligne de commande open-source conçue pour simplifier le processus de création, de gestion et d'exploitation de clusters Kubernetes sur Amazon EKS (Elastic Kubernetes Service). Il permet aux utilisateurs de créer des clusters EKS en quelques commandes simples, ce qui réduit considérablement le temps et la complexité associés à la configuration manuelle.

Voici les fonctionnalités qu'offre cet outil :

- Create, get, list and delete clusters
- Create, drain and delete nodegroups
- Scale a nodegroup
- Update a cluster
- Use custom AMIs
- Configure VPC Networking
- Configure access to API endpoints
- Support for GPU nodegroups
- Spot instances and mixed instances
- IAM Management and Add-on Policies
- List cluster Cloudformation stacks
- Install coredns
- Write kubeconfig file for a cluster

Installation de eksctl

L'installation de eksctl est assez simple. Sur Windows, nous pouvons utiliser Chocolatey, un gestionnaire de paquets, pour installer eksctl.

Commande d'installation avec Chocolatey :

```
choco install -y eksctl
```

Configuration de eksctl

Après l'installation, vous pouvez configurer eksctl avec vos informations AWS. Assurez-vous que vos clés d'accès AWS sont correctement configurées sur votre système.

Commande pour la configuration initiale :

```
eksctl configure
```

Suivez les instructions pour fournir les informations nécessaires, comme la région AWS, les clés d'accès IAM, etc.

Déploiement du cluster

Maintenant que eksctl est configuré, vous pouvez déployer votre cluster EKS avec une seule commande.

Commande de déploiement du cluster :

```
eksctl create cluster --name <nom-du-cluster> --version <version-de-kubernetes> --node-type <type-de-noeud> --nodes <nombre-de-noeuds>
```

Remplacez les valeurs entre < > avec les spécifications de votre cluster.

Vérification du déploiement

Une fois le cluster déployé, vous pouvez vérifier son état pour vous assurer que tout s'est bien passé.

Commande de vérification du cluster :

```
eksctl get cluster
```

Cela affichera les détails de votre cluster EKS.

Gestion du cluster

Vous pouvez gérer votre cluster EKS à l'aide de eksctl pour effectuer diverses opérations telles que l'ajout ou la suppression de nœuds.

Commandes de gestion du cluster :

- Ajout de nœuds :

```
eksctl scale nodegroup --cluster <nom-du-cluster> --nodes <nombre-de-noeuds> --name <nom-du-nœud>
```

- Suppression de nœuds :

```
eksctl delete nodegroup --cluster <nom-du-cluster> --name <nom-du-nœud>
```

Intégration avec vos fichiers de configuration Kubernetes

Maintenant que votre cluster est opérationnel, vous devez configurer **kubectl** pour qu'il puisse communiquer avec le cluster EKS.

Commande pour configurer kubectl :

```
aws eks --region <votre-région> update-kubeconfig --name <nom-du-cluster>
```

Assurez-vous que vous avez le CLI AWS installé et configuré avec vos informations d'identification.

Après avoir exécuté cette commande, kubectl sera configuré pour se connecter à votre cluster EKS. On peut donc utiliser les commandes de kubectl pour interagir directement avec EKS.

Annexes :

Je vous fournis un lien vers mon repo qui contient un cluster simple avec trois namespaces dev, staging et production. Vous pouvez par la suite :

Appliquez vos fichiers de configuration sur le cluster EKS

Il vous suffit maintenant de lancer les commandes dans le dossier k8s du projet : « init.ps1 »