aws

Bonnes pratiques pour

# Amazon Elastic Container Service

# Amazon Elastic Container Service: Bonnes pratiques pour

# Table of Contents

# Introduction

Amazon Elastic Container Service (Amazon ECS) est un service de gestion de conteneurs hautement évolutif et rapide, qui permet d'exécuter, d'arrêter et de gérer facilement des conteneurs sur un cluster. Ce guide couvre un grand nombre des meilleures pratiques opérationnelles les plus importantes tout en expliquant les principaux sujets qui sous-tendent le fonctionnement des applications basées sur Amazon ECS. L'objectif est de fournir une approche concrète et exploitable pour l'exploitation et le dépannage des applications basées sur Amazon ECS.

Ce guide sera révisé régulièrement afin d'intégrer les nouvelles meilleures pratiques Amazon ECS. Si vous avez des questions ou des commentaires sur l'un des contenus de ce guide, soulevez un problème dans le référentiel GitHub. Pour de plus amples informations, veuillez consulterGuide des meilleures pratiques Amazon ECSsur GitHub.

- Bonnes pratiques - Mise en réseau
- Meilleures pratiques - Mise à l'échelle automatique et gestion de la capacité
- Meilleures pratiques - Stockage persistant
- Bonnes pratiques de sécurité

# Bonnes pratiques - Mise en réseau

Les applications modernes sont généralement construites à partir de plusieurs composants distribués qui communiquent entre eux. Par exemple, une application mobile ou Web peut communiquer avec un point de terminaison API, et l'API peut être alimentée par plusieurs microservices qui communiquent sur Internet.

Ce guide présente les meilleures pratiques pour la création d'un réseau où les composants de votre application peuvent communiquer entre eux de manière sécurisée et évolutive.

Rubriques



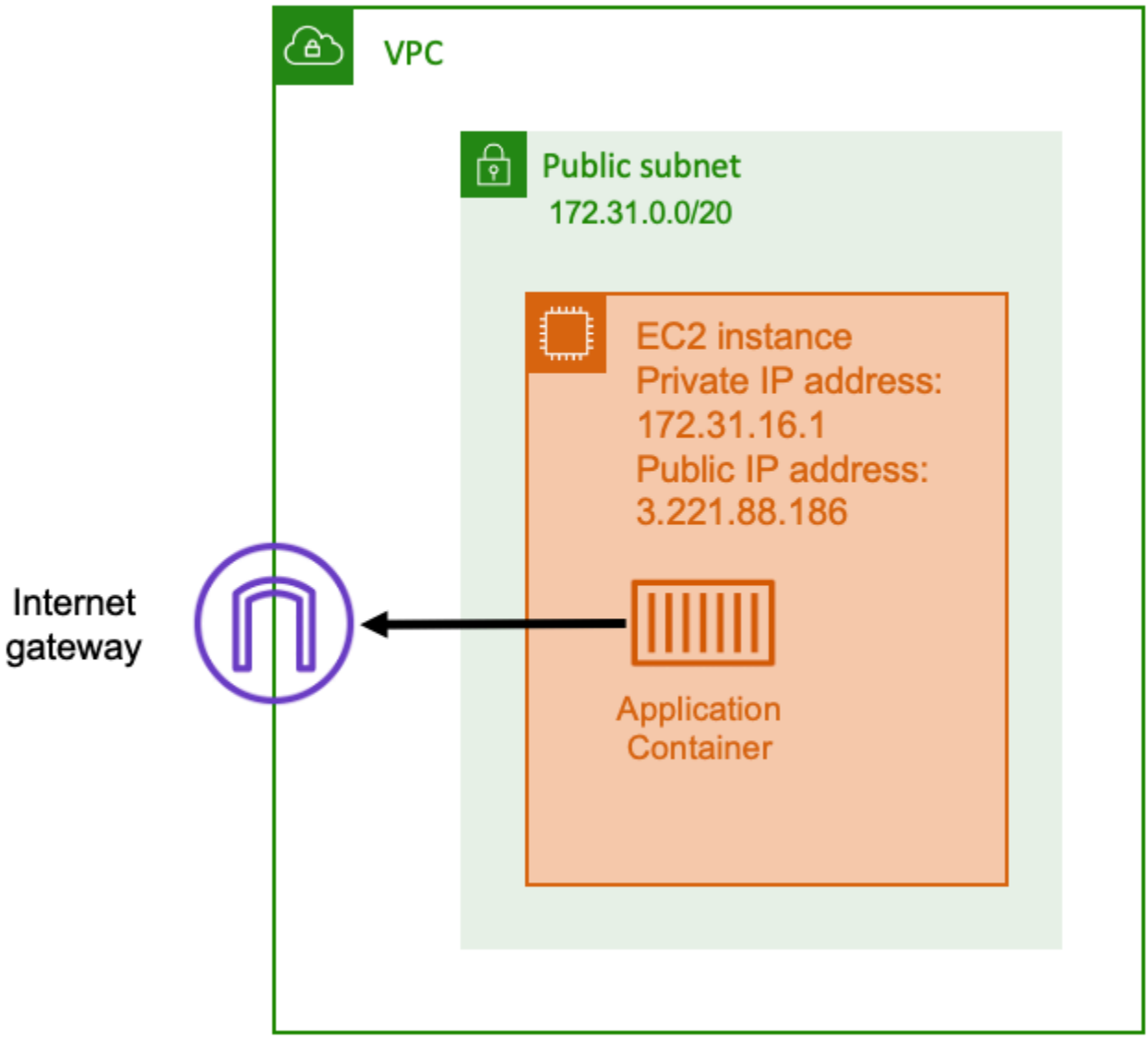
## Connexion Internet

La plupart des applications conteneurisées ont au moins certains composants qui ont besoin d'un accès sortant à Internet. Par exemple, le backend d'une application mobile nécessite un accès sortant pour les notifications push.

Amazon Virtual Private Cloud propose deux méthodes principales pour faciliter la communication entre votre VPC et Internet.

Rubriques

# Utilisation d'un sous-réseau public et d'une passerelle Internet




En utilisant un sous-réseau public comportant une route vers une passerelle Internet, votre application conteneurisée peut s'exécuter sur un hôte d'un VPC sur un sous-réseau public. Une adresse IP publique est attribuée à l'hôte qui exécute votre conteneur. Cette adresse IP publique est routable à partir d'Internet. Pour de plus amples informations, veuillez consulterPasserelles Internetdans leManuel de l'utilisateur Amazon VPC.

Cette architecture réseau facilite la communication directe entre l'hôte qui exécute votre application et d'autres hôtes sur Internet. La communication est bidirectionnelle. Cela signifie que non seulement vous pouvez établir une connexion sortante à n'importe quel autre hôte sur Internet, mais que

d'autres hôtes sur Internet peuvent également tenter de se connecter à votre hôte. Par conséquent, vous devez prêter une attention particulière à vos règles de groupe de sécurité et de pare-feu. Ceci permet de s'assurer que les autres hôtes sur Internet ne peuvent pas ouvrir de connexions que vous ne souhaitez pas ouvrir.

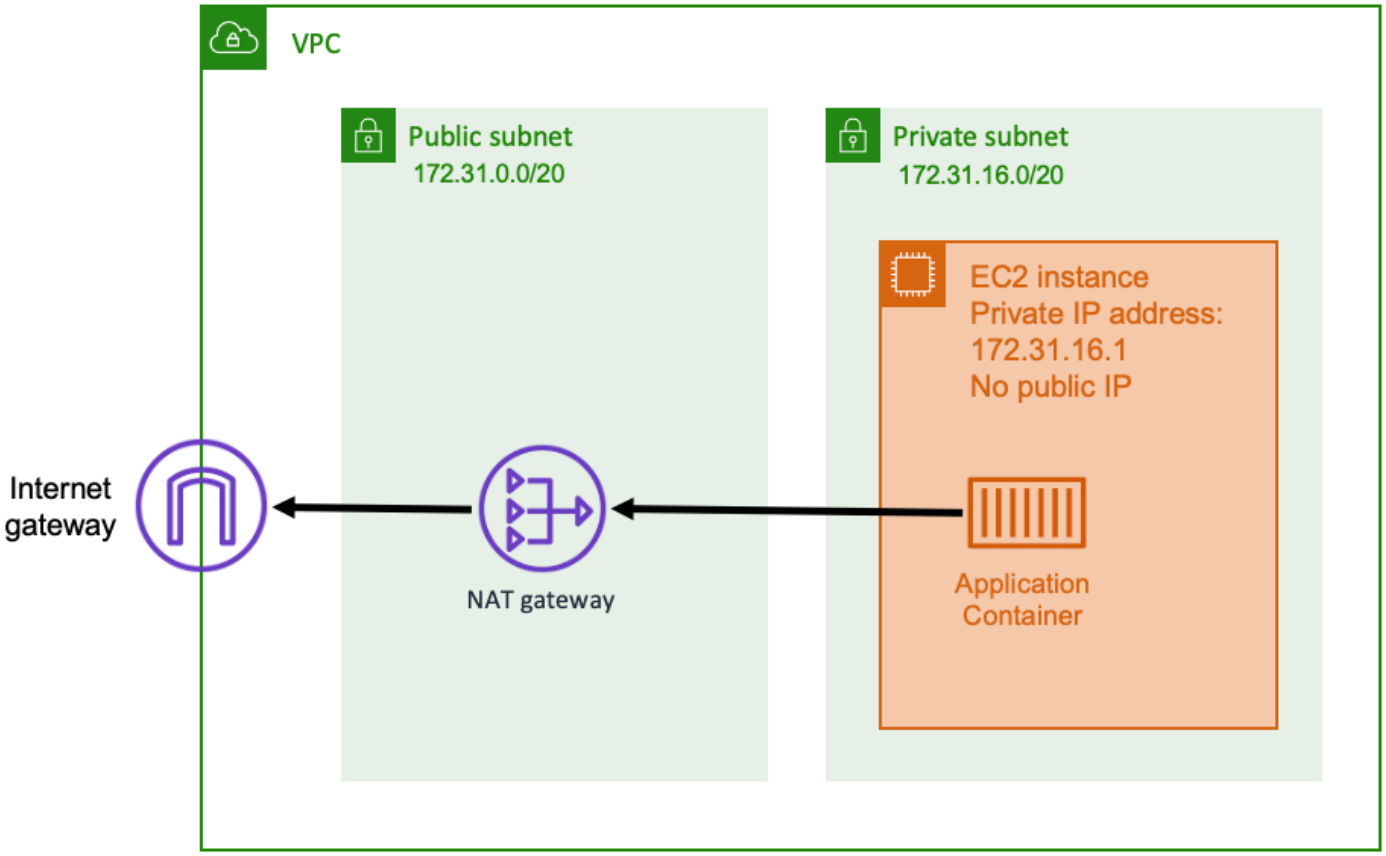
Par exemple, si votre application est exécutée sur Amazon EC2, assurez-vous que le port 22 pour l'accès SSH n'est pas ouvert. Sinon, votre instance pourrait recevoir des tentatives de connexion SSH constantes de robots macilieux sur Internet. Ces robots font le chalut à travers les adresses IP publiques. Après avoir trouvé un port SSH ouvert, ils tentent de forcer les mots de passe pour essayer d'accéder à votre instance. Pour cette raison, de nombreuses organisations limitent l'utilisation des sous-réseaux publics et préfèrent avoir la plupart, sinon la totalité, de leurs ressources à l'intérieur de sous-réseaux privés.

L'utilisation de sous-réseaux publics pour la mise en réseau convient aux applications publiques qui nécessitent de grandes quantités de bande passante ou une latence minimale. Les cas d'utilisation applicables comprennent le streaming vidéo et les services de jeux vidéo.

Cette approche de mise en réseau est prise en charge à la fois lorsque vous utilisez Amazon EC2 et lorsque vous l'utilisez surAWS Fargate.

- Utilisation d'Amazon EC2 : vous pouvez lancer des instances EC2 sur un sous-réseau public. Amazon ECS utilise ces instances EC2 comme capacité de cluster, et tous les conteneurs qui s'exécutent sur les instances peuvent utiliser l'adresse IP publique sous-jacente de l'hôte pour la mise en réseau sortante. Cela s'applique à la fois à la`host`and`bridge`modes réseau. Cependant, le`awsvpc`Le mode réseau ne fournit pas d'ENI de tâche avec des adresses IP publiques. Par conséquent, ils ne peuvent pas utiliser directement une passerelle Internet.
- Utilisation de Fargate : lorsque vous créez votre service Amazon ECS, spécifiez des sous-réseaux publics pour la configuration réseau de votre service, et assurez-vous que leAttribuez une adresse IP publiqueest activée. Chaque tâche Fargate est mise en réseau dans le sous-réseau public et possède sa propre adresse IP publique pour la communication directe avec Internet.

# Utilisation d'un sous-réseau privé et d'une passerelle NAT




En utilisant un sous-réseau privé et une passerelle NAT, vous pouvez exécuter votre application conteneurisée sur un hôte situé dans un sous-réseau privé. En tant que tel, cet hôte a une adresse IP privée qui est routable à l'intérieur de votre VPC, mais qui n'est pas routable à partir d'Internet. Cela signifie que d'autres hôtes à l'intérieur du VPC peuvent établir des connexions à l'hôte en utilisant son adresse IP privée, mais les autres hôtes sur Internet ne peuvent pas effectuer de communications entrantes avec l'hôte.

Avec un sous-réseau privé, vous pouvez utiliser une passerelle de traduction d'adresses réseau (NAT) pour autoriser un hôte d'un sous-réseau privé à se connecter à Internet. Les hôtes sur Internet reçoivent une connexion entrante qui semble provenir de l'adresse IP publique de la passerelle NAT située à l'intérieur d'un sous-réseau public. La passerelle NAT est chargée de servir de pont entre Internet et le VPC privé. Cette configuration est souvent préférée pour des raisons de sécurité, car cela signifie que votre VPC est protégé contre l'accès direct des attaquants sur Internet. Pour de plus amples informations, veuillez consulterPasserelles NATdans leManuel de l'utilisateur Amazon VPC.

Cette approche de mise en réseau privée convient aux scénarios dans lesquels vous souhaitez protéger vos conteneurs contre un accès externe direct. Les scénarios applicables comprennent les systèmes de traitement des paiements ou les conteneurs qui stockent les données utilisateur et les mots de passe. Vous êtes facturé pour la création et l'utilisation d'une passerelle NAT dans votre compte. Des tarifs s'appliquent également aux tarifs horaires et au traitement de données d'une passerelle NAT. À des fins de redondance, vous devez disposer d'une passerelle NAT dans chaque zone de disponibilité. De cette façon, la perte de disponibilité d'une seule zone de disponibilité ne compromet pas votre connectivité sortante. Pour cette raison, si vous avez une charge de travail réduite, il peut être plus rentable d'utiliser des sous-réseaux privés et des passerelles NAT.

Cette approche de mise en réseau est prise en charge à la fois lors de l'utilisation d'Amazon EC2 et lors de l'utilisation deAWS Fargate.

- Utilisation d'Amazon EC2 : vous pouvez lancer des instances EC2 sur un sous-réseau privé. Les conteneurs qui s'exécutent sur ces hôtes EC2 utilisent la mise en réseau des hôtes sous-jacents, et les demandes sortantes passent par la passerelle NAT.
- Utilisation de Fargate : lorsque vous créez votre service Amazon ECS, spécifiez des sous-réseaux privés pour la configuration réseau de votre service et n'activez pas l'optionAttribuez une adresse IP publiqueOption. Chaque tâche Fargate est hébergée dans un sous-réseau privé. Son trafic sortant est acheminé via n'importe quelle passerelle NAT que vous avez associée à ce sous-réseau privé.

## Réception de connexions entrantes depuis Internet

Si vous exécutez un service public, vous devez accepter le trafic entrant provenant d'Internet. Par exemple, votre site Web public doit accepter les requêtes HTTP entrantes provenant des navigateurs. Dans ce cas, d'autres hôtes sur Internet doivent également initier une connexion entrante à l'hôte de votre application.

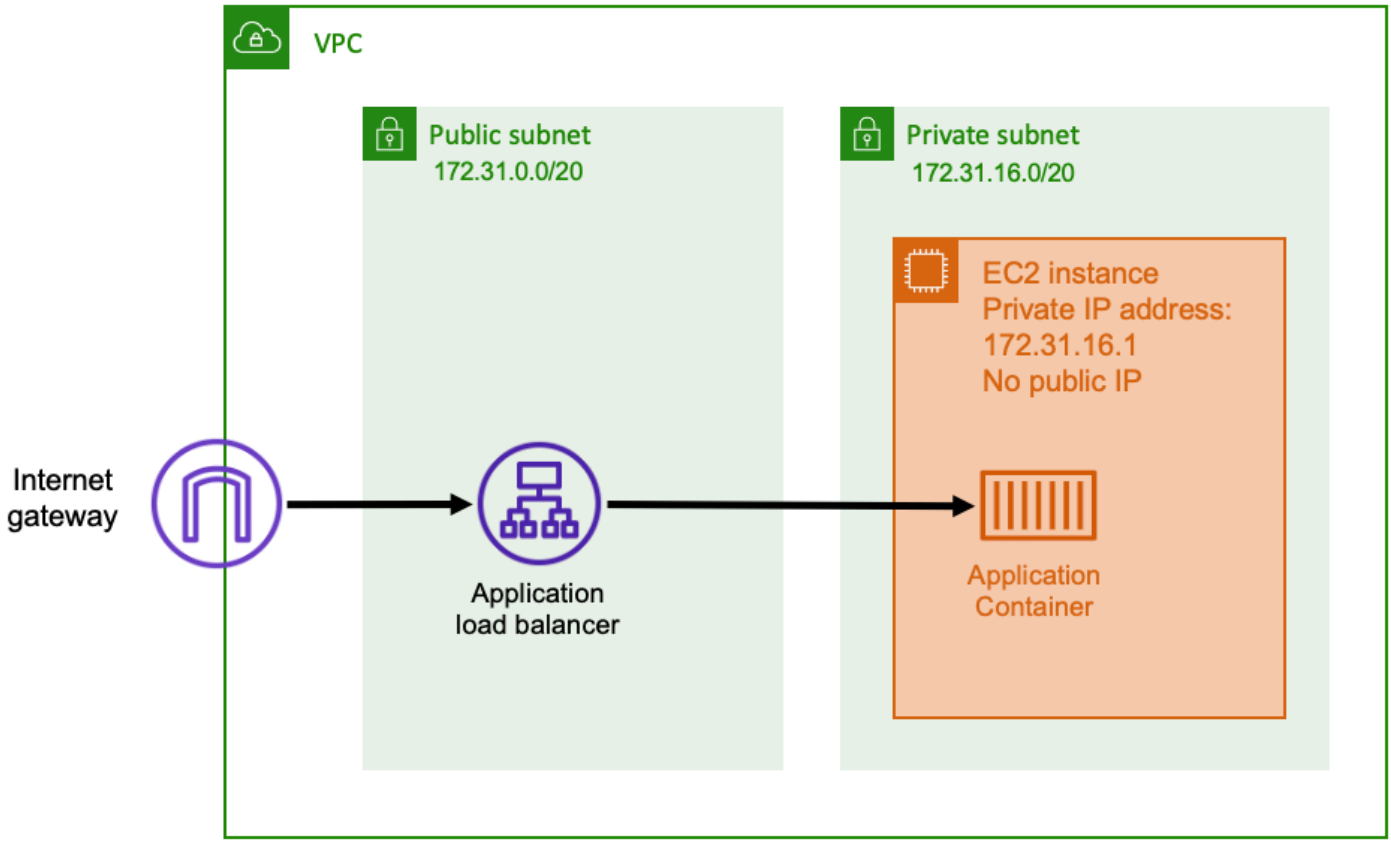
Une approche à ce problème consiste à lancer vos conteneurs sur des hôtes qui se trouvent dans un sous-réseau public avec une adresse IP publique. Cependant, nous ne recommandons pas cette méthode pour les applications à grande échelle. Pour ceux-ci, une meilleure approche consiste à avoir une couche d'entrée évolutive qui se trouve entre Internet et votre application. Pour cette approche, vous pouvez utiliser n'importe quel desAWSrépertoriés dans cette section en tant qu'entrée.

Rubriques

- Equilibreur de charge d'application
- Équilibreur de charge du réseau
- Amazon API Gateway

## Equilibreur de charge d'application

Un Application Load Balancer fonctionne au niveau de la couche d'application. Il s'agit de la septième couche du modèle OSI Open Systems Interconnection (OSI). Cela rend un Application Load Balancer adapté aux services HTTP publics. Si vous disposez d'un site Web ou d'une API REST HTTP, un équilibreur de charge d'application est un équilibreur de charge approprié pour cette charge de travail. Pour de plus amples informations, veuillez consulterQu'est-ce qu'un Application Load Balancer ?dans leGuide de l'utilisateur des équilibreurs de charge d'application.




Avec cette architecture, vous créez un Application Load Balancer dans un sous-réseau public afin qu'il dispose d'une adresse IP publique et puisse recevoir des connexions entrantes à partir d'Internet. Lorsque l'Application Load Balancer reçoit une connexion entrante, ou plus précisément

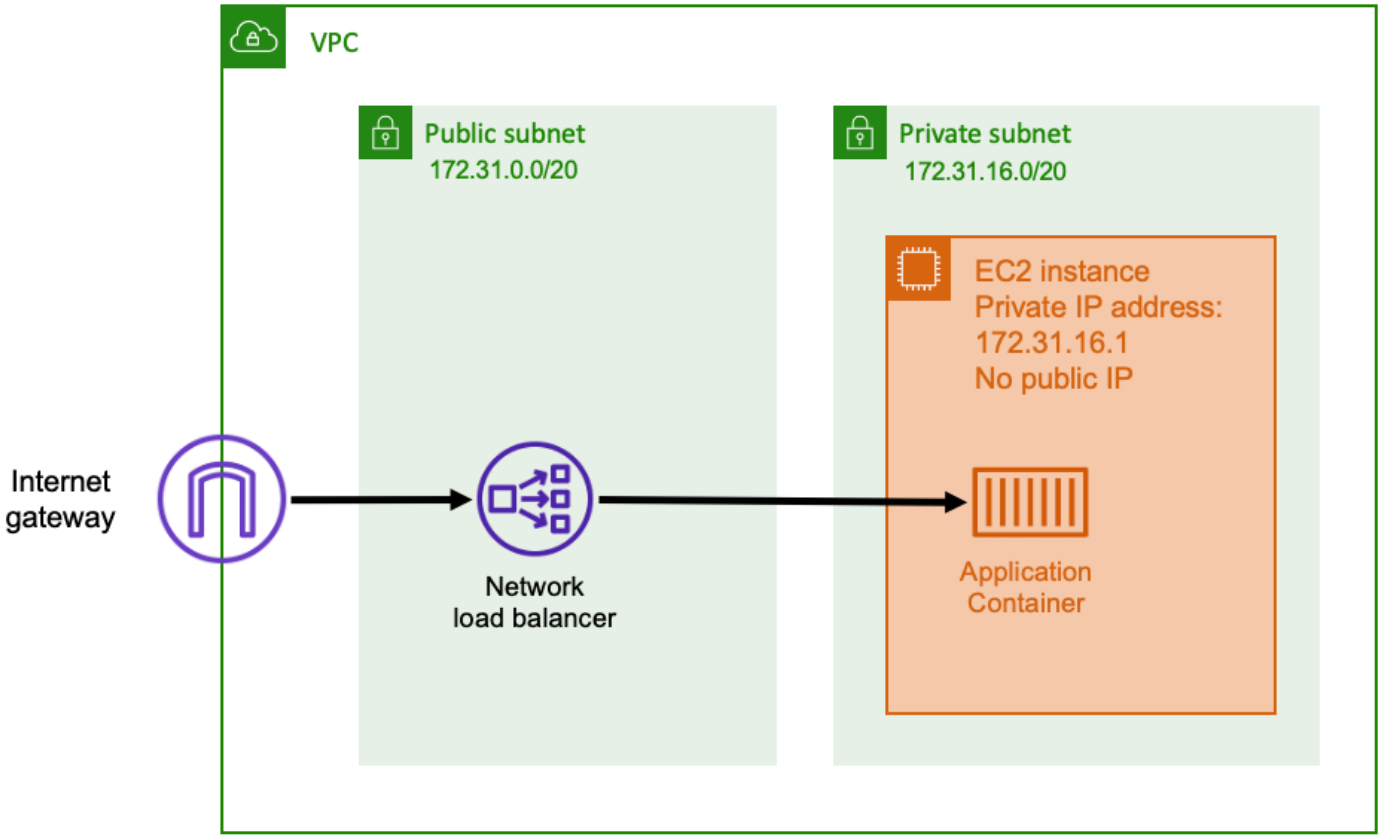
une requête HTTP, il ouvre une connexion à l'application à l'aide de son adresse IP privée. Ensuite, il transfère la demande sur la connexion interne.

L'équilibreur de charge Application Load Balancer offre les avantages suivants.

- Terminaison SSL/TLS : un équilibreur de charge d'application peut supporter des communications HTTPS sécurisées et des certificats pour les communications avec les clients. Il peut éventuellement mettre fin à la connexion SSL au niveau de l'équilibreur de charge afin que vous n'ayez pas à gérer les certificats dans votre propre application.
- Routage avancé : un Application Load Balancer peut avoir plusieurs noms d'hôte DNS. Il dispose également de fonctionnalités de routage avancées pour envoyer des requêtes HTTP entrantes vers différentes destinations en fonction de mesures telles que le nom d'hôte ou le chemin d'accès de la requête. Cela signifie que vous pouvez utiliser un seul Application Load Balancer comme entrée pour de nombreux services internes différents, ou même des microservices sur différents chemins d'accès d'une API REST.
- Prise en charge de gRPC et websockets — Un équilibreur de charge d'application peut gérer plus qu'un simple HTTP. Il peut également équilibrer la charge des services basés sur gRPC et websocket, avec prise en charge HTTP/2.
- Sécurité : un Application Load Balancer aide à protéger votre application contre le trafic malveillant. Il comprend des fonctionnalités telles que les atténuations HTTP de synchronisation et est intégré à AWS Web Application Firewall (AWS WAF).AWS WAFpeut également filtrer le trafic malveillant susceptible de contenir des modèles d'attaque, tels que l'injection SQL ou l'écriture de scripts intersites.

## Équilibreur de charge du réseau

Un Network Load Balancer fonctionne à la quatrième couche du modèle OSI Open Systems Interconnection (OSI). Il convient aux protocoles non HTTP ou aux scénarios où le chiffrement de bout en bout est nécessaire, mais ne possède pas les mêmes fonctionnalités spécifiques à HTTP qu'un équilibreur de charge d'application. Par conséquent, un Network Load Balancer est le mieux adapté aux applications qui n'utilisent pas HTTP. Pour de plus amples informations, veuillez consulterQu'est-ce qu'un Network Load Balancer ?dans leGuide de l'utilisateur des Network Load Balancers.




Lorsqu'un Network Load Balancer est utilisé comme entrée, il fonctionne de la même manière qu'un équilibreur de charge Application Load Balancer. C'est parce qu'il est créé dans un sous-réseau public et possède une adresse IP publique à laquelle il est possible d'accéder sur Internet. L'Network Load Balancer ouvre alors une connexion à l'adresse IP privée de l'hôte exécutant votre conteneur et envoie les paquets du côté public vers le côté privé.

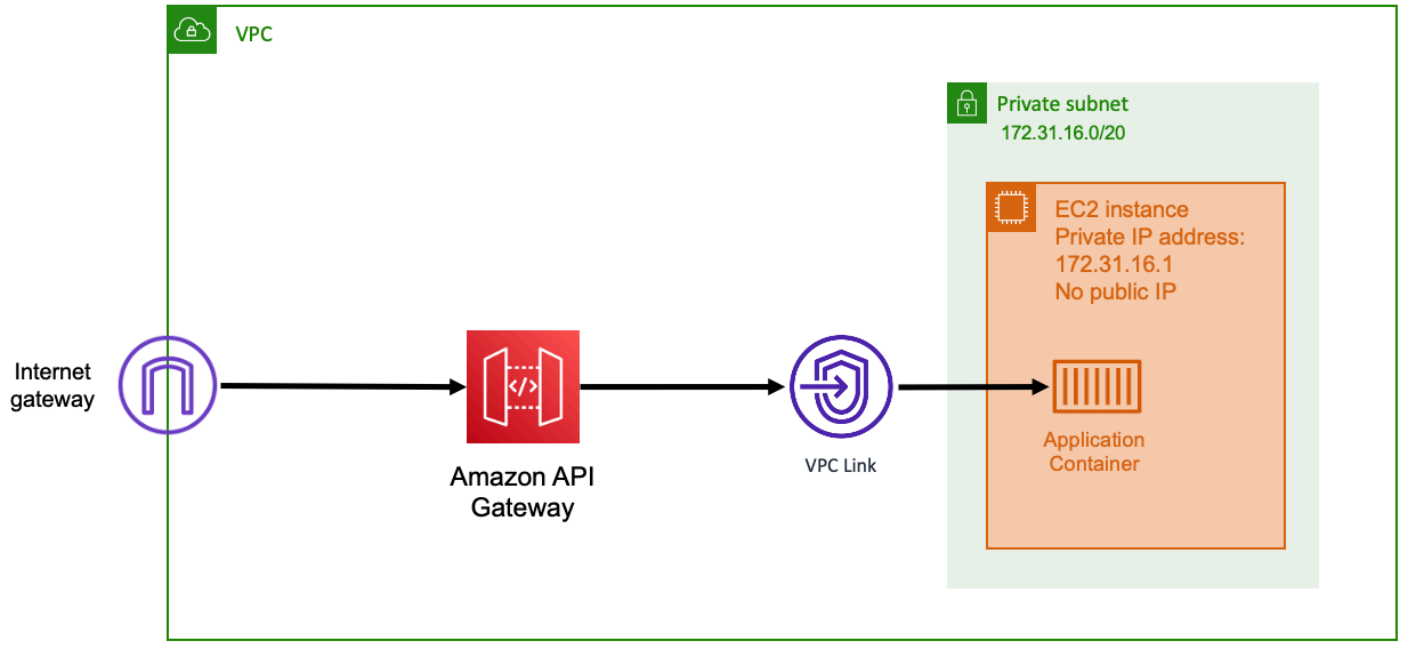
Étant donné que l'Network Load Balancer fonctionne à un niveau inférieur de la pile réseau, il ne dispose pas du même ensemble de fonctionnalités que l'Application Load Balancer. Cependant, il a les caractéristiques importantes suivantes.

- Cryptage de bout en bout — Étant donné qu'un équilibreur de charge réseau fonctionne à la quatrième couche du modèle OSI, il ne lit pas le contenu des paquets. Il convient ainsi aux communications d'équilibrage de charge qui ont besoin d'un chiffrement de bout en bout.
- Chiffrement TLS : en plus du chiffrement de bout en bout, l'Network Load Balancer peut également mettre fin aux connexions TLS. De cette façon, vos applications back-end n'ont pas besoin d'implémenter leur propre TLS.

- Prise en charge UDP — Étant donné qu'un Network Load Balancer fonctionne à la quatrième couche du modèle OSI, il convient aux charges de travail et protocoles autres que TCP non HTTP.

## Amazon API Gateway

L'API HTTP d'Amazon API Gateway est une entrée moins serveur qui convient aux applications HTTP avec des rafales soudaines dans les volumes de demandes ou des volumes de demandes faibles. Pour de plus amples informations, veuillez consulterQu'est-ce qu'Amazon API Gateway ?dans leManuel du développement API Gateway.




Le modèle de tarification de l'Application Load Balancer et de l'Network Load Balancer inclut un prix horaire pour maintenir les équilibreurs de charge disponibles pour accepter les connexions entrantes à tout moment. En revanche, API Gateway facture séparément pour chaque demande. Cela a pour effet que, si aucune demande n'est présentée, il n'y a pas de frais. Sous des charges de trafic élevées, un équilibreur de charge d'application ou un Network Load Balancer peut traiter un plus grand volume de demandes à un prix par demande moins cher que la API Gateway. Toutefois, si le nombre total de demandes est faible ou si vous avez des périodes de trafic faible, le prix cumulé pour l'utilisation de la API Gateway devrait être plus rentable que de payer des frais horaires pour maintenir un équilibreur de charge sous-utilisé.

API Gateway fonctionne à l'aide d'un lien VPC qui permet auxAWSpour se connecter à des hôtes à l'intérieur du sous-réseau privé de votre VPC, à l'aide de son adresse IP privée. Il peut détecter ces

adresses IP privées en regardantAWS Cloud Mapenregistrements de découverte de service gérés par la découverte de service Amazon ECS.

API Gateway prend en charge les fonctions ci-dessous.

- Terminaison SSL/TLS
- Routage de différents chemins HTTP vers différents microservices back-end

Outre les fonctionnalités précédentes, API Gateway prend également en charge l'utilisation d'autorisations Lambda personnalisées que vous pouvez utiliser pour protéger votre API contre une utilisation non autorisée. Pour de plus amples informations, veuillez consulterRemarques sur le terrain : API basées sur un conteneur sans serveur avec Amazon ECS et Amazon API Gateway.

# Choix d'un mode réseau

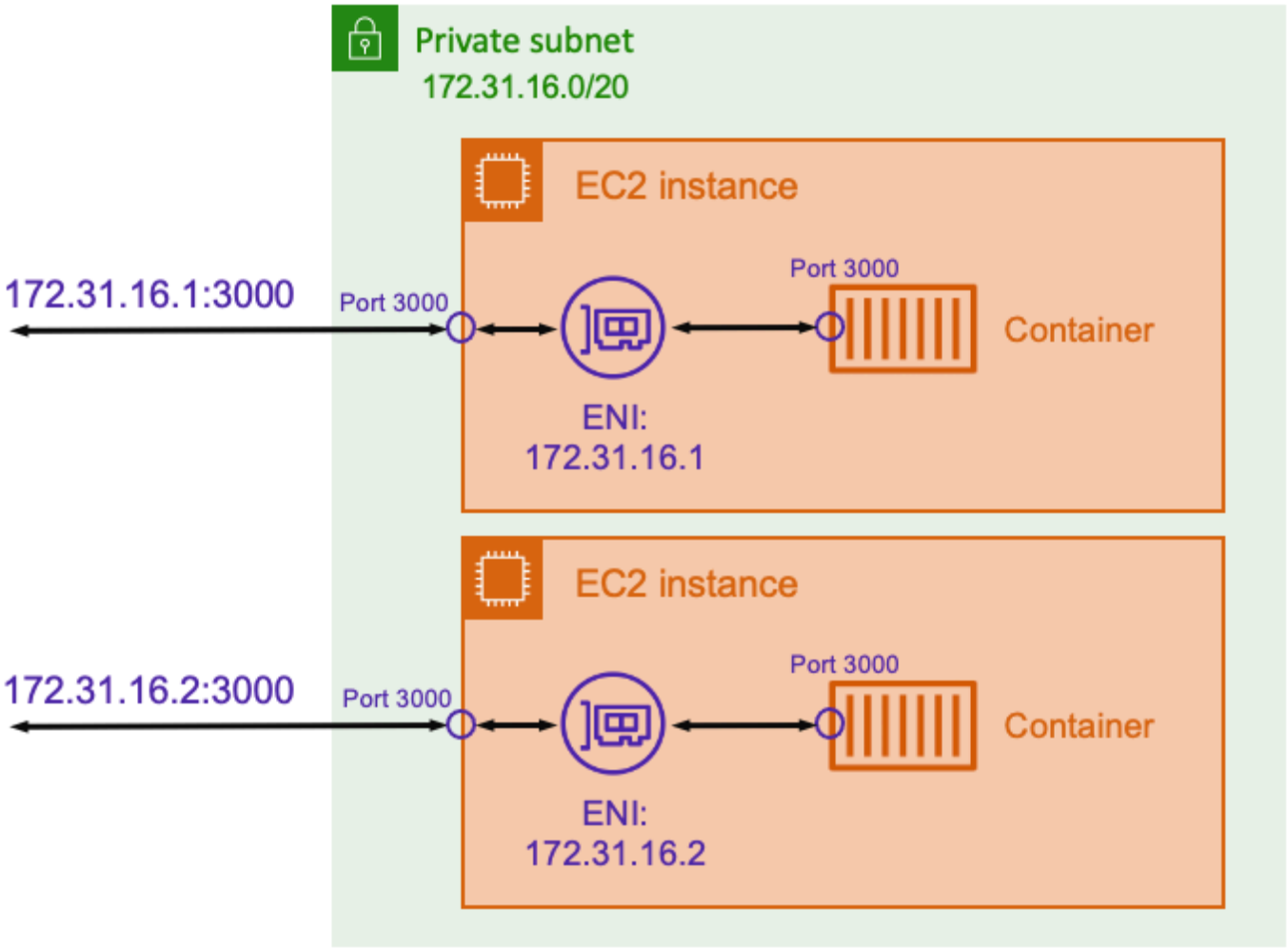
Les approches mentionnées précédemment pour l'architecture des connexions réseau entrantes et sortantes peuvent s'appliquer à n'importe laquelle de vos charges de travail surAWS, même s'ils ne sont pas à l'intérieur d'un conteneur. Lors de l'exécution de conteneurs surAWS, vous devez envisager un autre niveau de mise en réseau. L'un des principaux avantages de l'utilisation de conteneurs est que vous pouvez emballer plusieurs conteneurs sur un seul hôte. Ce faisant, vous devez choisir la façon dont vous souhaitez mettre en réseau les conteneurs qui s'exécutent sur le même hôte. Voici les options à choisir.

Rubriques

- Mode hôte
- Mode pont
- Mode AWSVPC

## Mode hôte

La .`host`est le mode réseau le plus basique pris en charge dans Amazon ECS. En utilisant le mode hôte, la mise en réseau du conteneur est liée directement à l'hôte sous-jacent qui exécute le conteneur.




Supposons que vous exécutez un conteneur Node.js avec une application Express qui écoute sur le port`3000`semblable à celle illustrée dans le schéma précédent. Lorsque le`host`est utilisé, le conteneur reçoit le trafic sur le port 3000 en utilisant l'adresse IP de l'instance Amazon EC2 hôte sous-jacent. Nous ne recommandons pas d'utiliser ce mode.

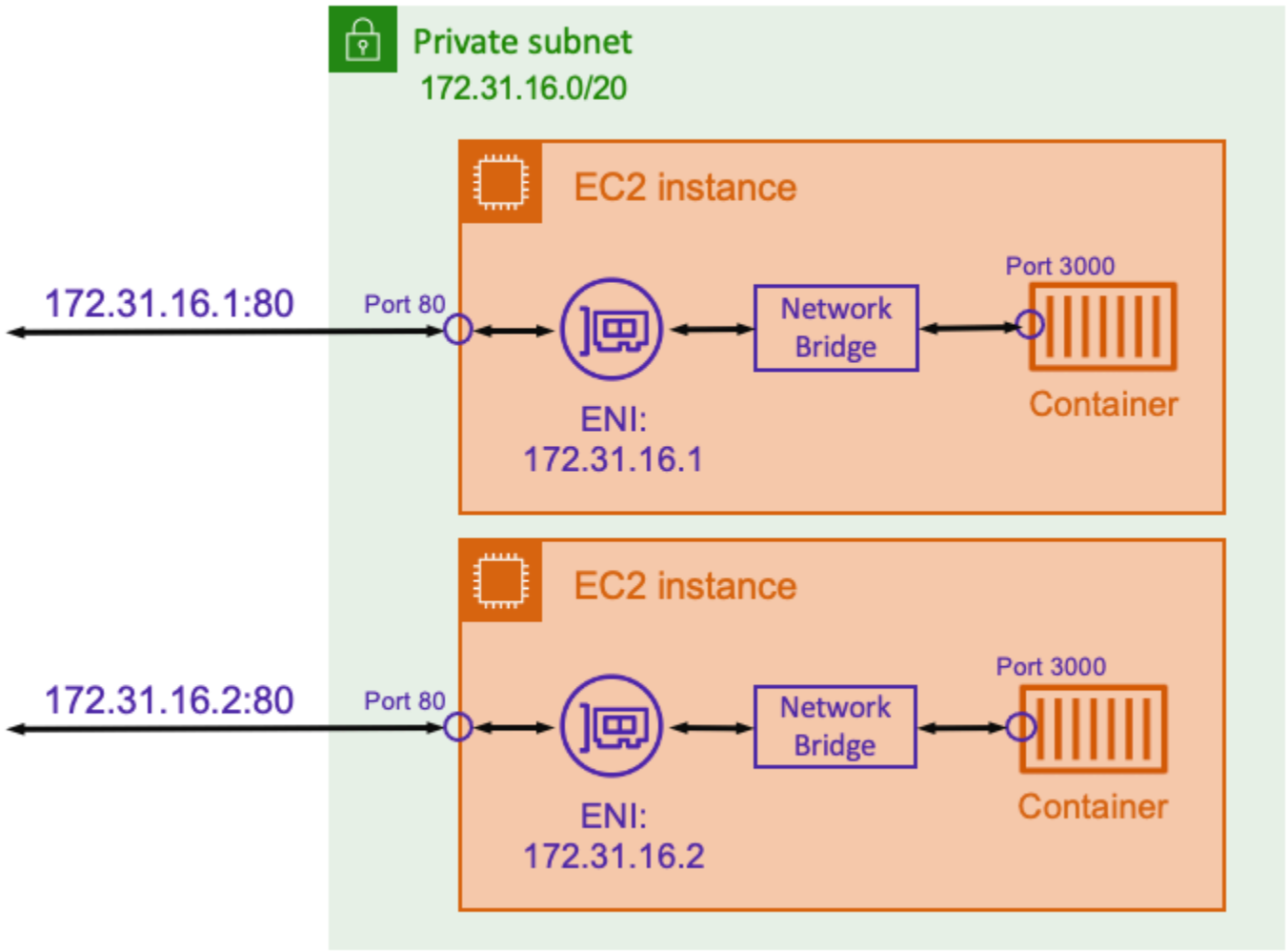
L'utilisation de ce mode réseau présente des inconvénients importants. Vous ne pouvez pas exécuter plus d'une seule instanciation d'une tâche sur chaque hôte. En effet, seule la première tâche peut se lier à son port requis sur l'instance Amazon EC2. Il n'y a pas non plus moyen de remapper un port de conteneur lorsqu'il utilise`host`Mode réseau Par exemple, si une application doit écouter un numéro de port particulier, vous ne pouvez pas remapper le numéro de port directement. Au lieu de cela, vous devez gérer les conflits de port en modifiant la configuration de l'application.

Il y a également des implications pour la sécurité lors de l'utilisation de`host`Mode réseau Ce mode permet aux conteneurs d'emprunter l'identité de l'hôte et permet aux conteneurs de se connecter à des services réseau de bouclage privés sur l'hôte.

La `.hostn`'est pris en charge que pour les tâches Amazon ECS hébergées sur les instances Amazon EC2. Il n'est pas pris en charge lors de l'utilisation d'Amazon ECS sur Fargate.

## Mode pont

avec`bridge`, vous utilisez un pont réseau virtuel pour créer une couche entre l'hôte et la mise en réseau du conteneur. De cette façon, vous pouvez créer des mappages de ports qui remappent un port hôte vers un port de conteneur. Les mappages peuvent être statiques ou dynamiques.

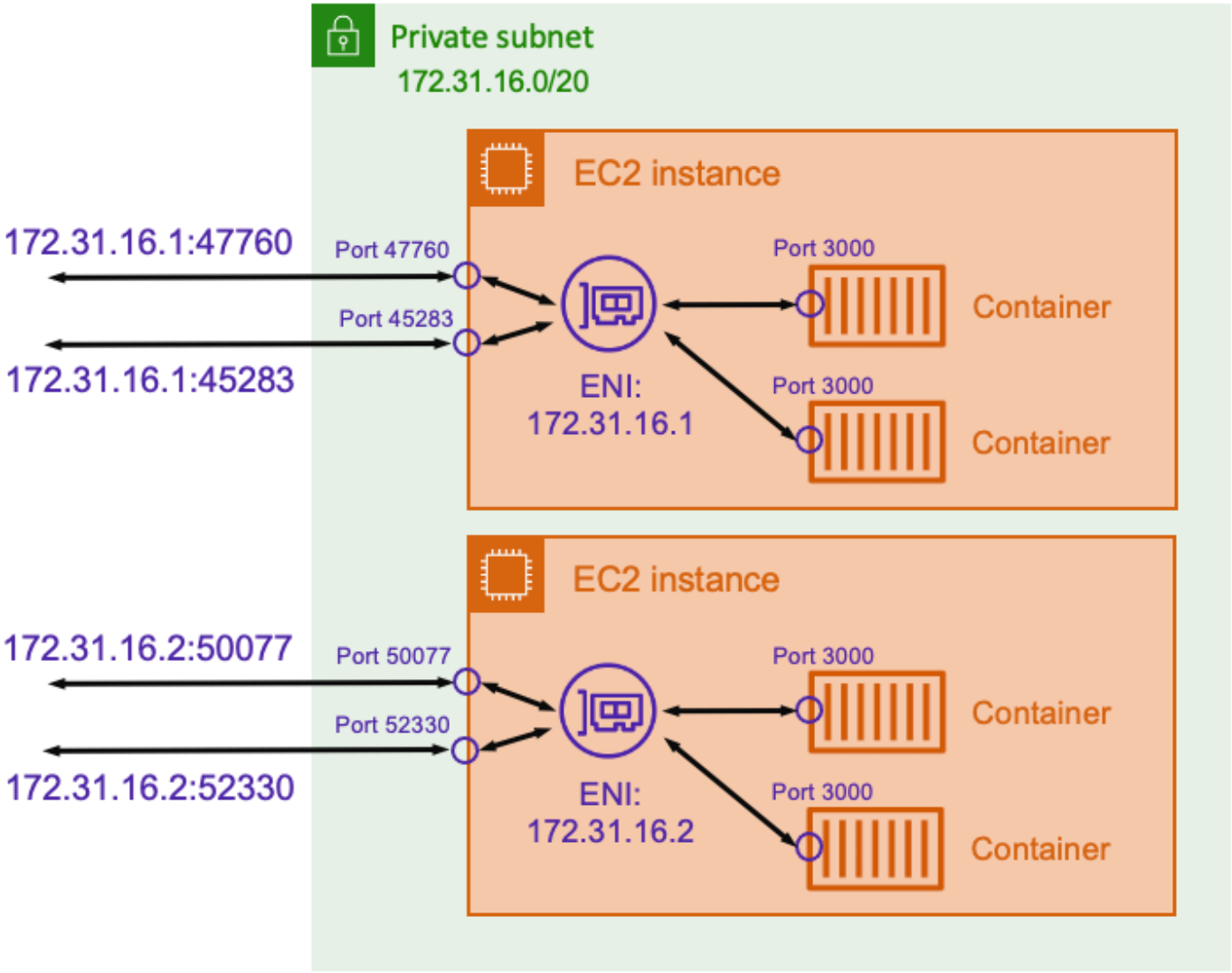



Avec un mappage de port statique, vous pouvez définir explicitement le port hôte que vous souhaitez mapper à un port de conteneur. À l'aide de l'exemple ci-dessus, le port`80`sur l'hôte est mappé

au port`3000`sur le conteneur. Pour communiquer avec l'application conteneurisée, vous envoyez du trafic au port`80`à l'adresse IP de l'instance Amazon EC2. Du point de vue de l'application conteneurisée, il voit que le trafic entrant sur le port`3000`.

Si vous souhaitez uniquement modifier le port de trafic, les mappages de port statiques sont appropriés. Cependant, cela présente toujours le même inconvénient que l'utilisation de la méthode`host`Mode réseau Vous ne pouvez pas exécuter plus d'une seule instanciation d'une tâche sur chaque hôte. En effet, un mappage de port statique permet uniquement de mapper un seul conteneur au port 80.

Pour résoudre ce problème, envisagez d'utiliser la méthode`bridge`Le mode réseau est associé à un mappage de port dynamique comme illustré dans le schéma suivant.



En ne spécifiant pas de port hôte dans le mappage de port, Docker peut choisir un port aléatoire inutilisé dans la plage de ports éphémères et l'affecter comme port hôte public pour le conteneur. Par exemple, l'application Node.js écoute sur le port`3000`sur le conteneur peut se voir attribuer un port à nombre élevé aléatoire tel que`47760`sur l'hôte Amazon EC2. Cela signifie que vous pouvez exécuter plusieurs copies de ce conteneur sur l'hôte. De plus, chaque conteneur peut se voir attribuer son propre port sur l'hôte. Chaque copie du conteneur reçoit du trafic sur le port`3000`. Toutefois, les clients qui envoient du trafic vers ces conteneurs utilisent les ports hôtes attribués de manière aléatoire.

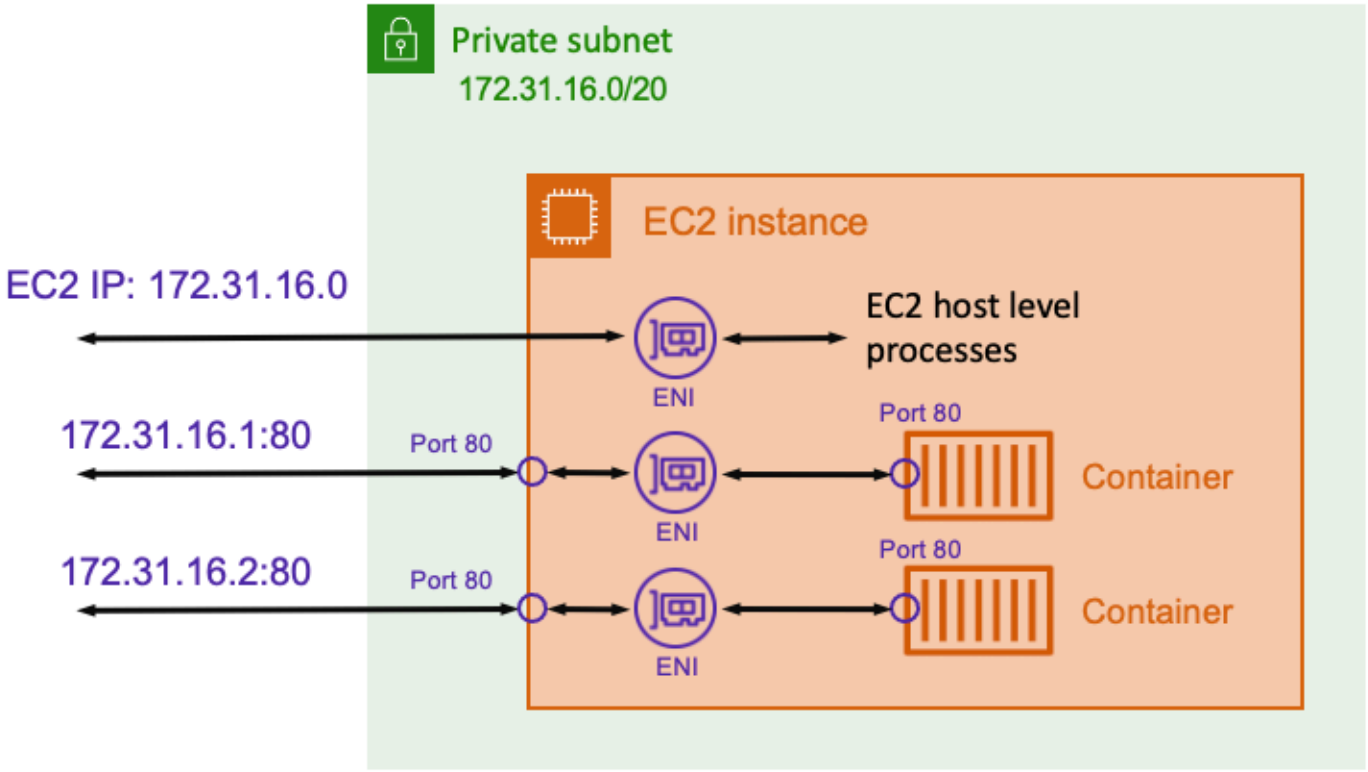
Amazon ECS vous permet de suivre les ports attribués de manière aléatoire pour chaque tâche. Pour ce faire, il met automatiquement à jour les groupes cibles d'équilibrage de charge etAWS Cloud Mappour avoir la liste des adresses IP et des ports de tâche. Il est ainsi plus facile d'utiliser les services fonctionnant à l'aide de`bridge`avec des ports dynamiques.

Cependant, un inconvénient de l'utilisation de la méthode`bridge`est qu'il est difficile de verrouiller le service aux communications de service. Étant donné que les services peuvent être affectés à n'importe quel port aléatoire et inutilisé, il est nécessaire d'ouvrir des plages de ports étendues entre les hôtes. Cependant, il n'est pas facile de créer des règles spécifiques afin qu'un service particulier ne puisse communiquer qu'avec un autre service spécifique. Les services ne disposent pas de ports spécifiques à utiliser pour les règles de mise en réseau des groupes de sécurité.

La `.bridgen`'est pris en charge que pour les tâches Amazon ECS hébergées sur les instances Amazon EC2. Il n'est pas pris en charge lors de l'utilisation d'Amazon ECS sur Fargate.

## Mode AWSVPC

Avec le`awsvpc`, Amazon ECS crée et gère une Elastic Network Interface (ENI) pour chaque tâche et chaque tâche reçoit sa propre adresse IP privée dans le VPC. Cette ENI est distincte des hôtes sous-jacents ENI. Si une instance Amazon EC2 exécute plusieurs tâches, l'ENI de chaque tâche est également distincte.




Dans l'exemple précédent, l'instance Amazon EC2 est affectée à un ENI. L'ENI représente l'adresse IP de l'instance EC2 utilisée pour les communications réseau au niveau de l'hôte. Chaque tâche a également un ENI correspondant et une adresse IP privée. Parce que chaque ENI est séparé, chaque conteneur peut se lier au port80sur la tâche ENI. Par conséquent, vous n'avez pas besoin de suivre les numéros de port. Au lieu de cela, vous pouvez envoyer du trafic vers le port80à l'adresse IP de la tâche ENI.

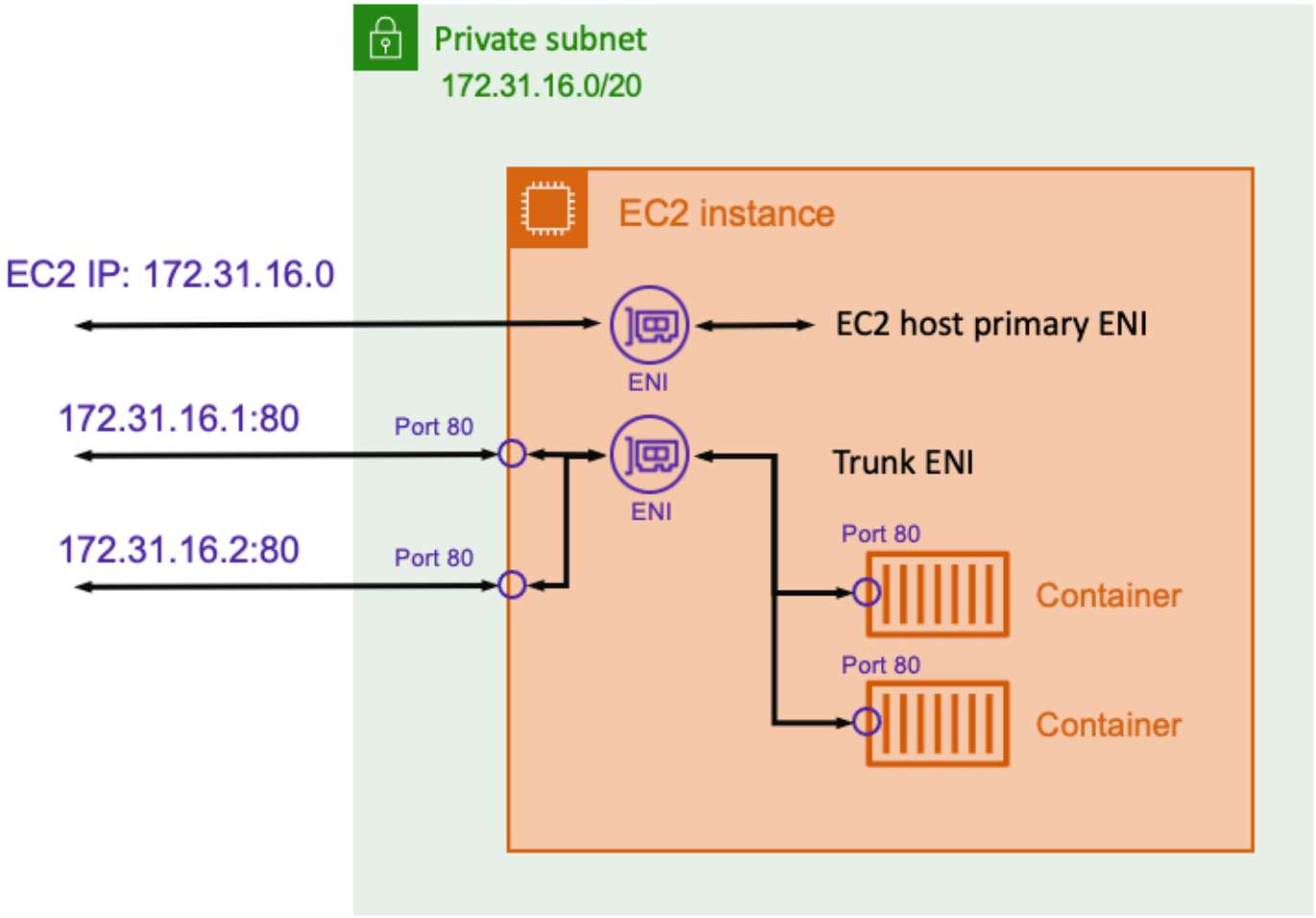
L'avantage d'utiliser leawsvpcest que chaque tâche dispose d'un groupe de sécurité séparé pour autoriser ou refuser le trafic. Cela signifie que vous disposez d'une plus grande flexibilité pour contrôler les communications entre les tâches et les services à un niveau plus précis. Vous pouvez également configurer une tâche pour refuser le trafic entrant d'une autre tâche située sur le même hôte.

La .awsvpcest pris en charge pour les tâches Amazon ECS hébergées sur Amazon EC2 et Fargate. N'oubliez pas que, lorsque vous utilisez Fargate, leawsvpcle mode réseau est requis.

Lorsque vous utilisez leawsvpcil y a quelques défis que vous devez garder à l'esprit.

## Augmentation de la densité des tâches avec ENI Trunking

Le plus grand inconvénient de l'utilisation du`awsvpc`Le mode réseau avec les tâches qui sont hébergées sur les instances Amazon EC2 est que les instances EC2 ont une limite sur le nombre d'ENI qui peuvent être attachées à elles. Cela limite le nombre de tâches que vous pouvez placer sur chaque instance. Amazon ECS fournit la fonctionnalité de liaison ENI qui augmente le nombre d'ENI disponibles pour obtenir une densité de tâches plus élevée.




Lors de l'utilisation de la liaison ENI, deux pièces jointes ENI sont utilisées par défaut. Le premier est l'ENI primaire de l'instance, qui est utilisé pour tous les processus de niveau hôte. Le second est le tronc ENI, créé par Amazon ECS. Cette fonction est prise en charge uniquement pour des types d'instance Amazon EC2 spécifiques.

Considérez cet exemple. Sans réseau ENI, un`c5.large`qui a deux vCPUs ne peut héberger que deux tâches. Cependant, avec les liaisons ENI, un`c5.large`qui a deux vCPU peut héberger jusqu'à dix tâches. Chaque tâche a une adresse IP et un groupe de sécurité différents. Pour plus

d'informations sur les types d'instance disponibles et leur densité, consultezTypes d'instances Amazon EC2 pris en chargedans leGuide du développeur Amazon Elastic Container Service.

La liaison ENI n'a aucun impact sur les performances d'exécution en termes de latence ou de bande passante. Cependant, cela augmente le temps de démarrage de la tâche. Vous devez vous assurer que, si la liaison ENI est utilisée, vos règles de mise à l'échelle automatique et les autres charges de travail qui dépendent du temps de démarrage de la tâche fonctionnent toujours comme vous le souhaitez.

Pour de plus amples informations, veuillez consulterConnexion d'interface réseau Elasticdans leGuide du développeur Amazon Elastic Container Service.

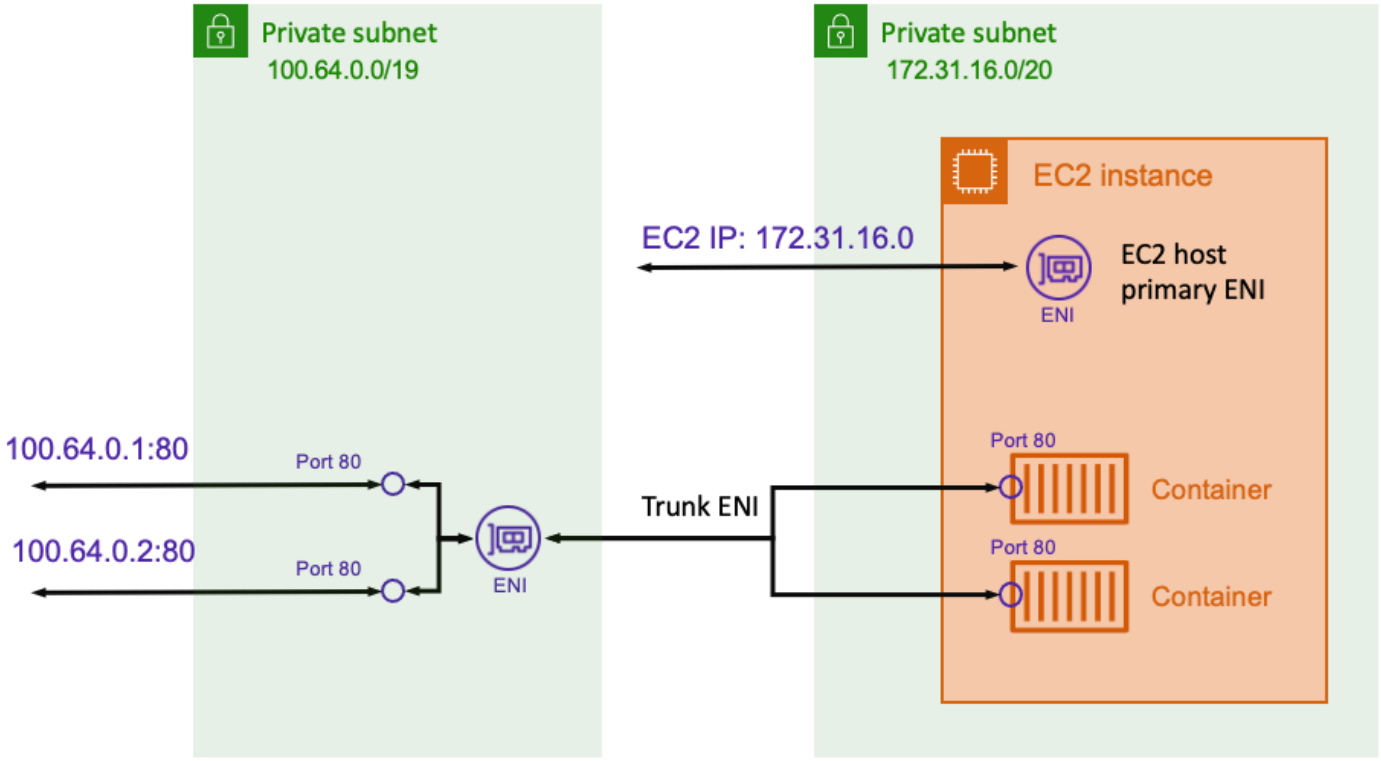
## Prévenir l'épuisement de l'adresse IP

En attribuant une adresse IP distincte à chaque tâche, vous pouvez simplifier votre infrastructure globale et maintenir des groupes de sécurité offrant un niveau de sécurité élevé. Cependant, cette configuration peut conduire à l'épuisement de l'adresse IP.

Le VPC par défaut sur votreAWSa des sous-réseaux préprovisionnés qui ont un`/20`Plage CIDR Cela signifie que chaque sous-réseau a 4 091 adresses IP disponibles. Notez que plusieurs adresses IP dans le`/20`sont réservées pour une utilisation spécifique AWS. Considérez cet exemple. Vous distribuez vos applications sur trois sous-réseaux dans trois zones de disponibilité pour une haute disponibilité. Dans ce cas, vous pouvez utiliser environ 12 000 adresses IP sur les trois sous-réseaux.

À l'aide de la liaison ENI, chaque instance Amazon EC2 que vous lancez nécessite deux adresses IP. Une adresse IP est utilisée pour l'ENI principal, et l'autre adresse IP est utilisée pour l'ENI principal. Chaque tâche Amazon ECS sur l'instance nécessite une adresse IP. Si vous lancez une charge de travail extrêmement importante, vous risquez de manquer d'adresses IP disponibles. Cela peut entraîner des échecs de lancement Amazon EC2 ou des échecs de lancement de tâche. Ces erreurs se produisent car les ENI ne peuvent pas ajouter d'adresses IP à l'intérieur du VPC s'il n'y a pas d'adresses IP disponibles.

Lorsque vous utilisez le`awsvpc`, vous devez évaluer vos besoins en adresse IP et vous assurer que vos plages CIDR de sous-réseau répondent à vos besoins. Si vous avez déjà commencé à utiliser un VPC qui a de petits sous-réseaux et commence à manquer d'espace d'adressage, vous pouvez ajouter un sous-réseau secondaire.




En utilisant la liaison ENI, le CNI Amazon VPC peut être configuré pour utiliser les ENI dans un espace d'adressage IP différent de celui de l'hôte. Ce faisant, vous pouvez donner à votre hôte Amazon EC2 et à vos tâches différentes plages d'adresses IP qui ne se chevauchent pas. Dans le diagramme d'exemple, l'adresse IP de l'hôte EC2 se trouve dans un sous-réseau qui a la propriété`172.31.16.0/20`Plage IP. Toutefois, les tâches qui s'exécutent sur l'hôte se voient attribuer des adresses IP dans le`100.64.0.0/19`Plage. En utilisant deux plages IP indépendantes, vous n'avez pas à vous soucier des tâches qui consomment trop d'adresses IP et ne laissent pas assez d'adresses IP pour les instances.

## Utilisation du mode double pile IPv6

La `.awsvpc`est compatible avec les VPC configurés pour le mode double pile IPv6. Un VPC utilisant le mode double pile peut communiquer via IPv4, IPv6 ou les deux. Chaque sous-réseau du VPC peut avoir à la fois une plage CIDR IPv4 et une plage CIDR IPv6. Pour de plus amples informations, veuillez consulterAdressage IP dans votre VPCdans leManuel de l'utilisateur Amazon VPC.

Vous ne pouvez pas désactiver la prise en charge d'IPv4 de votre VPC et de vos sous-réseaux afin de résoudre les problèmes d'épuisement IPv4. Cependant, avec la prise en charge IPv6, vous pouvez utiliser de nouvelles fonctionnalités, en particulier la passerelle Internet de sortie

uniquement. Une passerelle Internet de sortie uniquement permet aux tâches d'utiliser leur adresse IPv6 publiquement routable pour lancer des connexions sortantes à Internet. Mais la passerelle Internet de sortie uniquement n'autorise pas les connexions depuis Internet. Pour de plus amples informations, veuillez consulterPasserelles Internet de sortie uniquementdans leManuel de l'utilisateur Amazon VPC.

# Connexion àAWSà partir de votre VPC

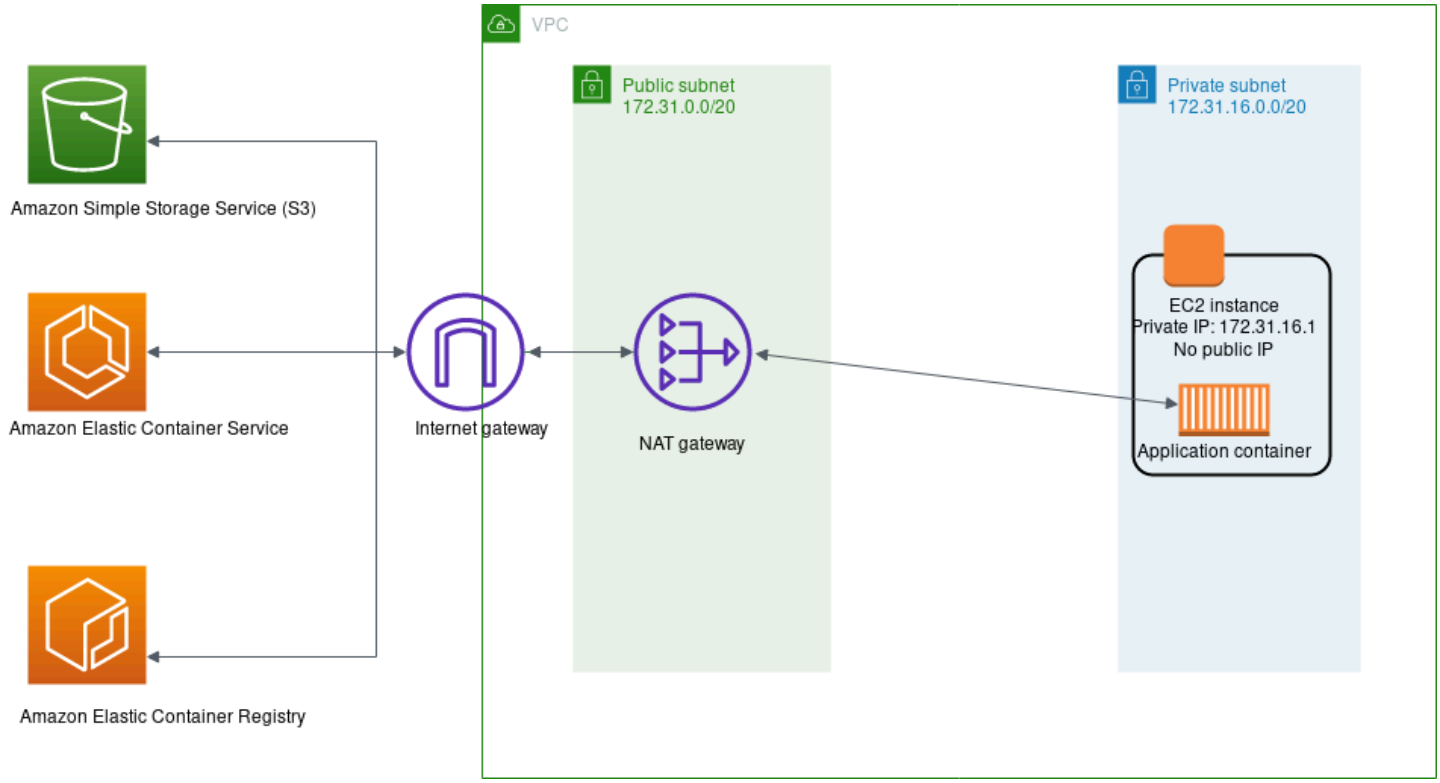
Pour qu'Amazon ECS fonctionne correctement, l'agent de conteneur ECS qui s'exécute sur chaque hôte doit communiquer avec le plan de contrôle Amazon ECS. Si vous stockez vos images de conteneur dans Amazon ECR, les hôtes Amazon EC2 doivent communiquer avec le point de terminaison du service Amazon ECR et Amazon S3, où les couches d'images sont stockées. Si vous utilisez d'autresAWSpour votre application conteneurisée, comme les données persistantes stockées dans DynamoDB, vérifiez que ces services disposent également de la prise en charge réseau nécessaire.

Rubriques

- Passerelle NAT
- AWS PrivateLink

## Passerelle NAT

L'utilisation d'une passerelle NAT est le moyen le plus simple de vous assurer que vos tâches Amazon ECS peuvent accéder à d'autresAWSServices . Pour de plus amples informations sur cette approche, veuillez consulterUtilisation d'un sous-réseau privé et d'une passerelle NAT.




Voici les inconvénients de l'utilisation de cette approche :

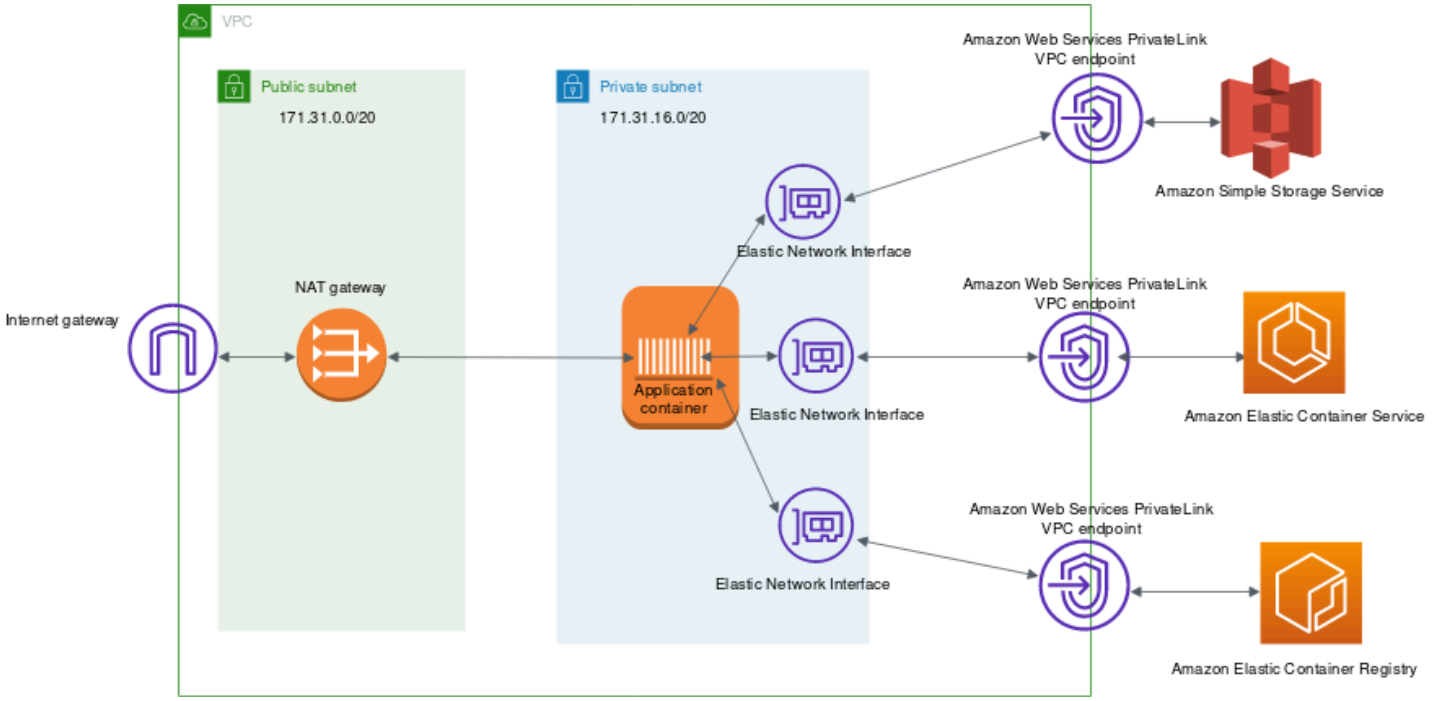
- Vous ne pouvez pas limiter les destinations avec lesquelles la passerelle NAT peut communiquer. Vous ne pouvez pas non plus limiter les destinations vers lesquelles votre pneu principal peut communiquer sans perturber toutes les communications sortantes de votre VPC.
- Les passerelles NAT facturent chaque Go de données qui transitent. Si vous utilisez la passerelle NAT pour télécharger des fichiers volumineux à partir d'Amazon S3 ou pour effectuer un volume élevé de requêtes de base de données vers DynamoDB, vous êtes facturé pour chaque Go de bande passante. De plus, les passerelles NAT prennent en charge jusqu'à 5 Gb/s de bande passante et sont mises à l'échelle automatiquement jusqu'à 45 Gb/s. Si vous routez via une seule passerelle NAT, les applications qui nécessitent des connexions à très large bande passante peuvent rencontrer des contraintes de mise en réseau. Pour contourner le problème, vous pouvez diviser votre charge de travail entre plusieurs sous-réseaux et donner à chaque sous-réseau sa propre passerelle NAT.

## AWS PrivateLink

AWS PrivateLinkfournit une connectivité privée entre les VPC,AWSet vos réseaux locaux sans exposer votre trafic à l'Internet public.

L'une des technologies utilisées pour y parvenir est le point de terminaison VPC. Le point de terminaison d'un VPC permet de se connecter confidentiellement à des connexions privées entre votre VPCAWSet services de points de terminaison d'un VPC. Le trafic entre votre VPC et les autres services ne quitte pas le réseau Amazon. Le point de terminaison d'un VPC n'a pas besoin d'une passerelle Internet, d'une passerelle réseau privé virtuel, d'un périphérique NAT, d'une connexion VPN ou d'uneAWS Direct ConnectConnexion Les instances Amazon EC2 de votre VPC ne requièrent pas d'adresses IP publiques pour communiquer avec les ressources du service.

Le schéma suivant illustre la manière dont la communication versAWSfonctionne lorsque vous utilisez des points de terminaison VPC au lieu d'une passerelle Internet.AWS PrivateLinkfournit des interfaces réseau élastiques (ENI) à l'intérieur du sous-réseau, et les règles de routage VPC sont utilisées pour envoyer toute communication au nom d'hôte du service via l'ENI, directement à la destinationAWSservice. Ce trafic n'a plus besoin d'utiliser la passerelle NAT ou la passerelle Internet.




Voici quelques-uns des points de terminaison VPC courants utilisés avec le service Amazon ECS.

- Point de terminaison de VPC de passerelle S3
- Point de terminaison d'un VPC DynamoDB
- Point de terminVPC ECS
- Point de terminVPC ECR

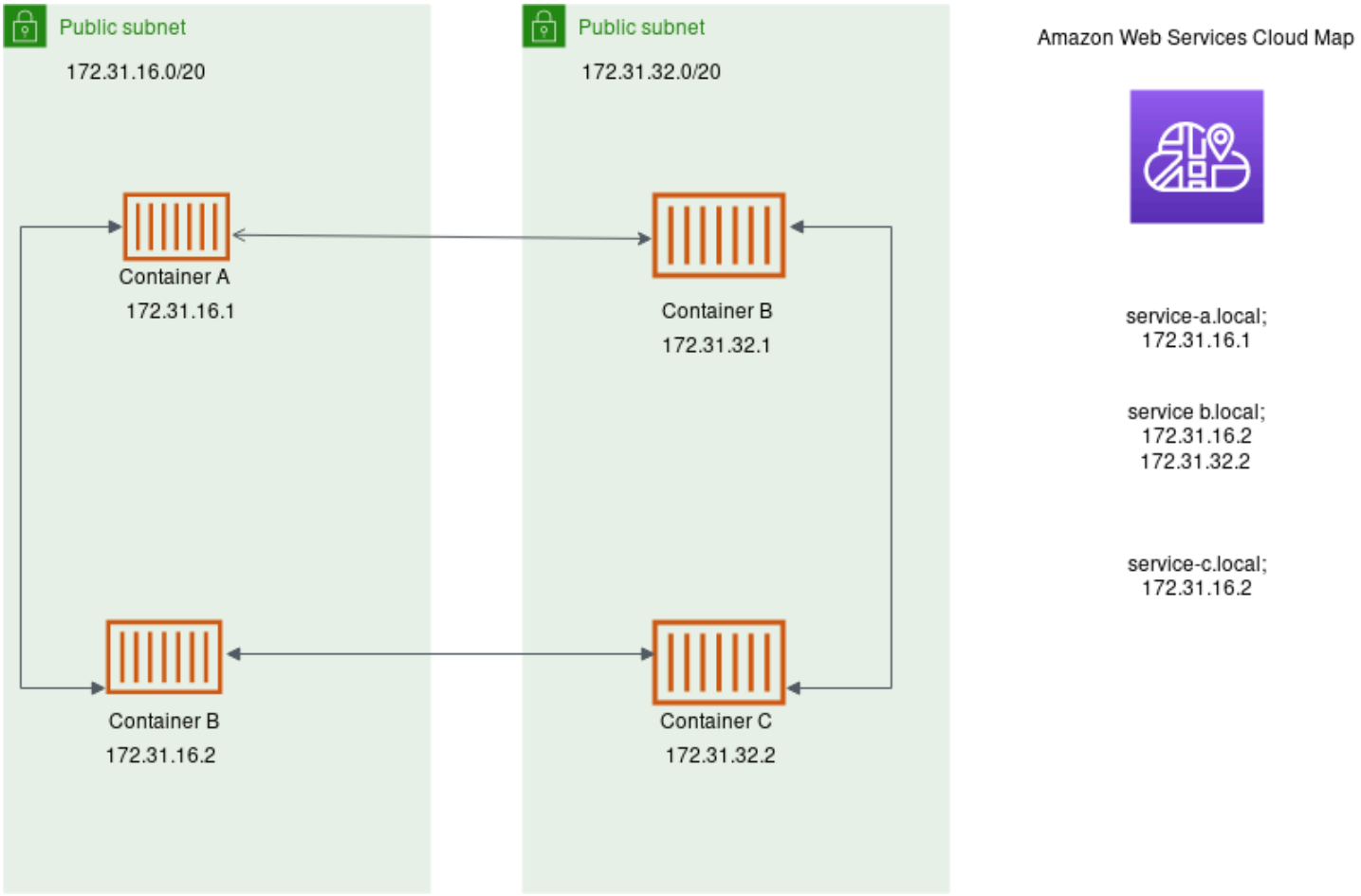
Nombreux autresAWSprennent en charge les points de terminaison de VPC. Si vous faites un usage intensif de toutAWS, vous devez rechercher la documentation spécifique de ce service et comment créer un point de terminaison VPC pour ce trafic.

# Mise en réseau entre les services Amazon ECS dans un VPC

En utilisant les conteneurs Amazon ECS dans un VPC, vous pouvez déverser des applications monolithiques dans des parties distinctes qui peuvent être déployées et mises à l'échelle indépendamment dans un environnement sécurisé. Cependant, il peut être difficile de s'assurer que toutes ces parties, à l'intérieur et à l'extérieur d'un VPC, peuvent communiquer entre elles. Il existe plusieurs approches pour faciliter la communication, toutes avec des avantages et des inconvénients différents.

## Utilisation de la découverte de service

Une approche de communication de service à service est la communication directe grâce à la découverte de service. Dans cette approche, vous pouvez utiliser la méthodeAWS Cloud Mapintégration de découverte de service avec Amazon ECS. À l'aide de la découverte de service, Amazon ECS synchronise la liste des tâches lancées surAWS Cloud Map, qui gère un nom d'hôte DNS qui se résout en adresses IP internes d'une ou de plusieurs tâches de ce service particulier. D'autres services dans Amazon VPC peuvent utiliser ce nom d'hôte DNS pour envoyer du trafic directement à un autre conteneur en utilisant son adresse IP interne. Pour de plus amples informations, veuillez consulterDécouverte de servicedans leGuide du développeur Amazon Elastic Container Service.




Dans le schéma précédent, il existe trois services.`serviceA`a un conteneur et communique avec`serviceB`, qui a deux conteneurs.`serviceB`doit également communiquer avec`serviceC`, qui a un conteneur. Chaque conteneur de ces trois services peut utiliser les noms DNS internes deAWS Cloud Mappour trouver les adresses IP internes d'un conteneur à partir du service en aval auquel il doit communiquer.

Cette approche de la communication service-service fournit une faible latence. À première vue, c'est aussi simple car il n'y a pas de composants supplémentaires entre les conteneurs. Le trafic circule directement d'un conteneur à l'autre conteneur.

Cette approche est appropriée lors de l'utilisation de la méthode`awsvpc`, où chaque tâche a sa propre adresse IP unique. La plupart des logiciels ne prennent en charge que l'utilisation de DNSA, qui se résolvent directement en adresses IP. Lorsque vous utilisez le`awsvpc`, l'adresse IP de chaque tâche est unAEnregistrement. Cependant, si vous utilisez`bridge`, plusieurs conteneurs peuvent partager la même adresse IP. En outre, les mappages de ports dynamiques entraînent l'attribution aléatoire de numéros de port aux conteneurs sur cette seule adresse IP. À ce stade, unAn'est plus suffisant pour la découverte de service. Vous devez également utiliser unSRVEnregistrement. Ce type

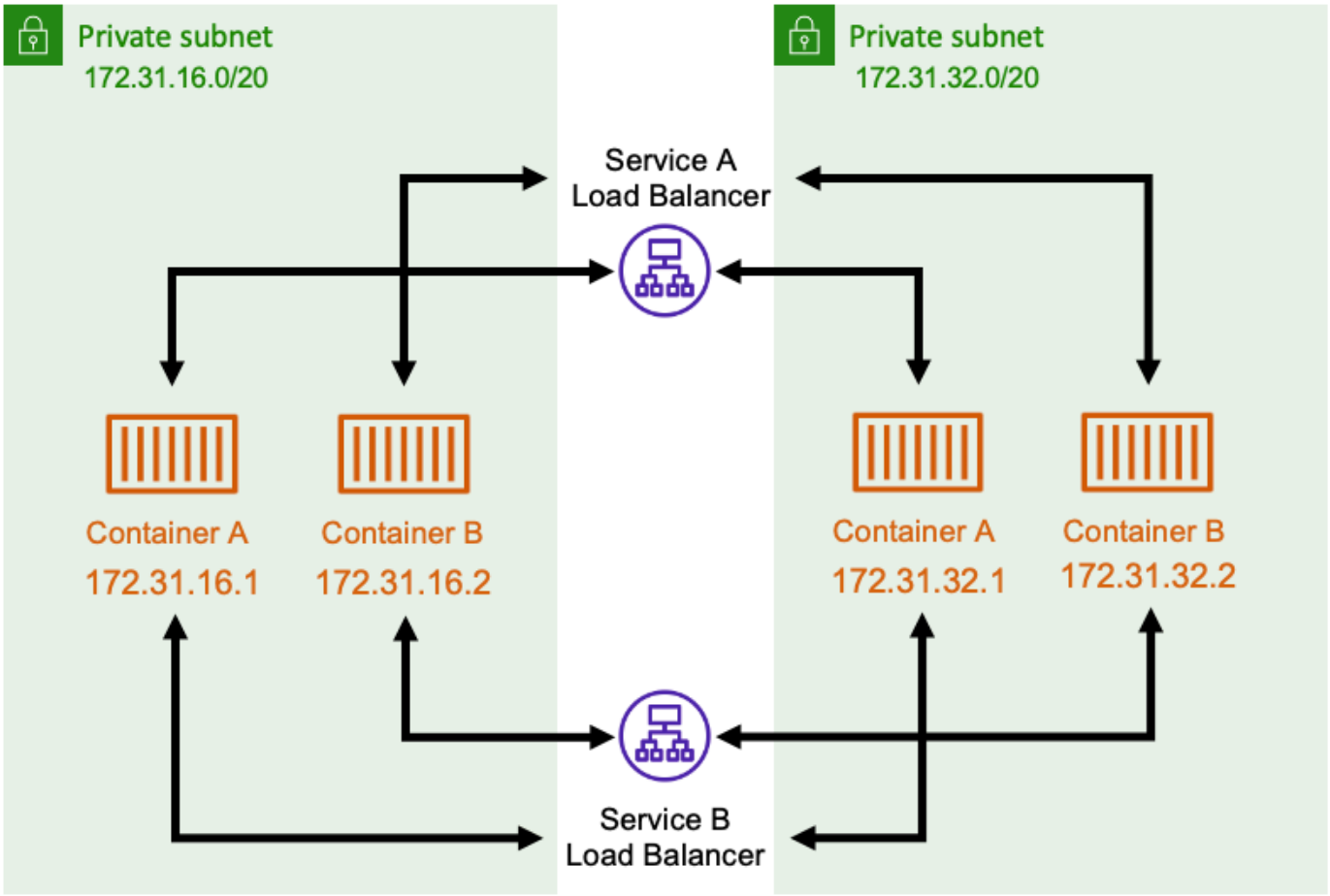
d'enregistrement peut garder une trace des adresses IP et des numéros de port, mais nécessite que vous configurez les applications de manière appropriée. Certaines applications prédéfinies que vous utilisez peuvent ne pas prendre en charge`SRV`Enregistrements.

Un autre avantage de la`awsvpc`est que vous disposez d'un groupe de sécurité unique pour chaque service. Vous pouvez configurer ce groupe de sécurité pour autoriser les connexions entrantes uniquement à partir des services en amont spécifiques qui doivent communiquer avec ce service.

Le principal inconvénient de la communication directe service-service à l'aide de la découverte de service est que vous devez implémenter une logique supplémentaire pour avoir de nouvelles tentatives et gérer les échecs de connexion. Les enregistrements DNS ont une période de durée de vie (TL) qui contrôle combien de temps ils sont mis en cache. Il faut un certain temps pour que l'enregistrement DNS soit mis à jour et que le cache expire afin que vos applications puissent récupérer la dernière version de l'enregistrement DNS. Ainsi, votre application peut finir par résoudre l'enregistrement DNS pour pointer vers un autre conteneur qui n'est plus là. Votre application doit gérer les nouvelles tentatives et avoir une logique pour ignorer les mauvais moteurs.

## Utilisation d'un équilibreur de charge interne

Une autre approche de la communication service-service consiste à utiliser un équilibreur de charge interne. Un équilibreur de charge interne existe entièrement à l'intérieur de votre VPC et n'est accessible qu'aux services à l'intérieur de votre VPC.




L'équilibreur de charge maintient une haute disponibilité en déployant des ressources redondantes dans chaque sous-réseau. Lorsqu'un conteneur de`service`Adoit communiquer avec un conteneur de`service`B, il ouvre une connexion à l'équilibreur de charge. L'équilibreur de charge ouvre alors une connexion à un conteneur depuis`service B`. L'équilibreur de charge sert de lieu centralisé pour gérer toutes les connexions entre chaque service.

Si un conteneur de`service`Bs'arrête, l'équilibreur de charge peut supprimer ce conteneur du pool. L'équilibreur de charge effectue également des contrôles de santé sur chaque cible en aval de son pool et peut supprimer automatiquement les cibles défectueuses du pool jusqu'à ce qu'elles redeviennent saines. Les applications n'ont plus besoin de connaître le nombre de conteneurs en aval. Ils ouvrent leurs connexions à l'équilibreur de charge.

Cette approche est avantageuse pour tous les modes de réseau. L'équilibreur de charge peut garder le suivi des adresses IP de tâche lors de l'utilisation de`awsvpc`, ainsi que des combinaisons plus avancées d'adresse IP et de port lors de l'utilisation du`bridge`Mode réseau Il répartit uniformément

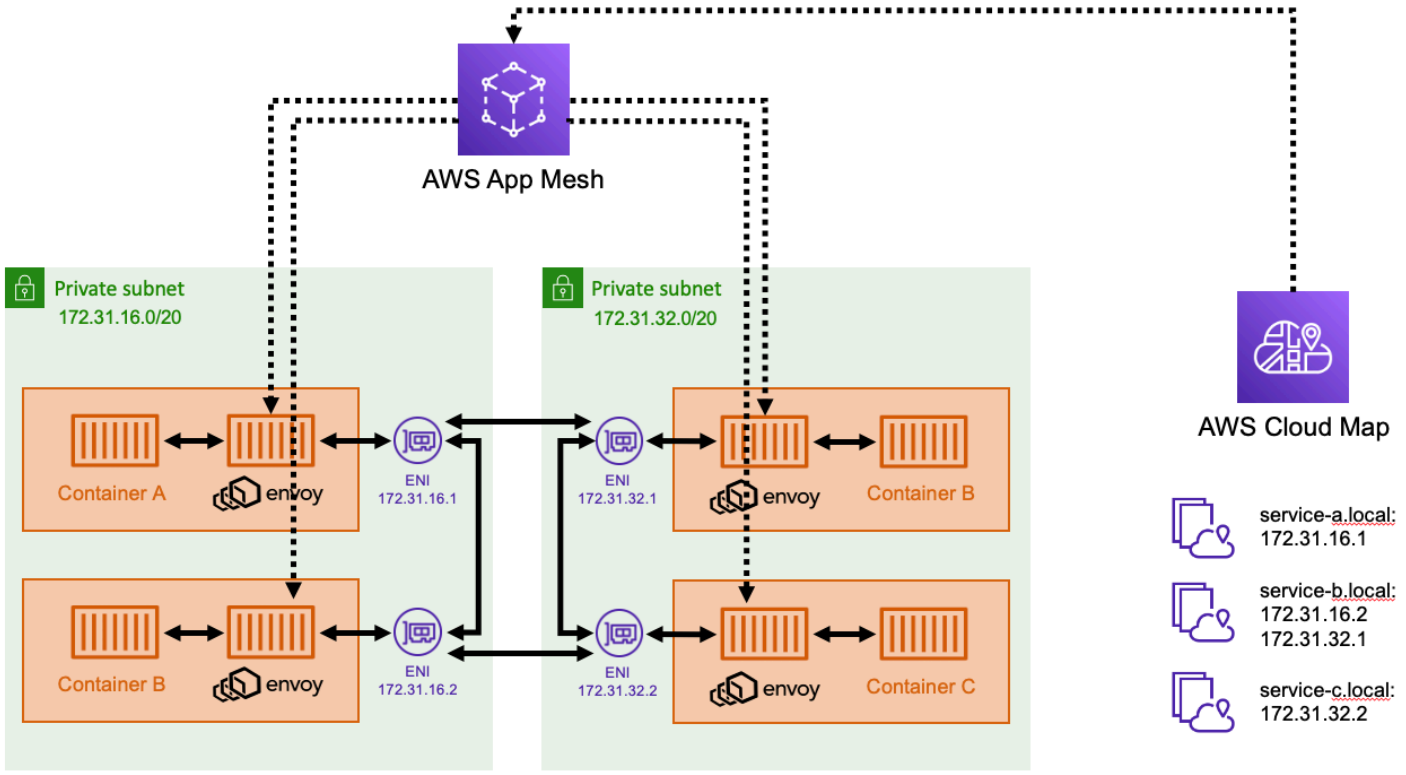
le trafic entre toutes les combinaisons d'adresses IP et de ports, même si plusieurs conteneurs sont réellement hébergés sur la même instance Amazon EC2, uniquement sur des ports différents.

Le seul inconvénient de cette approche est le coût. Pour être hautement disponible, l'équilibreur de charge doit disposer de ressources dans chaque zone de disponibilité. Cela ajoute des coûts supplémentaires en raison des frais généraux liés au paiement de l'équilibreur de charge et de la quantité de trafic qui passe par l'équilibreur de charge.

Toutefois, vous pouvez réduire les frais généraux en faisant partager plusieurs services un équilibreur de charge. Ceci est particulièrement approprié pour les services REST qui utilisent un équilibreur de charge d'application. Vous pouvez créer des règles de routage basées sur le chemin qui acheminent le trafic vers différents services. Par exemple,`/api/user/*`peut acheminer vers un conteneur qui fait partie de la`user`service, alors que`/api/order/*`peut acheminer vers le`order`service. Avec cette approche, vous ne payez qu'un seul équilibreur de charge d'application et disposez d'une URL cohérente pour votre API. Cependant, vous pouvez diviser le trafic vers divers microservices sur le backend.

## Utilisation d'un mesh de service

AWS App Meshest un maillage de services qui peut vous aider à gérer un grand nombre de services et à mieux contrôler la façon dont le trafic est acheminé entre les services. App Mesh fonctionne comme un intermédiaire entre la découverte de service de base et l'équilibrage de charge. Avec App Mesh, les applications n'interagissent pas directement les unes avec les autres, mais elles n'utilisent pas non plus d'équilibrage de charge centralisé. Au lieu de cela, chaque copie de votre tâche est accompagnée d'un side-car proxy Envoy. Pour de plus amples informations, veuillez consulter Présentation d'AWS App Mesh dans le Guide de l'utilisateur AWS App Mesh.




Dans le schéma précédent, chaque tâche a un side-car de proxy Envoy. Ce side-car est responsable de la transmission par proxy de tout le trafic entrant et sortant pour la tâche. Le plan de contrôle App Mesh utiliseAWS Cloud Mappour obtenir la liste des services disponibles et les adresses IP de tâches spécifiques. Ensuite, App Mesh fournit la configuration au side-car du proxy Envoy. Cette configuration inclut la liste des conteneurs disponibles auxquels vous pouvez connecter. Le sidecar proxy Envoy effectue également des vérifications de l'état de chaque cible pour s'assurer qu'elles sont disponibles.

Cette approche fournit les fonctionnalités de découverte de service, avec la facilité de l'équilibrage de charge géré. Les applications n'implémentent pas autant de logique d'équilibrage de charge dans leur code car le sidecar proxy Envoy gère cet équilibrage de charge. Le proxy Envoy peut être configuré pour détecter les échecs et réessayer les demandes échouées. En outre, il peut également être configuré pour utiliser des MTL pour chiffrer le trafic en transit et s'assurer que vos applications communiquent à une destination vérifiée.

Il existe peu de différences entre un proxy Envoy et un équilibreur de charge. En bref, avec le proxy Envoy, vous êtes responsable du déploiement et de la gestion de votre propre sidecar proxy Envoy. Le sidecar proxy Envoy utilise une partie du CPU et de la mémoire que vous allouez à la tâche

Amazon ECS. Cela ajoute une surcharge à la consommation de ressources de la tâche, ainsi qu'une charge de travail opérationnelle supplémentaire pour maintenir et mettre à jour le proxy si nécessaire.

App Mesh et un proxy Envoy permettent une latence extrêmement faible entre les tâches. Cela est dû au fait que le proxy Envoy s'exécute collocalisé à chaque tâche. Il n'y a qu'une seule instance pour l'instance de saut réseau, entre un proxy Envoy et un autre proxy Envoy. Cela signifie qu'il y a également moins de frais de réseau que lors de l'utilisation d'équilibreurs de charge. Lors de l'utilisation d'équilibreurs de charge, il y a deux sauts réseau. La première est de la tâche en amont à l'équilibreur de charge, et la seconde est de l'équilibreur de charge à la tâche en aval.

## Services de mise en réseauAWScomptes et VPC

Si vous faites partie d'une organisation avec plusieurs équipes et divisions, vous déployez probablement des services indépendamment dans des VPC distincts au sein d'unAWSou dans des VPC qui sont associés à plusieursAWS. Quelle que soit la manière dont vous déployez vos services, nous vous recommandons de compléter vos composants réseau pour faciliter l'acheminement du trafic entre les VPC. Pour cela, plusieursAWSpeuvent être utilisés pour compléter vos composants réseau existants.

- AWS Transit Gateway : vous devez d'abord considérer ce service de mise en réseau. Ce service sert de hub central pour le routage de vos connexions entre les VPC Amazon,AWScomptes et réseaux locaux. Pour de plus amples informations, veuillez consulterQu'est-ce qu'une passerelle de transit ?dans leGuide des passerelles de transit Amazon VPC.
- Prise en charge VPC et VPN Amazon : vous pouvez utiliser ce service pour créer des connexions VPN de site à site pour connecter des réseaux locaux à votre VPC. Pour plus d'informations, consultez Présentation de AWS Site-to-Site VPN dans le Guide de l'utilisateur AWS Site-to-Site VPN.
- Amazon VPC — Vous pouvez utiliser l'appairage Amazon VPC pour vous aider à connecter plusieurs VPC, soit dans le même compte, soit entre plusieurs comptes. Pour de plus amples informations, veuillez consulter Qu'est-ce que l'appairage de VPC ? dans le Amazon VPC Peering Guide.
- VPC partagés : vous pouvez utiliser un VPC et des sous-réseaux VPC sur plusieursAWS. Pour de plus amples informations, veuillez consulterUtilisation des VPC partagésdans leManuel de l'utilisateur Amazon VPC.

# Optimisation et dépannage

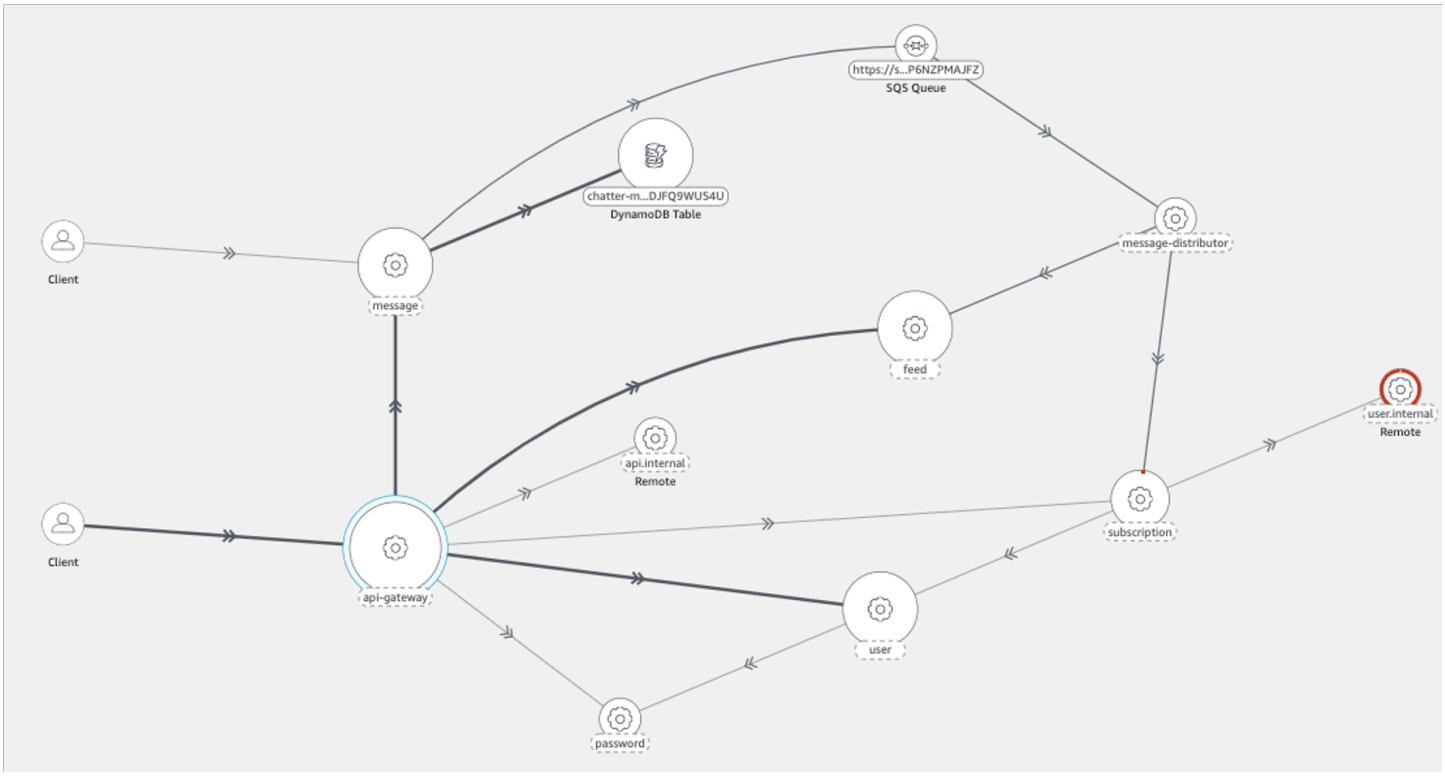
Les services et fonctionnalités suivants peuvent vous aider à obtenir des informations sur les configurations de votre réseau et de vos services. Vous pouvez utiliser ces informations afin de résoudre les problèmes de réseau et d'optimiser vos services.

## Informations sur le conteneur CloudWatch

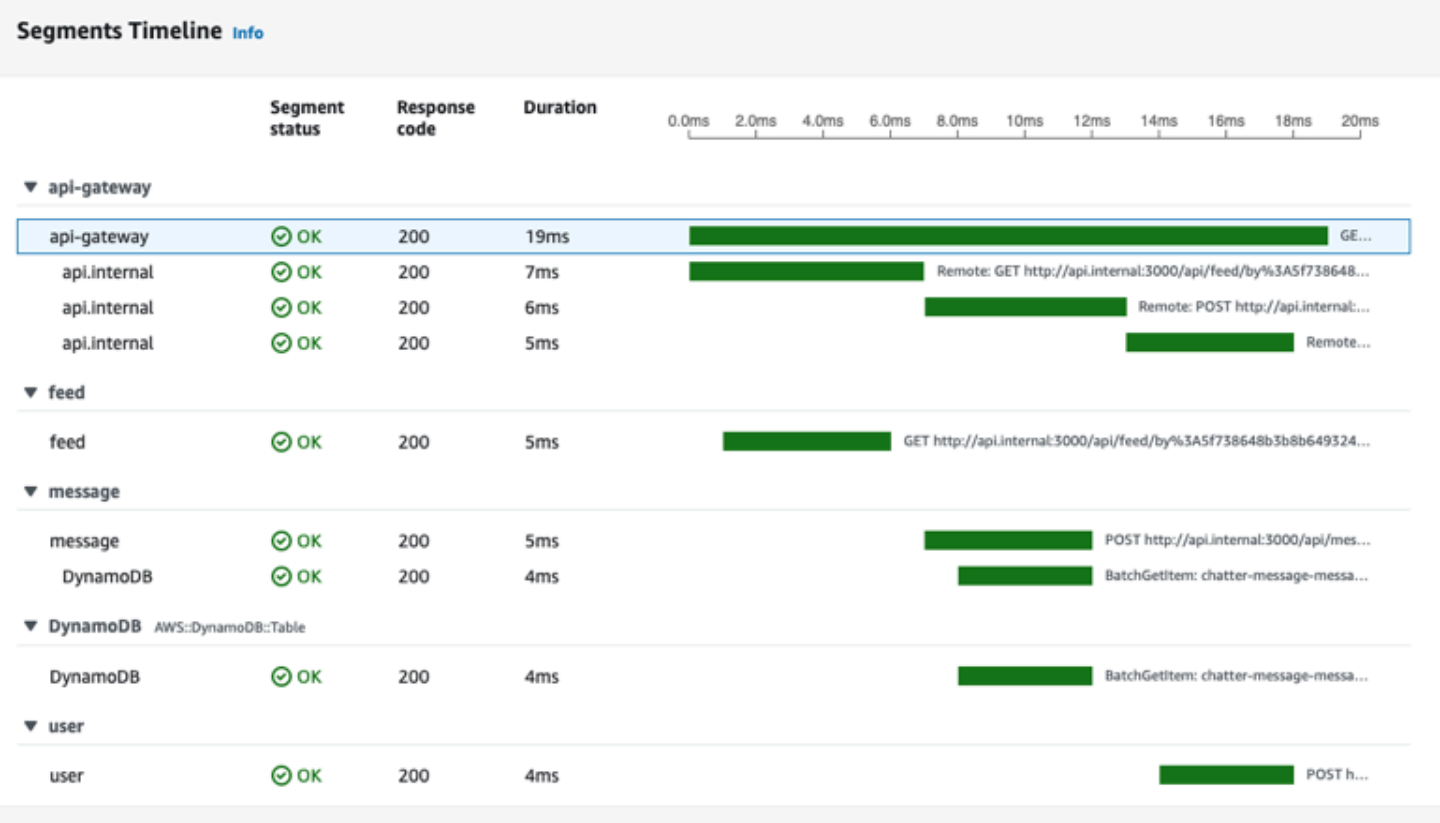
CloudWatch Containeur Insights collecte, regroupe et récapitule les métriques et les journaux de vos applications et microservices conteneurisées. Les métriques incluent l'utilisation de ressources telles que l'UC, la mémoire, le disque et le réseau. Ils sont disponibles dans les tableaux de bord automatiques CloudWatch. Pour de plus amples informations, veuillez consulterConfiguration de Container Insights sur Amazon ECSdans leGuide de l'utilisateur Amazon CloudWatch..

## AWS X-Ray

AWS X-Rayest un service de suivi que vous pouvez utiliser pour collecter des informations sur les demandes réseau que votre application effectue. Vous pouvez utiliser le SDK pour instrumenter votre application et capturer les horaires et les codes de réponse du trafic entre vos services, et entre vos services etAWSPoints de terminaison du service. Pour de plus amples informations, veuillez consulterPrésentation d'AWS X-Raydans leAWS X-RayManuel du développeur.



Vous pouvez également explorerAWS X-Raygraphiques de la façon dont vos services se connectent les uns avec les autres. Vous pouvez également les utiliser pour explorer des statistiques agrégées sur les performances de chaque liaison service-service. Enfin, vous pouvez approfondir toute transaction spécifique pour voir comment les segments représentant les appels réseau sont associés à cette transaction particulière.




Vous pouvez utiliser ces fonctionnalités pour identifier s'il existe un goulot d'étranglement réseau ou si un service spécifique au sein de votre réseau ne fonctionne pas comme prévu.

## Journaux de flux VPC

Vous pouvez utiliser les journaux de flux Amazon VPC pour analyser les performances réseau et les problèmes de connectivité de débogage. Lorsque les journaux de flux VPC sont activés, vous pouvez capturer un journal de toutes les connexions dans votre VPC. Il s'agit notamment de connexions aux interfaces réseau associées à Elastic Load Balancing, Amazon RDS, NAT passerelles et d'autresAWSque vous utilisez peut-être. Pour plus d'informations, consultez Journaux de flux VPC dans le Amazon VPC Guide de l'utilisateur.

# Conseils sur le réglage réseau

Il y a quelques paramètres que vous pouvez affiner afin d'améliorer votre réseau.

## nofile ulimit

Si vous pensez que votre application a un trafic élevé et qu'elle gère de nombreuses connexions simultanées, vous devez tenir compte du quota système pour le nombre de fichiers autorisés. Lorsqu'il y a beaucoup de sockets réseau ouvertes, chacune doit être représentée par un descripteur de fichier. Si votre quota de descripteur de fichier est trop faible, il limitera vos sockets réseau. Cela entraîne des échecs de connexions ou des erreurs. Vous pouvez mettre à jour le quota spécifique du conteneur pour le nombre de fichiers dans la définition de tâche Amazon ECS. Si vous utilisez Amazon EC2 (au lieu deAWS Fargate), vous devrez peut-être également ajuster ces quotas sur votre instance Amazon EC2 sous-jacente.

## net sysctl

Une autre catégorie de paramètres réglables est le paramètre`sysctl`Paramètres réseau. Vous devez vous référer aux paramètres spécifiques de votre distribution Linux de votre choix. Beaucoup de ces paramètres ajustent la taille des tampons de lecture et d'écriture. Cela peut être utile dans certaines situations lorsque vous exécutez des instances Amazon EC2 à grande échelle qui ont beaucoup de conteneurs sur elles.

# Meilleures pratiques - Mise à l'échelle automatique et gestion de la capacité

Amazon ECS est utilisé pour exécuter des charges de travail d'applications conteneurisées de toutes tailles. Cela inclut à la fois les environnements de test minimaux et les environnements de production importants fonctionnant à l'échelle mondiale.

Avec Amazon ECS, comme tousAWSServices, vous ne payez qu'en fonction de votre consommation. Lorsqu'elle est conçue de manière appropriée, vous pouvez économiser des coûts en demandant à votre application de consommer uniquement les ressources dont elle a besoin au moment où elle en a besoin. Ce guide des meilleures pratiques montre comment exécuter vos charges de travail Amazon ECS d'une manière qui répond à vos objectifs de niveau de service tout en continuant à fonctionner de manière rentable.

Rubriques



## Détermination de la taille de

L'un des choix les plus importants à faire lors du déploiement de conteneurs sur Amazon ECS est la taille de vos conteneurs et de vos tâches. La taille de vos conteneurs et de vos tâches est essentielle à la mise à l'échelle et à la planification de la capacité. Dans Amazon ECS, deux mesures de ressources sont utilisées pour la capacité : UC et mémoire. Le processeur est mesuré en unités de 1/1024 d'un vCPU complet (où 1024 unités est égal à 1 vCPU entier). La mémoire est mesurée en mégaoctets. Dans votre définition de tâche, vous pouvez déclarer les réservations et les limites de ressources.

Lorsque vous déclarez une réservation, vous déclarez le minimum de ressources qu'une tâche nécessite. Votre tâche reçoit au moins la quantité de ressources demandée. Votre application

peut être en mesure d'utiliser plus de CPU ou de mémoire que la réservation que vous déclarez. Cependant, cela est soumis à toutes les limites que vous avez également déclarées. L'utilisation d'un montant supérieur au montant de la réservation est connue sous le nom d'éclatement. Dans Amazon ECS, les réservations sont garanties. Par exemple, si vous utilisez des instances Amazon EC2 pour fournir de la capacité, Amazon ECS ne place pas de tâche sur une instance où la réservation ne peut pas être effectuée.

Une limite est la quantité maximale d'unités CPU ou de mémoire que votre conteneur ou tâche peut utiliser. Toute tentative d'utiliser plus de CPU que cette imit entraîne une limitation. Toute tentative d'utilisation de plus de mémoire entraîne l'arrêt de votre conteneur.

Le choix de ces valeurs peut être difficile. En effet, les valeurs les plus adaptées à votre application dépendent grandement des besoins en ressources de votre application. Le test de charge de votre application est la clé d'une planification réussie des besoins en ressources et d'une meilleure compréhension des exigences de votre application.

## Applications sans état

Pour les applications sans état qui s'adaptent horizontalement, comme une application derrière un équilibreur de charge, nous vous recommandons de déterminer d'abord la quantité de mémoire que votre application consomme lorsqu'elle sert des requêtes. Pour ce faire, vous pouvez utiliser des outils traditionnels tels que`psoutopou` des solutions de surveillance telles que CloudWatch Container Insights.

Lorsque vous déterminez une réservation d'UC, pensez à la manière dont vous souhaitez mettre à l'échelle votre application en fonction des besoins de votre entreprise. Vous pouvez utiliser des réservations de CPU plus petites, telles que 256 unités de CPU (ou 1/4 de vCPU), pour effectuer une mise à l'échelle plus fine qui minimise les coûts. Mais ils pourraient ne pas évoluer assez rapidement pour répondre à des pics importants de la demande. Vous pouvez utiliser des réservations de CPU plus importantes pour évoluer plus rapidement et ainsi faire correspondre plus rapidement les pics de demande. Cependant, les réservations de CPU plus importantes sont plus coûteuses.

## Autres applications

Pour les applications qui ne sont pas mises à l'échelle horizontale, telles que les travailleurs singleton ou les serveurs de base de données, la capacité et le coût disponibles représentent vos considérations les plus importantes. Vous devez choisir la quantité de mémoire et de CPU en fonction des tests de charge qui indiquent que vous devez servir le trafic pour atteindre votre objectif

de niveau de service. Amazon ECS veille à ce que l'application soit placée sur un hôte disposant d'une capacité suffisante.

# Configuration de la mise à l'échelle automatique du service

Un service Amazon ECS est un ensemble géré de tâches. Chaque service a une définition de tâche associée, un nombre de tâches souhaité et une stratégie de placement facultative. La mise à l'échelle automatique du service Amazon ECS est implémentée via le service Application Auto Scaling. Application Auto Scaling utilise les mesures CloudWatch comme source de mesures de mise à l'échelle. Il utilise également des alarmes CloudWatch pour définir des seuils de mise à l'échelle ou de sortie de votre service. Vous indiquez les seuils de mise à l'échelle, soit en définissant une cible de mesure, appeléemise à l'échelle de suivi cible, ou en spécifiant des seuils, appelésmise à l'échelle par étapes. Une fois l'application Auto Scaling configuré, il calcule en permanence le nombre de tâches souhaité approprié pour le service. Il informe également Amazon ECS lorsque le nombre de tâches souhaité doit changer, soit en le mettant à l'échelle, soit en le mettant à l'échelle.

Pour utiliser efficacement la mise à l'échelle automatique du service, vous devez choisir une mesure de mise à l'échelle appropriée. Dans les sections suivantes, nous expliquons comment choisir une mesure.

## Caractérisation de votre application

Une mise à l'échelle correcte d'une application nécessite de connaître les conditions dans lesquelles l'application doit être mise à l'échelle et quand elle doit être mise à l'échelle. Essentiellement, une demande devrait être réduite si l'on prévoit que la demande dépasse la capacité. Inversement, une application peut être mise à l'échelle pour économiser les coûts lorsque les ressources dépassent la demande.

### Identification d'une mesure d'utilisation

Pour mettre à l'échelle efficacement, il est essentiel d'identifier une mesure indiquant l'utilisation ou la saturation. Cette mesure doit présenter les propriétés suivantes pour être utile pour la mise à l'échelle.

- La mesure doit être corrélée avec la demande. Lorsque les ressources sont maintenues stables, mais que la demande change, la valeur de mesure doit également changer. La mesure devrait augmenter ou diminuer lorsque la demande augmente ou diminue.
- La valeur de mesure doit être mise à l'échelle proportionnelle à la capacité. Lorsque la demande reste constante, l'ajout de ressources supplémentaires doit entraîner une modification

proportionnelle de la valeur de mesure. Par conséquent, le doublement du nombre de tâches devrait entraîner une diminution de 50 % de la mesure.

La meilleure façon d'identifier une mesure d'utilisation consiste à tester la charge dans un environnement de préproduction tel qu'un environnement intermédiaire. Des solutions de test de charge commerciales et open source sont largement disponibles. Ces solutions peuvent généralement générer une charge synthétique ou simuler un trafic utilisateur réel.

Pour démarrer le processus de test de charge, vous devez commencer par créer des tableaux de bord pour les mesures d'utilisation de votre application. Ces mesures comprennent l'utilisation de l'UC, l'utilisation de la mémoire, les opérations d'E/S, la profondeur de la file d'attente d'E/S et le débit réseau. Vous pouvez collecter ces mesures avec un service tel que CloudWatch Container Insights. Vous pouvez également le faire en utilisant Amazon Managed Service for Prometheus avec Amazon Managed Service for Grafana. Au cours de ce processus, assurez-vous de collecter et de tracer des mesures pour les temps de réponse de votre application ou les taux d'achèvement des travaux.

Lors du test de charge, commencez par une petite demande ou un taux d'insertion de travail. Maintenez cette vitesse stable pendant plusieurs minutes pour permettre à votre application de se réchauffer. Ensuite, augmentez lentement le taux et maintenez-le stable pendant quelques minutes. Répétez ce cycle en augmentant le taux à chaque fois jusqu'à ce que les temps de réponse ou de traitement de votre application soient trop lents pour atteindre vos objectifs de niveau de service (LOS).

Lors du test de charge, examinez chacune des mesures d'utilisation. Les mesures qui augmentent en même temps que la charge sont les meilleurs candidats à être vos meilleures mesures d'utilisation.

Ensuite, identifiez la ressource qui atteint la saturation. En même temps, examinez également les mesures d'utilisation pour voir lequel s'aplatit en premier à un niveau élevé. Ou, examinez lequel atteint le pic, puis bloque votre application en premier. Par exemple, si l'utilisation du processeur augmente de 0 % à 70 -80 % à mesure que vous ajoutez de la charge, puis reste à ce lebel après avoir ajouté encore plus de charge, alors il est sûr de dire que le CPU est saturé. Selon l'architecture du processeur, il pourrait ne jamais atteindre 100 %. Par exemple, supposons que l'utilisation de la mémoire augmente à mesure que vous ajoutez de la charge, puis votre application se bloque soudainement lorsqu'elle atteint la limite de mémoire de la tâche ou de l'instance Amazon EC2. Dans cette situation, il est probable que la mémoire a été entièrement consommée. Plusieurs ressources peuvent être consommées par votre application. Par conséquent, choisissez la mesure qui représente la ressource qui est supprimée en premier.

Enfin, réessayez le test de chargement après avoir doublé le nombre de tâches ou d'instances Amazon EC2. Supposons que la mesure clé augmente, ou diminue, à la moitié du taux qu'auparavant. Si tel est le cas, la mesure est proportionnelle à la capacité. Il s'agit d'une bonne mesure d'utilisation pour la mise à l'échelle automatique.

Considérons maintenant ce scénario hypothétique. Supposons que vous chargez testez une application et que l'utilisation du processeur atteigne finalement 80 % à 100 demandes par seconde. Lorsque plus de charge est ajoutée, cela ne fait plus augmenter l'utilisation du processeur. Cependant, cela fait que votre application répond plus lentement. Ensuite, vous exécutez à nouveau le test de charge, en doublant le nombre de tâches, mais en maintenant le taux à sa valeur de crête précédente. Si vous constatez que l'utilisation moyenne de l'UC tombe à environ 40 %, l'utilisation moyenne de l'UC est un bon candidat pour une mesure de mise à l'échelle. D'un autre côté, si l'utilisation du processeur reste à 80 % après avoir augmenté le nombre de tâches, l'utilisation moyenne du processeur n'est pas une bonne mesure de mise à l'échelle. Dans ce cas, des recherches plus poussées sont nécessaires pour trouver une mesure appropriée.

## Modèles d'application courants et propriétés de mise à l'échelle

Les logiciels de toutes sortes sont exécutés surAWS. De nombreuses charges de travail sont locales, tandis que d'autres sont basées sur des logiciels open source populaires. Peu importe leur origine, nous avons observé des modèles de conception courants pour les services. Comment mettre à l'échelle efficacement dépend en grande partie du motif.

### Le serveur dédié à l'UC efficace

Le serveur CPU efficace n'utilise presque aucune ressource autre que le débit du processeur et du réseau. Chaque demande peut être traitée par l'application seule. Les demandes ne dépendent pas d'autres services tels que les bases de données. L'application peut gérer des centaines de milliers de demandes simultanées et peut utiliser efficacement plusieurs processeurs pour le faire. Chaque requête est soit traitée par un thread dédié avec une surcharge mémoire faible, soit il y a une boucle d'événement asynchrone qui s'exécute sur chaque CPU qui traite les demandes. Chaque réplica de l'application est également capable de traiter une requête. La seule ressource qui pourrait être épuisée avant le CPU est la bande passante réseau. Dans les services de liaison CPU, l'utilisation de la mémoire, même au débit maximal, représente une fraction des ressources disponibles.

Ce type d'application convient à la mise à l'échelle automatique basée sur le processeur. L'application bénéficie d'une flexibilité maximale en termes de mise à l'échelle. Il peut être mis à l'échelle verticale en lui fournissant des instances Amazon EC2 plus grandes ou des vCPUs Fargate. Et, il peut également être mis à l'échelle horizontalement en ajoutant plus de réplicas. L'ajout de

réplicas ou le doublement de la taille de l'instance réduit de moitié l'utilisation moyenne du processeur par rapport à la capacité.

Si vous utilisez la capacité Amazon EC2 pour cette application, envisagez de la placer sur des instances optimisées pour le calcul, telles que lec5ouc6gfamille.

Le serveur dédié à la mémoire efficace

Le serveur de mémoire efficace alloue une quantité importante de mémoire par requête. À la concurrence maximale, mais pas nécessairement au débit, la mémoire est épuisée avant que les ressources du processeur ne soient épuisées. La mémoire associée à une requête est libérée à la fin de la requête. Les demandes supplémentaires peuvent être acceptées tant qu'il y a de la mémoire disponible.

Ce type d'application convient à la mise à l'échelle automatique basée sur la mémoire. L'application bénéficie d'une flexibilité maximale en termes de mise à l'échelle. Il peut être mis à l'échelle à la fois verticalement en lui fournissant des ressources mémoire Amazon EC2 ou Fargate plus importantes. Et, il peut également être mis à l'échelle horizontalement en ajoutant plus de réplicas. L'ajout de réplicas ou le doublement de la taille de l'instance peut réduire de moitié l'utilisation moyenne de la mémoire par rapport à la capacité.

Si vous utilisez la capacité Amazon EC2 pour cette application, envisagez de la placer sur des instances optimisées pour la mémoire, telles quer5our6gfamille.

Certaines applications liées à la mémoire ne libèrent pas la mémoire associée à une requête lorsqu'elle se termine, de sorte qu'une réduction de la concurrence n'entraîne pas une réduction de la mémoire utilisée. Pour cela, nous vous déconseillons d'utiliser la mise à l'échelle basée sur la mémoire.

Le serveur basé sur le travail

Le serveur basé sur le travail traite une demande pour chaque thread de travail individuel l'une après l'autre. Les threads de travail peuvent être des threads légers, tels que des threads POSIX. Ils peuvent également être des threads plus lourds, tels que des processus UNIX. Peu importe le thread qu'ils sont, il y a toujours une concurrence maximale que l'application peut prendre en charge. Habituellement, la limite de concurrence est définie proportionnellement aux ressources de mémoire disponibles. Si la limite de concurrence est atteinte, des demandes supplémentaires sont placées dans une file d'attente en souffrance. En cas de dépassement de la file d'attente des arriérés, les demandes entrantes supplémentaires sont immédiatement rejetées. Les applications courantes qui correspondent à ce modèle incluent le serveur Web Apache et Gunicorn.

La concurrence de requête est généralement la meilleure mesure pour mettre à l'échelle cette application. Étant donné qu'il existe une limite de concurrence pour chaque réplica, il est important d'effectuer une mise à l'échelle avant que la limite moyenne ne soit atteinte.

La meilleure façon d'obtenir des mesures de concurrence de requête est de faire en sorte que votre application les signale à CloudWatch. Chaque réplica de votre application peut publier le nombre de demandes simultanées sous forme de mesure personnalisée à une fréquence élevée. Nous recommandons que la fréquence soit réglée pour être au moins une fois par minute. Une fois plusieurs rapports collectés, vous pouvez utiliser la concurrence moyenne comme mesure de mise à l'échelle. Cette mesure est calculée en prenant la concurrence totale et en la divisant par le nombre de réplicas. Par exemple, si la concurrence totale est de 1000 et que le nombre de réplicas est de 10, la concurrence moyenne est de 100.

Si votre application se trouve derrière un Application Load Balancer, vous pouvez également utiliser le`ActiveConnectionCount`pour l'équilibreur de charge en tant que facteur dans la mesure de mise à l'échelle. La .`ActiveConnectionCount`doit être divisée par le nombre de réplicas pour obtenir une valeur moyenne. La valeur moyenne doit être utilisée pour la mise à l'échelle, par opposition à la valeur de comptage brute.

Pour que cette conception fonctionne le mieux, l'écart type de latence de réponse doit être faible à de faibles taux de demande. Nous recommandons que, pendant les périodes de faible demande, la plupart des demandes soient traitées dans un court laps de temps, et il n'y a pas beaucoup de demandes qui prennent beaucoup plus de temps que la moyenne pour répondre. Le temps de réponse moyen devrait être proche du temps de réponse du 95e centile. Sinon, des dépassements de file d'attente peuvent se produire en conséquence. Cela conduit à des erreurs. Nous vous recommandons de fournir des réplicas supplémentaires si nécessaire pour atténuer le risque de débordement.

Le serveur en attente

Le serveur en attente effectue un traitement pour chaque requête, mais il dépend fortement d'un ou de plusieurs services en aval pour fonctionner. Les applications de conteneur font souvent un usage intensif des services en aval tels que les bases de données et d'autres services d'API. Cela peut prendre un certain temps pour que ces services répondent, en particulier dans des scénarios de grande capacité ou de concurrence élevée. C'est parce que ces applications ont tendance à utiliser peu de ressources CPU et leur concurrence maximale en termes de mémoire disponible.

Le service d'attente convient soit dans le modèle de serveur lié à la mémoire, soit dans le modèle de serveur basé sur le travail, selon la façon dont l'application est conçue. Si la concurrence de

l'application est limitée uniquement par la mémoire, l'utilisation moyenne de la mémoire doit être utilisée comme mesure de mise à l'échelle. Si la concurrence de l'application est basée sur une limite de travail, la concurrence moyenne doit être utilisée comme mesure de mise à l'échelle.

Serveur basé sur Java

Si votre serveur Java est lié au processeur et évolue proportionnellement aux ressources du processeur, il peut être adapté au modèle de serveur lié au processeur efficace. Si tel est le cas, l'utilisation moyenne de l'UC peut être appropriée en tant que mesure de mise à l'échelle. Cependant, de nombreuses applications Java ne sont pas liées au processeur, ce qui les rend difficiles à mettre à l'échelle.

Pour que vous obteniez les meilleures performances, nous vous recommandons d'allouer autant de mémoire que possible au tas Java Virtual Machine (JVM). Les versions récentes de la JVM, y compris la mise à jour de Java 8 191 ou ultérieure, définissent automatiquement la taille du tas aussi grande que possible pour tenir dans le conteneur. Cela signifie que, en Java, l'utilisation de la mémoire est rarement proportionnelle à l'utilisation de l'application. À mesure que le taux de demande et la concurrence augmentent, l'utilisation de la mémoire reste constante. Pour cette raison, nous ne recommandons pas de mettre à l'échelle les serveurs Java en fonction de l'utilisation de la mémoire. Au lieu de cela, nous recommandons généralement une mise à l'échelle sur l'utilisation du processeur.

Dans certains cas, les serveurs basés sur Java rencontrent l'épuisement du tas avant d'épuiser le processeur. Si votre application est sujette à l'épuisement du tas à une concurrence élevée, les connexions moyennes sont la meilleure mesure de mise à l'échelle. Si votre application est sujette à l'épuisement du tas à haut débit, le taux de demande moyen est la meilleure mesure de mise à l'échelle.

Serveurs qui utilisent d'autres runtimes collectés

De nombreuses applications serveur sont basées sur des runtimes qui effectuent la collecte des ordures telles que .NET et Ruby. Ces applications serveur peuvent s'intégrer dans l'un des modèles décrits plus haut. Cependant, comme pour Java, nous ne recommandons pas de mettre à l'échelle ces applications en fonction de la mémoire, car leur utilisation moyenne de la mémoire observée n'est souvent pas corrélée avec le débit ou la concurrence.

Pour ces applications, nous vous recommandons de mettre à l'échelle l'utilisation du processeur si l'application est liée au processeur. Sinon, nous vous recommandons de mettre à l'échelle en fonction du débit moyen ou de la concurrence moyenne, en fonction de vos résultats de test de charge.

ProJob d'

De nombreuses charges de travail impliquent un traitement asynchrone des tâches. Ils incluent des applications qui ne reçoivent pas de demandes en temps réel, mais qui s'abonnent à une file d'attente de travail pour recevoir des tâches. Pour ces types d'applications, la mesure de mise à l'échelle appropriée est presque toujours la profondeur de la file d'attente. La croissance de la file d'attente indique que le travail en attente dépasse la capacité de traitement, alors qu'une file d'attente vide indique qu'il y a plus de capacité que de travail à faire.

AWS, tels qu'Amazon SQS et Amazon Kinesis Data Streams, fournissent des mesures CloudWatch qui peuvent être utilisées pour la mise à l'échelle. Pour Amazon SQS,`ApproximateNumberOfMessagesVisible`est la meilleure métrique. Pour les flux de Kinesis Data Streams, envisagez d'utiliser le`MillisBehindLatest`, publiée par la bibliothèque Client Kinesis brary (KCL). Cette mesure doit être calculée en moyenne pour tous les consommateurs avant de l'utiliser pour la mise à l'échelle.

# Capacité et disponibilité

La disponibilité des applications est essentielle pour garantir une expérience sans erreur et réduire la latence des applications. La disponibilité dépend de la disponibilité des ressources qui sont accessibles et qui ont une capacité suffisante pour répondre à la demande.AWSfournit plusieurs mécanismes pour gérer la disponibilité. Pour les applications hébergées sur Amazon ECS, il s'agit de la mise à l'échelle automatique et des zones de disponibilité (AZS). La mise à l'échelle automatique gère le nombre de tâches ou d'instances en fonction des mesures que vous définissez, tandis que les zones de disponibilité vous permettent d'héberger votre application dans des emplacements isolés mais géographiquement proches.

Comme pour la taille des tâches, la capacité et la disponibilité présentent certains compromis que vous devez considérer. Idéalement, la capacité serait parfaitement alignée sur la demande. Il y aurait toujours assez de capacité pour répondre aux demandes et traiter les tâches afin d'atteindre les objectifs de niveau de service (LOS), y compris un faible taux de latence et de taux d'erreur. La capacité ne serait jamais trop élevée, entraînant des coûts excessifs ; elle ne serait jamais trop faible, entraînant des taux de latence et d'erreur élevés.

La mise à l'échelle automatique est un processus latent. Tout d'abord, les mesures en temps réel doivent être livrées à CloudWatch. Ensuite, ils doivent être agrégés pour l'analyse, ce qui peut prendre jusqu'à plusieurs minutes en fonction de la granularité de la métrique. CloudWatch compare les mesures aux seuils d'alarme pour identifier une pénurie ou un excès de ressources. Pour éviter

l'instabilité, configurez les alarmes de manière à ce que le seuil défini soit franchi pendant quelques minutes avant que l'alarme ne s'éteigne. Il faut également du temps pour provisionner de nouvelles tâches et pour mettre fin à des tâches qui ne sont plus nécessaires.

En raison de ces retards potentiels dans le système décrit, il est important de conserver une certaine marge de manœuvre en surprovisionnant. Cela peut aider à faire face à des rafales de demande à court terme. Cela permet également à votre application de traiter des demandes supplémentaires sans atteindre la saturation. Comme bonne pratique, vous pouvez définir votre cible de mise à l'échelle entre 60 et 80 % de l'utilisation. Cela permet à votre application de mieux gérer les rafales de demande supplémentaire alors que la capacité supplémentaire est encore en cours d'approvisionnement.

Une autre raison pour laquelle nous vous recommandons de surprovisionner est de pouvoir répondre rapidement aux défaillances de la zone de disponibilité.AWSrecommande que les charges de travail de production soient servies à partir de plusieurs zones de disponibilité. En effet, si une défaillance de la zone de disponibilité se produit, vos tâches qui s'exécutent dans les zones de disponibilité restantes peuvent toujours répondre à la demande. Si votre application s'exécute dans deux zones de disponibilité, vous devez doubler votre nombre de tâches normal. Cela vous permet de fournir une capacité immédiate en cas de panne potentielle. Si votre application s'exécute dans trois zones de disponibilité, nous vous recommandons d'exécuter 1,5 fois le nombre de tâches normal. Autrement dit, exécutez trois tâches pour deux qui sont nécessaires pour servir ordinaire.

## Optimisation de la vitesse de dimensionnement

La mise à l'échelle automatique est un processus réactif qui prend du temps pour prendre effet. Cependant, il existe des moyens de réduire le temps nécessaire à la mise à l'échelle.

Réduire la taille de l'image Les images plus volumineuses prennent plus de temps à télécharger depuis un référentiel d'images et à décompresser Par conséquent, garder des tailles d'image plus petites réduit le temps nécessaire au démarrage d'un conteneur. Pour réduire la taille de l'image, vous pouvez suivre les recommandations suivantes :

- Si vous pouvez construire un binaire statique ou utiliser Golang, construisez votre image`FROM`gratter et inclure uniquement votre application binaire dans l'image résultante.
- Utilisez des images de base minimisées provenant de fournisseurs de distribution en amont, tels qu'Amazon Linux ou Ubuntu.
- N'incluez aucun artefact de construction dans votre image finale. L'utilisation de builds en plusieurs étapes peut aider avec cela.

- CompactRUNchaque fois que cela est possible. EACHRUNcrée un nouveau calque d'image, ce qui entraîne un aller-retour supplémentaire pour télécharger le calque. Un seulRUNqui a plusieurs commandes jointes par&&a moins de couches qu'une avec plusieursRUNStades.
- Si vous souhaitez inclure des données, telles que des données d'inférence ML, dans votre image finale, n'incluez que les données nécessaires au démarrage et au démarrage du trafic. Si vous récupérez des données à la demande à partir d'Amazon S3 ou d'un autre stockage sans affecter le service, stockez vos données à ces endroits à la place.

Gardez vos images à proximité. Plus la latence du réseau est élevée, plus le téléchargement de l'image prend de temps. Hébergez vos images dans un référentiel dans le mêmeAWSRégion dans laquelle se trouve votre charge de travail. Amazon ECR est un référentiel d'images hautes performances disponible dans toutes les régions où Amazon ECS est disponible. Évitez de parcourir Internet ou un lien VPN pour télécharger des images de conteneur. L'hébergement de vos images dans la même Région améliore la fiabilité globale. Il atténue le risque de problèmes de connectivité réseau et de disponibilité dans une autre région. Vous pouvez également implémenter la réplication entre régions Amazon ECR pour vous aider dans ce domaine.

Réduire les seuils de vérification de l'état de l'équilibreur Les équilibreurs de charge effectuent des vérifications de l'état avant d'envoyer du trafic à votre application. La configuration par défaut du contrôle d'intégrité d'un groupe cible peut prendre 90 secondes ou plus. Pendant ce temps, il vérifie l'état de santé et la réception des demandes. L'abaissement de l'intervalle de vérification de l'état et du nombre de seuils permet à votre application d'accepter le trafic plus rapidement et de réduire la charge sur d'autres tâches.

Considérez les performances de démarrage à froid. Certaines applications utilisent des runtimes tels que Java effectuer la compilation JIT (Just-In-Time). Le processus de compilation au moins au démarrage peut montrer les performances de l'application. Une solution consiste à réécrire les parties critiques en matière de latence de votre charge de travail dans des langues qui n'imposent pas de pénalité de performances de démarrage à froid.

Utilisez les stratégies de dimensionnement par étapes et non pas les stratégies de dimensionnement avec suivi de la cible. Vous disposez de plusieurs options Application Auto Scaling pour les tâches Amazon ECS. Le suivi des cibles est le mode le plus simple à utiliser. Avec elle, tout ce que vous devez faire est de définir une valeur cible pour une mesure, telle que l'utilisation moyenne du processeur. Ensuite, le scaler automatique gère automatiquement le nombre de tâches nécessaires pour atteindre cette valeur. Cependant, nous vous recommandons d'utiliser la mise à l'échelle des étapes à la place afin que vous puissiez réagir plus rapidement aux changements de la demande.

Avec la mise à l'échelle des étapes, vous définissez les seuils spécifiques pour vos mesures de mise à l'échelle et le nombre de tâches à ajouter ou à supprimer lorsque les seuils sont franchis. Et, plus important encore, vous pouvez réagir très rapidement aux changements de la demande en minimisant le temps qu'une alarme seuil est en violation. Pour de plus amples informations, veuillez consulterAuto Scaling du servicedans leGuide du développeur Amazon Elastic Container Service.

Si vous utilisez des instances Amazon EC2 pour fournir une capacité de cluster, tenez compte des recommandations suivantes :

Utilisez des instances Amazon EC2 plus volumineuses et des volumes Amazon EBS plus rapides. Vous pouvez améliorer les vitesses de téléchargement et de préparation des images en utilisant une instance Amazon EC2 plus grande et un volume Amazon EBS plus rapide. Dans une famille d'instances Amazon EC2 donnée, le débit maximal du réseau et Amazon EBS augmente à mesure que la taille de l'instance augmente (par exemple, de`m5.xlarge`sur`m5.2xlarge`). En outre, vous pouvez également personnaliser les volumes Amazon EBS pour augmenter leur débit et leurs E/S par seconde. Par exemple, si vous utilisez`gp2`, utilisez des volumes plus importants qui offrent un débit de référence plus élevé. Si vous utilisez`gp3`, spécifiez le débit et les opérations d'E/S par seconde lorsque vous créez le volume.

Utilisez le mode réseau de pont pour les tâches exécutées sur les instances Amazon EC2. Tâches utilisant`bridge`sur Amazon EC2 démarre plus rapidement que les tâches qui utilisent le`awsvpc`Mode réseau. Quand`awsvpc`Mode réseau est utilisé, Amazon ECS attache une elastic network interface (ENI) à l'instance avant de lancer la tâche. Cela introduit une latence supplémentaire. Il y a cependant plusieurs compromis pour l'utilisation de la mise en réseau de ponts. Ces tâches n'ont pas leur propre groupe de sécurité, et il y a des implications pour l'équilibrage de charge. Pour de plus amples informations, veuillez consulterGroupes cibles de l'équilibreur de chargedans leGuide de l'utilisateur Elastic Load Balancing.

## Traitement des chocs de la demande

Certaines applications subissent des chocs importants soudains en demande. Cela se produit pour diverses raisons : un événement d'actualité, une grande vente, un événement médiatique ou tout autre événement qui devient viral et provoque une augmentation rapide et significative du trafic en très peu de temps. Si elle n'est pas planifiée, la demande peut dépasser rapidement les ressources disponibles.

La meilleure façon de gérer les chocs de la demande est de les anticiper et de les planifier en conséquence. Étant donné que la mise à l'échelle automatique peut prendre du temps, nous

vous recommandons de mettre à l'échelle votre application avant que le choc de la demande ne commence. Pour obtenir les meilleurs résultats, nous vous recommandons d'avoir un plan d'affaires qui implique une collaboration étroite entre les équipes qui utilisent un calendrier partagé. L'équipe qui planifie l'événement doit travailler en étroite collaboration avec l'équipe responsable de l'application à l'avance. Cela donne à cette équipe assez de temps pour avoir un plan de planification clair. Ils peuvent programmer la capacité afin qu'elle évolue avant l'événement et qu'elle soit mise à l'échelle après l'événement. Pour de plus amples informations, veuillez consulterMise à l'échelle planifiéedans leGuide de l'utilisateur Application Auto Scaling.

Si vous disposez d'un plan d'Support entreprises, veillez également à travailler avec votre responsable de compte technique (TAM). Votre TAM peut vérifier vos quotas de service et s'assurer que tous les quotas nécessaires sont levés avant le début de l'événement. De cette façon, vous ne touchez pas accidentellement les quotas de service. Ils peuvent également vous aider en préchauffant des services tels que des équilibreurs de charge pour vous assurer que votre événement se déroule sans heurt.

Le traitement des chocs de demande imprévus est un problème plus difficile. Les chocs imprévus, s'ils sont d'une amplitude suffisante, peuvent rapidement faire en sorte que la demande dépasse la capacité. Il peut également dépasser la capacité d'auto-scaling à réagir. La meilleure façon de se préparer aux chocs imprévus est de surapprovisionner les ressources. Vous devez disposer de suffisamment de ressources pour gérer la demande de trafic maximale prévue à tout moment.

Le maintien d'une capacité maximale en prévision des chocs imprévus de la demande peut s'avérer coûteux. Pour atténuer l'impact sur les coûts, trouvez une mesure ou un événement d'indicateur avancé qui prédit un choc important de la demande est imminent. Si la mesure ou l'événement fournit un préavis significatif de manière fiable, commencez le processus de mise à l'échelle immédiatement lorsque l'événement se produit ou lorsque la mesure franchit le seuil spécifique que vous avez défini.

Si votre application est sujette à des chocs soudains imprévus de la demande, envisagez d'ajouter à votre application un mode hautes performances qui sacrifie les fonctionnalités non critiques tout en conservant des fonctionnalités cruciales pour un client. Par exemple, supposons que votre application peut passer de la génération de réponses personnalisées coûteuses à la diffusion d'une page de réponse statique. Dans ce scénario, vous pouvez augmenter le débit de manière significative sans aucune mise à l'échelle de l'application.

Enfin, vous pouvez envisager de briser les services monolithiques pour mieux faire face aux chocs de la demande. Si votre application est un service monolithique coûteux à exécuter et lent à évoluer, vous pourriez être en mesure d'extraire ou de réécrire des éléments critiques pour les

performances et de les exécuter en tant que services distincts. Ces nouveaux services peuvent alors être mis à l'échelle indépendamment des composants moins critiques. Le fait de disposer de la flexibilité nécessaire pour mettre à l'échelle les fonctionnalités critiques en matière de performances séparément des autres parties de votre application peut réduire le temps nécessaire à l'ajout de capacité et contribuer à économiser les coûts.

# Capacité de cluster

Plus tôt dans cette rubrique, nous avons expliqué comment mettre à l'échelle le compte de réplica pour votre à l'aide de mesures de mise à l'échelle. Vos tâches doivent également s'exécuter sur des ressources, y compris des ressources CPU et mémoire. Cela revient à nouveau sur le thème de la capacité. Dans Amazon ECS, la capacité est fournie par deux fournisseurs principaux :AWS Fargateet Amazon EC2

Vous pouvez fournir de la capacité à un cluster Amazon ECS de plusieurs façons. Par exemple, vous pouvez lancer des instances Amazon EC2 et les enregistrer auprès du cluster au démarrage à l'aide de l'agent de conteneur Amazon ECS. Cependant, cette méthode peut être difficile car vous devez gérer la mise à l'échelle par vous-même. Par conséquent, nous vous recommandons d'utiliser les fournisseurs de capacité Amazon ECS. Ils gèrent la mise à l'échelle des ressources pour vous. Il existe trois types de fournisseurs de capacité : Amazon EC2, Fargate et Fargate Spot.

Les fournisseurs de capacité Fargate et Fargate Spot gèrent le cycle de vie des tâches Fargate pour vous. Fargate fournit une capacité à la demande, et Fargate Spot fournit une capacité Spot. Lorsqu'une tâche est lancée, ECS met en service une ressource Fargate pour vous. Cette ressource Fargate est fournie avec les unités de mémoire et de CPU qui correspondent directement aux limites de niveau de tâche que vous avez déclarées dans votre définition de tâche. Chaque tâche reçoit sa propre ressource Fargate, créant une relation 1:1 entre la tâche et les ressources de calcul.

Les tâches exécutées sur Fargate Spot sont sujettes à interruption. Les interruptions interviennent après un avertissement de deux minutes. Celles-ci se produisent pendant les périodes de forte demande. Fargate Spot fonctionne le mieux pour les charges de travail tolérantes aux interruptions telles que les travaux par lots, les environnements de développement ou de mise en scène. Ils conviennent également à tout autre scénario où la haute disponibilité et la faible latence ne sont pas une exigence.

Vous pouvez exécuter des tâches Fargate Spot en même temps que des tâches à la demande Fargate. En les utilisant ensemble, vous recevez une capacité de « rafale » de provision à un coût moindre.

ECS peut également gérer la capacité d'instance Amazon EC2 pour vos tâches. Chaque fournisseur de capacité Amazon EC2 est associé à un groupe Amazon EC2 Auto Scaling que vous spécifiez. Lorsque vous utilisez le fournisseur de capacité Amazon EC2, la mise à l'échelle automatique du cluster ECS conserve la taille du groupe Amazon EC2 Auto Scaling afin de garantir que toutes les tâches planifiées peuvent être placées.

## Bonnes pratiques en matière de capacité

Ajoutez une marge de manœuvre à votre service et non au fournisseur de capacité. Les fournisseurs de capacité Amazon EC2 offrent une valeur de capacité cible. Si vous définissez la valeur inférieure à 100 %, ECS fournit plus d'instances Amazon EC2 que nécessaire pour répondre à vos tâches. Il peut être utile d'avoir plusieurs instances Amazon EC2 prêtes à accepter des tâches. Toutefois, lorsque vous utilisez Amazon Virtual Private Cloud, le lancement de nouvelles tâches nécessite plus de temps pour télécharger l'image et joindre une interface réseau. Cette latence ajoutée pourrait nuire à votre résultat net.

Par conséquent, nous vous recommandons d'effectuer les opérations suivantes. Au lieu de réduire la capacité cible de votre fournisseur de capacité, augmentez le nombre de réplicas dans votre service en modifiant la mesure de mise à l'échelle du suivi cible ou les seuils de mise à l'échelle des étapes du service. Pour plus d'informations sur les stratégies de dimensionnement associées, consultez.Stratégies de dimensionnement Suivi de la cibleouStratégies de mise à l'échelle d'étapedans leGuide du développeur Amazon Elastic Container Service. Le fournisseur de capacité Amazon EC2 fournit la capacité nécessaire pour des tâches supplémentaires en ajoutant des instances supplémentaires au groupe Auto Scaling. Cela permet de s'assurer que les ressources de calcul et d'application sont disponibles lorsque vous en avez besoin. Par exemple, il peut aider en doublant le nombre de tâches dans un service ECS pour répondre à une augmentation immédiate de 100 % de la demande.

# Choix des tailles de tâches Fargate

Si vous exécutez vos tâches surAWS Fargate, vous devez déclarer les limites de l'UC et de la mémoire de la tâche dans votre définition de tâche. ECS utilise ces limites pour déterminer le type d'instance Fargate sur lequel exécuter votre tâche. Les limites que vous déterminez doivent être supérieures ou égales à toutes les réservations que vous avez déclarées. Dans la plupart des cas, vous pouvez les définir sur la somme des réservations de chacun des conteneurs déclarés dans votre définition de tâche. Ensuite, arrondissez également le nombre jusqu'à la taille d'instance Fargate la plus proche. Pour de plus amples informations sur les tailles disponibles, consultez.UC et mémoire de niveau tâchedans leGuide du développeur Amazon Elastic Container Service.

# Choix du type d'instance Amazon EC2

Si vous utilisez Amazon EC2 pour fournir de la capacité pour votre cluster ECS, vous pouvez choisir parmi un large choix de types d'instance. Tous les types et familles d'instances Amazon EC2 sont compatibles avec ECS.

Pour déterminer les types d'instance que vous pouvez utiliser, commencez par éliminer les types d'instance ou les familles d'instances qui ne répondent pas aux exigences spécifiques de votre application. Par exemple, si votre application nécessite un GPU, vous pouvez exclure tous les types d'instance qui ne possèdent pas de GPU. Cependant, vous devriez également considérer d'autres exigences, aussi. Par exemple, considérez l'architecture CPU, le débit réseau et si le stockage d'instance est requis. Ensuite, examinez la quantité de CPU et de mémoire fournie par chaque type d'instance. En règle générale, le processeur et la mémoire doivent être suffisamment grands pour contenir au moins un réplica de la tâche à exécuter.

Vous pouvez choisir parmi les types d'instance compatibles avec votre application. Avec des instances plus volumineuses, vous pouvez lancer plusieurs tâches en même temps. De plus, avec des instances plus petites, vous pouvez effectuer une mise à l'échelle plus fine afin de réduire les coûts. Vous n'avez pas besoin de choisir un seul type d'instance Amazon EC2 qui conviendra à toutes les applications de votre cluster. Au lieu de cela, vous pouvez créer plusieurs groupes Auto Scaling. Chaque groupe peut compter un type d'instance différent. Ensuite, vous pouvez créer un fournisseur de capacité Amazon EC2 pour chacun de ces groupes. Enfin, dans la stratégie Fournisseur de capacité de votre service et tâche, vous pouvez sélectionner le fournisseur de capacité qui convient le mieux à ses besoins.

# Utilisation d'Amazon EC2 Spot et de FARGATE_SPOT

La capacité ponctuelle peut permettre d'économiser considérablement les coûts par rapport aux instances à la demande. La capacité ponctuelle est une capacité excédentaire dont le prix est nettement inférieur à celui de la capacité à la demande ou réservée. La capacité ponctuelle convient aux charges de travail de traitement par lots et d'apprentissage automatique, ainsi qu'aux environnements de développement et de mise en scène. Plus généralement, il convient à toute charge de travail qui tolère des temps d'arrêt temporaires.

Comprenez que les conséquences suivantes, car la capacité Ponctuelle peut ne pas être disponible tout le temps.

- Tout d'abord, pendant les périodes de demande extrêmement élevée, la capacité ponctuelle peut ne pas être disponible. Cela peut entraîner le retard de la tâche Fargate Spot et des lancements d'instance Ponctuelle Amazon EC2. Dans ces événements, les services ECS tentent de nouveau les tâches de lancement, et les groupes Amazon EC2 Auto Scaling tentent également de réessayer les instances de lancement, jusqu'à ce que la capacité requise devienne disponible. Fargate et Amazon EC2 ne remplacent pas la capacité ponctuelle par une capacité à la demande.
- Deuxièmement, lorsque la demande globale de capacité augmente, les instances et les tâches ponctuelles peuvent être interrompues avec seulement un avertissement de deux minutes. Une fois l'avertissement envoyé, les tâches doivent commencer un arrêt ordonné si nécessaire avant que l'instance ne soit complètement terminée. Cela permet de minimiser la possibilité d'erreurs. Pour plus d'informations sur un arrêt gracieux, consultez.Arrêts gracieux avec ECS.

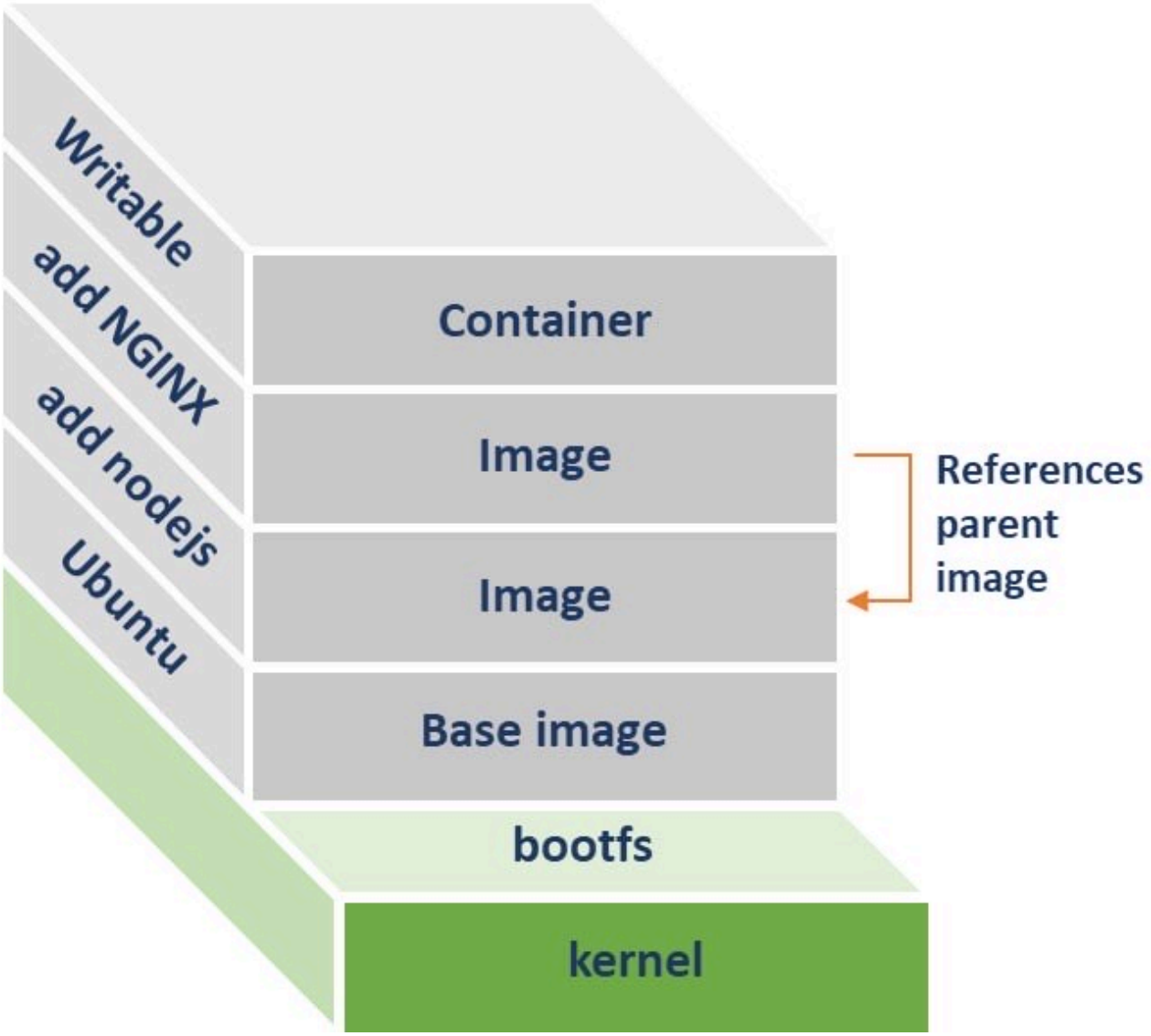
Pour réduire les pénuries de capacité ponctuelle, prenez en compte les recommandations suivantes :

- Utilisation de plusieurs régions et zones de disponibilité. La capacité ponctuelle varie selon la région et la zone de disponibilité. Vous pouvez améliorer la disponibilité des instances ponctuelles en exécutant vos charges de travail dans plusieurs régions et zones de disponibilité. Si possible, spécifiez des sous-réseaux dans toutes les zones de disponibilité des régions où vous exécutez vos tâches et instances.
- Utilisation de plusieurs types d'instances Amazon EC2 Lorsque vous utilisez des stratégies d'instance mixte avec Amazon EC2 Auto Scaling, plusieurs types d'instance sont lancés dans votre groupe Auto Scaling. Cela garantit qu'une demande de capacité ponctuelle peut être satisfaite si nécessaire. Pour maximiser la fiabilité et réduire la complexité, utilisez des types d'instances avec à peu près la même quantité de CPU et de mémoire dans votre stratégie d'instances mixtes. Ces instances peuvent provenir d'une génération différente ou de variantes du même type d'instance de base. Notez qu'ils peuvent venir avec des fonctionnalités supplémentaires que vous n'avez peut-être pas besoin. Un exemple d'une telle liste pourrait inclure m4.large, m5.large, m5a.large, m5d.large, m5n.large, m5dn.large et m5ad.large. Pour plus d'informations, consultez Groupes Auto Scaling avec types d'instance et options d'achat multiples dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.
- Utilisez la stratégie d'allocation Ponctuelle optimisée pour la capacité Avec Amazon EC2 Spot, vous pouvez choisir entre les stratégies d'allocation optimisées pour la capacité et les coûts. Si vous choisissez la stratégie optimisée pour la capacité lors du lancement d'une nouvelle instance, Amazon EC2 Spot sélectionne le type d'instance dont la disponibilité est la plus élevée dans la zone de disponibilité sélectionnée. Cela permet de réduire la possibilité que l'instance soit interrompue peu après son lancement.

# Meilleures pratiques - Stockage persistant

Vous pouvez utiliser Amazon ECS pour exécuter des applications conteneurisées avec état à grande échelle en utilisantAWS, tels qu'Amazon EFS, Amazon EBS ou Amazon FSx for Windows File Server, qui fournissent la persistance des données à des conteneurs éphémères intrinsèquement. Le termePersistance des donnéessignifie que les données elles-mêmes durent plus longtemps que le processus qui les a créées. Persistance des données dansAWSest réalisé par le couplage des services de calcul et de stockage. Comme Amazon EC2, vous pouvez également utiliser Amazon ECS pour dissocier le cycle de vie de vos applications conteneurisées des données qu'elles consomment et produisent. UtiliserAWS, les tâches Amazon ECS peuvent persister des données même après la fin des tâches.

Par défaut, les conteneurs ne conservent pas les données qu'ils produisent. Lorsqu'un conteneur est terminé, les données qu'il a écrites dans sa couche accessible en écriture sont détruites avec le conteneur. Cela rend les conteneurs adaptés aux applications sans état qui n'ont pas besoin de stocker des données localement. Les applications conteneurisées qui nécessitent la persistance des données ont besoin d'un backend de stockage qui n'est pas détruit lorsque le conteneur de l'application se termine.




Une image de conteneur est créée à partir d'une série de calques. Chaque calque représente une instruction dans le fichier Dockerfile à partir de laquelle l'image a été créée. Chaque couche est en lecture seule, à l'exception du conteneur. Autrement dit, lorsque vous créez un conteneur, une couche accessible en écriture est ajoutée sur les couches sous-jacentes. Tous les fichiers que le conteneur crée, supprime ou modifie sont écrits dans la couche accessible en écriture. Lorsque le conteneur se termine, la couche accessible en écriture est également supprimée simultanément. Un nouveau conteneur qui utilise la même image a son propre calque accessible en écriture. Cette couche n'inclut aucune modification. Par conséquent, les données d'un conteneur doivent toujours être stockées en dehors de la couche accessible en écriture du conteneur.

Avec Amazon ECS, vous pouvez exécuter des conteneurs avec état à l'aide de volumes. Amazon ECS est intégré à Amazon EFS en mode natif et utilise des volumes intégrés à Amazon EBS. Pour les conteneurs Windows, Amazon ECS s'intègre à Amazon FSx for Windows File Server pour fournir un stockage persistant.

Rubriques



# Choix du type de stockage adapté à vos conteneurs

Les applications qui s'exécutent dans un cluster Amazon ECS peuvent utiliser une variété deAWSLes services de stockage et les produits tiers afin de fournir du stockage persistant pour les charges de travail avec état. Vous devez choisir votre backend de stockage pour votre application conteneurisée en fonction de l'architecture et des exigences de stockage de votre application. Pour plus d'informations surAWSservices de stockage, voirStockage dans le cloudAWS.

Pour les clusters Amazon ECS qui contiennent des instances Linux ou qui sont utilisés avec Fargate, Amazon ECS s'intègre avec Amazon EFS et Amazon EBS pour fournir un stockage de conteneurs. La différence la plus distinctive entre Amazon EFS et Amazon EBS est que vous pouvez monter simultanément un système de fichiers Amazon EFS sur des milliers de tâches Amazon ECS. En revanche, les volumes Amazon EBS ne prennent pas en charge l'accès simultané. Dans ce contexte, Amazon EFS est l'option de stockage recommandée pour les applications conteneurisées qui s'adaptent horizontalement. C'est parce qu'il prend en charge la concurrence. Amazon EFS stocke vos données de manière redondante dans plusieurs zones de disponibilité et offre un accès à faible latence à partir des tâches Amazon ECS, quelle que soit la zone de disponibilité. Amazon EFS prend en charge les tâches exécutées sur Amazon EC2 et Fargate.

Supposons que vous ayez une application telle qu'une base de données transactionnelle qui nécessite une latence inférieure à la milliseconde et n'a pas besoin d'un système de fichiers partagé lorsqu'elle est mise à l'échelle horizontale. Pour une telle application, nous vous recommandons d'utiliser des volumes Amazon EBS pour le stockage persistant. Actuellement, Amazon ECS prend en charge les volumes Amazon EBS pour les tâches hébergées sur Amazon EC2 uniquement. La prise en charge des volumes Amazon EBS n'est pas disponible pour les tâches sur Fargate. Avant

d'utiliser des volumes Amazon EBS avec des tâches Amazon ECS, vous devez d'abord attacher des volumes Amazon EBS aux instances de conteneur et gérer les volumes séparément du cycle de vie de la tâche.

Pour les clusters contenant des instances Windows, Amazon FSx for Windows File Server fournit un stockage persistant pour les conteneurs. Les systèmes de fichiers Amazon FSx for Windows File Server prennent en charge les déploiements Multi-AZ. Grâce à ces déploiements, vous pouvez partager un système de fichiers avec des tâches Amazon ECS s'exécutant sur plusieurs zones de disponibilité.

Vous pouvez également utiliser le stockage d'instance Amazon EC2 pour la persistance des données pour les tâches Amazon ECS hébergées sur Amazon EC2 à l'aide de montages de liaison ou de volumes Docker. Lors de l'utilisation de montures de liaison ou de volumes Docker, les conteneurs stockent des données sur le système de fichiers d'instance de conteneur. Une limitation de l'utilisation d'un système de fichiers hôte pour le stockage de conteneurs est que les données ne sont disponibles que sur une seule instance de conteneur à la fois. Cela signifie que les conteneurs ne peuvent s'exécuter que sur l'hôte où résident les données. Par conséquent, l'utilisation du stockage hôte n'est recommandée que dans les scénarios où la réplication des données est gérée au niveau de l'application.

# Volumes Amazon EFS

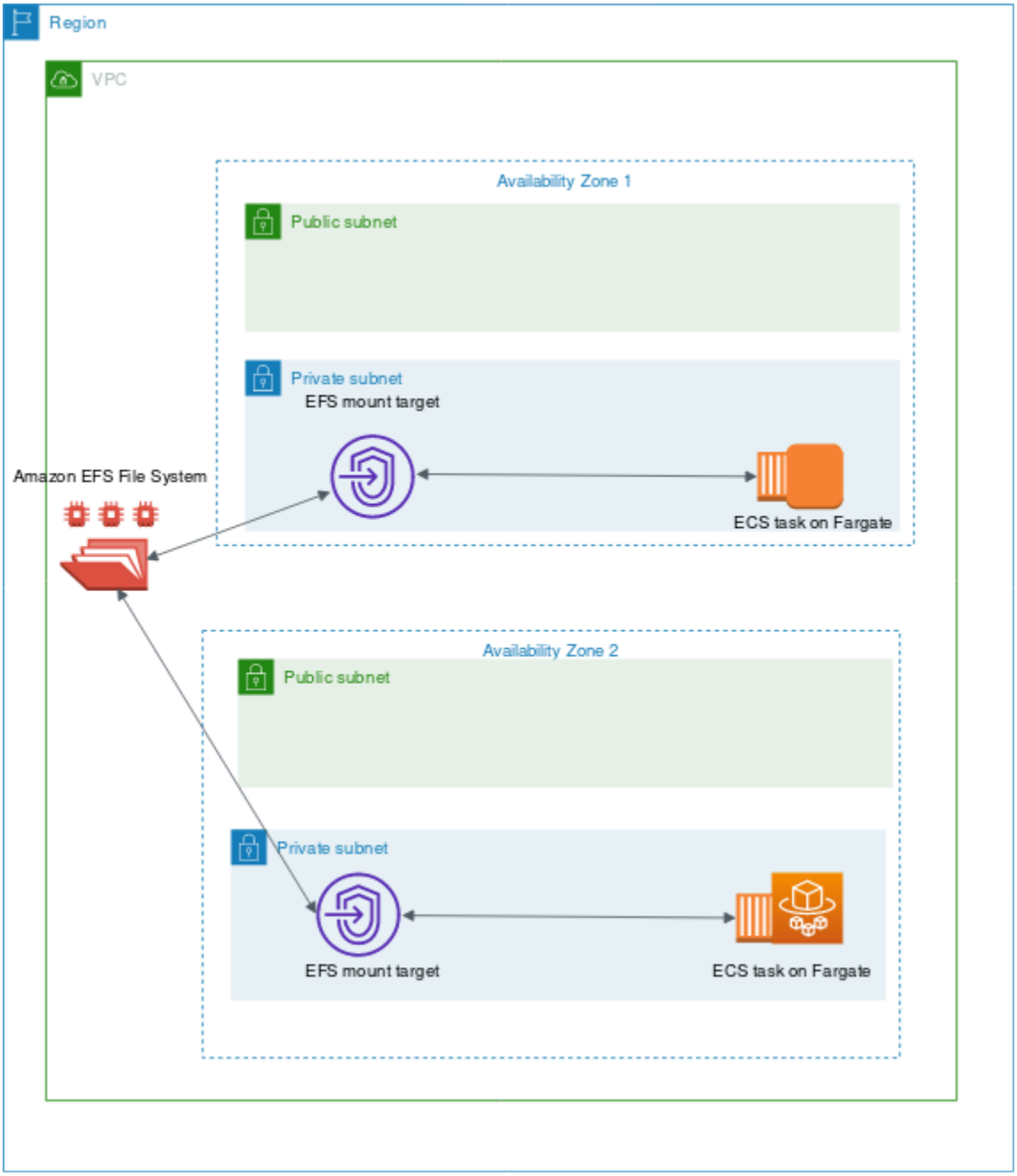
Amazon Elastic File System (Amazon EFS) fournit un système de fichiers NFS élastique simple, évolutif et entièrement géré. Il est conçu pour pouvoir évoluer à la demande jusqu'à des pétaoctets sans perturber les applications. Il peut être mis à l'échelle avant ou à l'extérieur lorsque vous ajoutez et supprimez des fichiers.

Vous pouvez exécuter vos applications avec état dans Amazon ECS à l'aide de volumes Amazon EFS pour fournir un stockage persistant. Tâches Amazon ECS exécutées sur des instances Amazon EC2 ou sur Fargate en utilisant la version de plate-forme`1.4.0`et versions ultérieures peuvent monter un système de fichiers Amazon EFS existant. Étant donné que plusieurs conteneurs peuvent monter et accéder simultanément à un système de fichiers Amazon EFS, vos tâches ont accès au même ensemble de données, quel que soit l'endroit où elles sont hébergées.

Pour monter un système de fichiers Amazon EFS dans votre conteneur, vous pouvez référencez le système de fichiers Amazon EFS et le point de montage du conteneur dans votre définition de tâche Amazon ECS. Voici un extrait d'une définition de tâche utilisant Amazon EFS pour le stockage de conteneurs.

```
...
"containerDefinitions":[
    {
        "mountPoints": [
            {
                "containerPath": "/opt/my-app",
                 "sourceVolume": "Shared-EFS-Volume"
            }
    }
  ]
...
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "transitEncryption": "DISABLED",
        "rootDirectory": ""
      },
      "name": "Shared-EFS-Volume"
    }
  ]
```

Amazon EFS stocke des données de manière redondante sur plusieurs zones de disponibilité au sein d'une même région. Une tâche Amazon ECS monte le système de fichiers Amazon EFS à l'aide d'une cible de montage Amazon EFS dans sa zone de disponibilité. Une tâche Amazon ECS ne peut monter un système de fichiers Amazon EFS que si le système de fichiers Amazon EFS a une cible de montage dans la zone de disponibilité dans laquelle la tâche s'exécute. Par conséquent, une meilleure pratique consiste à créer des cibles de montage Amazon EFS dans toutes les zones de disponibilité dans lesquelles vous prévoyez d'héberger des tâches Amazon ECS.




Pour de plus amples informations, veuillez consulterVolumes Amazon EFSdans leGuide du développeur Amazon Elastic Container Service.

## Contrôles de sécurité et d'accès

Amazon EFS offre des fonctionnalités de contrôle d'accès que vous pouvez utiliser pour vous assurer que les données stockées dans un système de fichiers Amazon EFS sont sécurisées et accessibles uniquement à partir des applications qui en ont besoin. Vous pouvez sécuriser les données en activant le chiffrement au repos et en transit. Pour de plus amples informations, veuillez

consulterChiffrement des données dans Amazon EFSdans leGuide de l'utilisateur Amazon Elastic File System.

Outre le chiffrement des données, vous pouvez également utiliser Amazon EFS pour restreindre l'accès à un système de fichiers. Il existe trois méthodes pour implémenter le contrôle d'accès dans EFS.

- Groupes de sécurité : avec les cibles de montage Amazon EFS, vous pouvez configurer un groupe de sécurité utilisé pour autoriser et refuser le trafic réseau. Vous pouvez configurer le groupe de sécurité attaché à Amazon EFS pour autoriser le trafic NFS (port 2049) à partir du groupe de sécurité attaché à vos instances Amazon ECS ou, lorsque vous utilisez le`awsvpc`, la tâche Amazon ECS.
- IAM : vous pouvez restreindre l'accès à un système de fichiers Amazon EFS à l'aide d'IAM. Lorsqu'elles sont configurées, les tâches Amazon ECS nécessitent un rôle IAM pour l'accès au système de fichiers pour monter un système de fichiers EFS. Pour de plus amples informations, veuillez consulterUtilisation d'IAM pour contrôler l'accès aux données du système de fichiersdans leGuide de l'utilisateur Amazon Elastic File System.

  Les stratégies IAM peuvent également appliquer des conditions prédéfinies, telles qu'exiger qu'un client utilise TLS lors de la connexion à un système de fichiers Amazon EFS. Pour de plus amples informations, veuillez consulterClés de condition Amazon EFS pour les clientsdans leGuide de l'utilisateur Amazon Elastic File System.
- Points d'accès Amazon EFSLes points d'accès Amazon EFS sont des points d'entrée spécifiques à l'application dans un système de fichiers Amazon EFS. Vous pouvez utiliser des points d'accès pour appliquer une identité d'utilisateur, y compris les groupes POSIX de l'utilisateur, pour toutes les demandes de système de fichiers effectuées via le point d'accès. Les points d'accès peuvent également appliquer de manière forcée un répertoire racine différent pour le système de fichiers. Il s'agit de manière forcée que les clients puissent uniquement accéder aux données stockées dans le répertoire spécifié ou dans les sous-répertoires.

Envisagez d'implémenter les trois contrôles d'accès sur un système de fichiers Amazon EFS pour une sécurité maximale. Par exemple, vous pouvez configurer le groupe de sécurité attaché à un point de montage Amazon EFS pour autoriser uniquement l'entrée de trafic NFS à partir d'un groupe de sécurité associé à votre instance de conteneur ou à la tâche Amazon ECS. En outre, vous pouvez configurer Amazon EFS pour exiger un rôle IAM pour accéder au système de fichiers, même si la connexion provient d'un groupe de sécurité autorisé. Enfin, vous pouvez utiliser les points d'accès

Amazon EFS pour appliquer les autorisations utilisateur POSIX et spécifier des répertoires racine pour les applications.

L'extrait de définition de tâche suivant indique comment monter un système de fichiers Amazon EFS à l'aide d'un point d'accès.

```
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "authorizationConfig": {
          "acessPointId": "fsap-1234",
          "iam": "ENABLED"
        },
        "transitEncryption": "ENABLED",
        "rootDirectory": ""
      },
      "name": "my-filesystem"
    }
]
```

## Performance

Amazon EFS offre deux modes de performance : Usage général et E/S max. Usage général convient aux applications sensibles à la latence telles que les systèmes de gestion de contenu et les outils CI/CD. En revanche, les systèmes de fichiers d'E/S Max conviennent aux charges de travail telles que l'analyse de données, le traitement des médias et l'apprentissage automatique. Ces charges de travail doivent effectuer des opérations parallèles à partir de centaines, voire de milliers de conteneurs et nécessitent le débit agrégé et les E/S par seconde le plus élevé possible. Pour de plus amples informations, veuillez consulterModes de performance Amazon EFSdans leGuide de l'utilisateur Amazon Elastic File System.

Certaines charges de travail sensibles à la latence nécessitent à la fois les niveaux d'E/S supérieurs fournis par le mode de performances E/S max et la latence plus faible fournie par le mode de performances Usage général. Pour ce type de charge de travail, nous vous recommandons de créer plusieurs systèmes de fichiers en mode de performance Usage général. De cette façon, vous pouvez répartir votre charge de travail applicative sur tous ces systèmes de fichiers, tant que la charge de travail et les applications peuvent assurer la prise en charge.

## Throughput

Tous les systèmes de fichiers Amazon EFS ont un débit mesuré associé qui est déterminé par la quantité de débit provisionné pour les systèmes de fichiers à l'aide deDébit allouéou la quantité de données stockées dans la classe de stockage EFS Standard ou One Zone pour les systèmes de fichiers à l'aide deDébit en mode rafale. Pour de plus amples informations, veuillez consulterPrésentation du débit mesurédans leGuide de l'utilisateur Amazon Elastic File System.

Le mode de débit par défaut pour les systèmes de fichiers Amazon EFS est le mode de découpage. Avec le mode d'éclatement, le débit disponible pour un système de fichiers évolue en fonction de la croissance d'un système de fichiers. Étant donné que les charges de travail basées sur des fichiers augmentent généralement, nécessitant des débits élevés pendant des périodes de temps et des débits inférieurs le reste du temps, Amazon EFS est conçu pour permettre des débits élevés pendant des périodes de temps. En outre, comme de nombreuses charges de travail sont lourdes en lecture, les opérations de lecture sont mesurées à un rapport de 1:3 par rapport aux autres opérations NFS (comme l'écriture).

Tous les systèmes de fichiers Amazon EFS offrent des performances de base cohérentes de 50 Mo/s pour chaque To de stockage Amazon EFS Standard ou Amazon EFS One Zone. Tous les systèmes de fichiers (quelle que soit leur taille) peuvent exploser jusqu'à 100 Mo/s. Les systèmes de fichiers disposant de plus de 1 To de stockage EFS Standard ou EFS One Zone peuvent atteindre 100 Mo/s par To. Étant donné que les opérations de lecture sont mesurées à un rapport de 1:3, vous pouvez piloter jusqu'à 300 Mibit/s pour chaque TiB de débit de lecture. Lorsque vous ajoutez des données à votre système de fichiers, le débit maximal disponible pour le système de fichiers évolue de manière linéaire et automatique avec votre stockage dans la classe de stockage Amazon EFS Standard. Si vous avez besoin d'un débit supérieur à ce que vous pouvez obtenir avec votre quantité de données stockées, vous pouvez configurer le débit provisionné en fonction de la quantité spécifique requise par votre charge de travail.

Le débit du système de fichiers est partagé entre toutes les instances Amazon EC2 connectées à un système de fichiers. Par exemple, un système de fichiers de 1 To pouvant atteindre 100 Mo/s de débit peut piloter 100 Mo/s à partir d'une seule instance Amazon EC2 peut chaque disque 10 Mo/s. Pour de plus amples informations, veuillez consulterPerformances Amazon EFSdans leGuide de l'utilisateur Amazon Elastic File System.

## Optimisation des coûts

Amazon EFS simplifie la mise à l'échelle du stockage pour vous. Les systèmes de fichiers Amazon EFS augmentent automatiquement à mesure que vous ajoutez des données supplémentaires.

Surtout avec Amazon EFSDébit en mode rafaleLe débit sur Amazon EFS augmente à mesure que la taille de votre système de fichiers augmente dans la classe de stockage standard. Pour améliorer le débit sans payer de frais supplémentaires pour le débit provisionné sur un système de fichiers EFS, vous pouvez partager un système de fichiers Amazon EFS avec plusieurs applications. À l'aide des points d'accès Amazon EFS, vous pouvez implémenter l'isolation du stockage dans des systèmes de fichiers Amazon EFS partagés. Ce faisant, même si les applications partagent toujours le même système de fichiers, elles ne peuvent pas accéder aux données à moins que vous ne l'autorisiez.

À mesure que vos données augmentent, Amazon EFS vous aide à déplacer automatiquement les fichiers rarement consultés vers une classe de stockage inférieure. La classe de stockage Amazon EFS Standard-Inrequent Access (IA) permet de réduire les coûts de stockage pour les fichiers qui ne sont pas consultés tous les jours. Il le fait sans sacrifier la haute disponibilité, la haute durabilité, l'élasticité et l'accès au système de fichiers POSIX fournis par Amazon EFS. Pour de plus amples informations, veuillez consulterClasses de stockage Amazon EFSdans leGuide de l'utilisateur Amazon Elastic File System.

Envisagez d'utiliser les politiques de cycle de vie Amazon EFS pour économiser automatiquement de l'argent en déplaçant les fichiers rarement consultés vers le stockage Amazon EFS IA. Pour de plus amples informations, veuillez consulterGestion du cycle de vie Amazon EFSdans leGuide de l'utilisateur Amazon Elastic File System.

Lorsque vous créez un système de fichiers Amazon EFS, vous pouvez choisir si Amazon EFS réplique vos données sur plusieurs zones de disponibilité (Standard) ou stocke vos données de manière redondante dans une seule zone de disponibilité. La classe de stockage Amazon EFS One Zone peut réduire les coûts de stockage d'une marge significative par rapport aux classes de stockage Amazon EFS Standard. Envisagez d'utiliser la classe de stockage Amazon EFS One Zone pour les charges de travail qui ne nécessitent pas de résilience Multi-AZ. Vous pouvez réduire davantage le coût du stockage Amazon EFS One Zone en déplaçant les fichiers rarement accessibles vers Amazon EFS One Zone-Inrequent Access. Pour de plus amples informations, veuillez consulterAccès peu fréquent Amazon EFS.

## Protection des données

Amazon EFS stocke vos données de manière redondante dans plusieurs zones de disponibilité pour les systèmes de fichiers à l'aide de classes de stockage standard Si vous sélectionnez des classes de stockage Amazon EFS One Zone, vos données sont stockées de manière redondante dans une seule zone de disponibilité. En outre, Amazon EFS est conçu pour fournir 99,999999999 % (11 9) de durabilité sur une année donnée.

Comme pour n'importe quel environnement, il est recommandé d'avoir une sauvegarde et de mettre en place des mesures de protection contre les suppressions accidentelles. Pour les données Amazon EFS, cette meilleure pratique comprend une sauvegarde fonctionnelle et régulièrement testée à l'aide deAWS Backup. Les systèmes de fichiers utilisant des classes de stockage Amazon EFS One Zone sont configurés pour sauvegarder automatiquement les fichiers par défaut lors de la création du système de fichiers, sauf si vous choisissez de désactiver cette fonctionnalité. Pour de plus amples informations, veuillez consulterProtection des données pour Amazon EFSdans leGuide de l'utilisateur Amazon Elastic File System.

## Cas d'utilisation

Amazon EFS fournit un accès partagé parallèle qui augmente et diminue automatiquement à mesure que des fichiers sont ajoutés et supprimés. Par conséquent, Amazon EFS convient à toute application nécessitant un stockage avec des fonctionnalités telles que faible latence, débit élevé et cohérence en lecture après écriture. Amazon EFS est un backend de stockage idéal pour les applications qui évoluent horizontalement et qui nécessitent un système de fichiers partagé. Des charges de travail telles que l'analyse de données, le traitement multimédia, la gestion de contenu et le service Web font partie des cas d'utilisation courants Amazon EFS.

Un cas d'utilisation où Amazon EFS peut ne pas convenir est pour les applications nécessitant une latence inférieure à la milliseconde. Il s'agit généralement d'une exigence pour les systèmes de base de données transactionnels. Nous vous recommandons d'exécuter des tests de performances de stockage afin de déterminer l'impact de l'utilisation d'Amazon EFS pour les applications sensibles à la latence. Si les performances des applications se dégradent lors de l'utilisation d'Amazon EFS, pensez à Amazon EBS io2 Block Express, qui fournit une latence d'E/S inférieure à la milliseconde et à faible variance sur les instances Nitro. Pour de plus amples informations, veuillez consulter Types de volumes Amazon EBS dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.

Certaines applications échouent si leur stockage sous-jacent est modifié de façon inattendue. Par conséquent, Amazon EFS n'est pas le meilleur choix pour ces applications. Au contraire, vous préférerez peut-être utiliser un système de stockage qui n'autorise pas l'accès simultané à partir de plusieurs emplacements.

# Volumes Docker

Les volumes Docker sont une fonctionnalité du moteur d'exécution du conteneur Docker qui permet aux conteneurs de conserver des données en montant un répertoire à partir du système de fichiers de l'hôte. Des pilotes de volume Docker (également appelés plug-ins) permettent d'intégrer des

volumes de conteneur avec des systèmes de stockage externe, tels qu'Amazon EBS. Les volumes Docker sont uniquement pris en charge avec des tâches Amazon EC2 sur des instances Amazon EC2.

Les tâches Amazon ECS peuvent utiliser des volumes Docker pour conserver des données à l'aide de volumes Amazon EBS. Pour ce faire, connectez un volume Amazon EBS à une instance Amazon EC2, puis montez le volume dans une tâche à l'aide de volumes Docker. Un volume Docker peut être partagé entre plusieurs tâches Amazon ECS sur l'hôte.

La limitation des volumes Docker est que le système de fichiers utilisé par la tâche est lié à l'instance Amazon EC2 spécifique. Si l'instance s'arrête pour une raison quelconque et que la tâche est placée sur une autre instance, les données sont perdues. Vous pouvez affecter des tâches à des instances pour vous assurer que les volumes EBS associés sont toujours disponibles pour les tâches.

Pour de plus amples informations, veuillez consulterVolumes Dockerdans leGuide du développeur Amazon Elastic Container Service.

## Cycle de vie des volumes Amazon EBS

Il existe deux modèles d'utilisation clés avec le stockage de conteneurs et Amazon EBS. La première est lorsqu'une application doit persister des données et empêcher la perte de données lorsque son conteneur se termine. Un exemple de ce type d'application serait une base de données transactionnelle comme MySQL. Lorsqu'une tâche MySQL se termine, une autre tâche est censée la remplacer. Dans ce scénario, le cycle de vie du volume est séparé du cycle de vie de la tâche. Lorsque vous utilisez EBS pour conserver des données de conteneur, il est recommandé d'utiliser des contraintes de placement de tâche pour limiter le placement de la tâche à un seul hôte avec le volume EBS attaché.

La seconde est lorsque le cycle de vie du volume est indépendant du cycle de vie de la tâche. Ceci est particulièrement utile pour les applications qui nécessitent un stockage à haute performance et à faible latence, mais qui n'ont pas besoin de persister des données après la fin de la tâche. Par exemple, une charge de travail ETL qui traite de grands volumes de données peut nécessiter un stockage à haut débit. Amazon EBS convient à ce type de charge de travail car il fournit des volumes hautes performances pouvant atteindre 256 000 E/S par seconde. Lorsque la tâche se termine, le réplica de remplacement peut être placé en toute sécurité sur n'importe quel hôte Amazon EC2 du cluster. Tant que la tâche a accès à un backend de stockage capable de répondre à ses exigences en matière de performances, la tâche peut remplir sa fonction. Par conséquent, aucune contrainte de placement de tâches n'est nécessaire dans ce cas.

Si les instances Amazon EC2 de votre cluster sont associées à plusieurs types de volumes Amazon EBS, vous pouvez utiliser des contraintes de placement des tâches pour vous assurer que les tâches sont placées sur des instances avec un volume Amazon EBS approprié attaché. Supposons par exemple qu'un cluster ait certaines instances avec ungp2, tandis que d'autres utilisentio1volumes. Vous pouvez attacher des attributs personnalisés aux instances avecio1, puis utilisez les contraintes de placement des tâches pour vous assurer que vos tâches intensives d'E/S sont toujours placées sur des instances de conteneur avecio1volumes.

Procédez comme suit :AWS CLIest utilisée pour placer des attributs sur une instance de conteneur Amazon ECS.

```
aws ecs put-attributes \
     --attributes name=EBS,value=io1,targetId=<your-container-instance-arn>
```

## Disponibilité des données Amazon EBS

Les conteneurs sont généralement de courte durée, fréquemment créés et terminés à mesure que les applications évoluent horizontalement. Il est recommandé d'exécuter des charges de travail dans plusieurs zones de disponibilité afin d'améliorer la disponibilité de vos applications. Amazon ECS vous permet de contrôler le placement des tâches à l'aide de stratégies de placement des tâches et de contraintes de placement des tâches. Lorsqu'une charge de travail persiste ses données à l'aide de volumes Amazon EBS, ses tâches doivent être placées dans la même zone de disponibilité que le volume Amazon EBS. Nous vous recommandons également de définir une contrainte de placement qui limite la zone de disponibilité dans laquelle une tâche peut être placée. Cela garantit que vos tâches et les volumes correspondants sont toujours situés dans la même zone de disponibilité.

Lorsque vous exécutez des tâches autonomes, vous pouvez contrôler la zone de disponibilité de la tâche en définissant des contraintes de placement à l'aide de l'attribut zone de disponibilité.

```
attribute:ecs.availability-zone == us-east-1a
```

Lorsque vous exécutez des applications qui pourraient bénéficier de l'exécution dans plusieurs zones de disponibilité, envisagez de créer un service Amazon ECS différent pour chaque zone de disponibilité. Cela garantit que les tâches nécessitant un volume Amazon EBS sont toujours placées dans la même zone de disponibilité que le volume associé.

Nous vous recommandons de créer des instances de conteneur dans chaque zone de disponibilité, en attachant des volumes Amazon EBS à l'aide deModèles de lancement, et en ajoutantAttributs personnalisésaux instances pour les différencier des autres instances de conteneur dans le cluster

Amazon ECS. Lors de la création de services, configurez les contraintes de placement des tâches pour vous assurer qu'Amazon ECS place les tâches dans la zone de disponibilité et l'instance appropriées. Pour de plus amples informations, veuillez consulterExemples de contrainte de placement des tâchesdans leGuide du développeur Amazon Elastic Container Service.

### Plugins de volume Docker

Les plugins Docker tels que Portworx fournissent une abstraction entre le volume Docker et le volume Amazon EBS. Ces plugins peuvent créer dynamiquement un volume Amazon EBS au démarrage de votre tâche nécessitant un volume. Portworx peut également attacher un volume à un nouvel hôte lorsqu'un conteneur se termine, et son réplica ultérieur est placé sur une autre instance de conteneur. Il réplique également les données de volume de chaque conteneur entre les nœuds Amazon ECS et entre les zones de disponibilité. Pour de plus amples informations, veuillez consulterPortworx.

## Amazon FSx for Windows File Server

Amazon FSx for Windows File Server fournit un stockage de fichiers entièrement géré, hautement fiable et évolutif accessible via le protocole SMB (Server Message Block) standard. Il est construit sur Windows Server et offre un large éventail de fonctionnalités administratives telles que les quotas d'utilisateurs, la restauration de fichiers de l'utilisateur final et l'intégration de Microsoft Active Directory (AD). Il offre des options de déploiement Single-AZ et Multi-AZ, des sauvegardes entièrement gérées et le chiffrement des données au repos et en transit.

Amazon ECS prend en charge l'utilisation d'Amazon FSx for Windows File Server dans les définitions de tâches Windows Amazon ECS permettant le stockage persistant en tant que point de montage via le protocole SMBv3 à l'aide d'une fonctionnalité SMB appelée GlobalMappings.

Pour configurer l'intégration Amazon FSx for Windows File Server et Amazon ECS, l'instance de conteneur Windows doit être un membre de domaine sur un service de domaine Active Directory (AD DS), hébergé par unAWS Directory Service for Microsoft Active Directory, Active Directory local ou Active Directory auto-hébergé sur Amazon EC2.AWS Secrets Managerest utilisé pour stocker des données sensibles telles que le nom d'utilisateur et le mot de passe d'une information d'identification Active Directory qui est utilisée pour mapper le partage sur l'instance de conteneur Windows.

Pour utiliser des volumes de système de fichiers Amazon FSx for Windows File Server pour vos conteneurs, vous devez spécifier les configurations du volume et du point de montage dans votre définition de tâche. Voici un extrait d'une définition de tâche utilisant Amazon FSx for Windows File Server pour le stockage de conteneurs.

```
{
 "containerDefinitions": [{
  "name": "container-using-fsx",
  "image": "iis:2",
  "entryPoint": [
   "powershell",
   "-command"
  ],
  "mountPoints": [{
   "sourceVolume": "myFsxVolume",
   "containerPath": "\\mount\\fsx",
   "readOnly": false
  }]
 }],
 "volumes": [{
  "fsxWindowsFileServerVolumeConfiguration": {
   "fileSystemId": "fs-ID",
   "authorizationConfig": {
    "domain": "ADDOMAIN.local",
    "credentialsParameter": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:SecretName"
   },
   "rootDirectory": "share"
  }
 }]
}
```

Pour de plus amples informations, veuillez consulterVolumes Amazon FSx for Windows File Serverdans leGuide du développeur Amazon Elastic Container Service.

## Contrôles de sécurité et d'accès

Amazon FSx for Windows File Server fournit les fonctionnalités de contrôle d'accès suivantes que vous pouvez utiliser pour vous assurer que les données stockées dans un système de fichiers Amazon FSx for Windows File Server sont sécurisées et accessibles uniquement à partir des applications qui en ont besoin.

### Chiffrement des données

Amazon FSx for Windows File Server prend en charge deux formes de chiffrement pour les systèmes de fichiers. Ils sont le chiffrement des données en transit et le chiffrement des données au repos. Le chiffrement des données en transit est pris en charge sur les partages de fichiers qui sont mappés

sur une instance de conteneur prenant en charge le protocole SMB 3.0 ou plus récent. Le chiffrement des données au repos est automatiquement activé lors de la création d'un système de fichiers Amazon FSx File System FSx Amazon FSx crypte automatiquement les données en transit à l'aide du chiffrement SMB lorsque vous accédez à votre système de fichiers sans que vous ayez à modifier vos applications. Pour de plus amples informations, veuillez consulterChiffrement des données dans Amazon FSxdans leGuide de l'utilisateur Amazon FSx for Windows File Server.

### Contrôle d'accès au niveau des dossiers à l'aide des listes ACL

L'instance Windows Amazon EC2 accède aux partages de fichiers Amazon FSx à l'aide des informations d'identification Active Directory. Il utilise des listes de contrôle d'accès Windows (ACL) standard pour un contrôle d'accès précis au niveau des fichiers et des dossiers. Vous pouvez créer plusieurs informations d'identification, chacune pour un dossier spécifique dans le partage qui correspond à une tâche spécifique.

Dans l'exemple suivant, la tâche a accès au dossier`App01`à l'aide d'une information d'identification enregistrée dans Secrets Manager. Son Amazon Resource Name (ARN) est`1234`.

```
"rootDirectory": "\\path\\to\\my\\data\App01",
"credentialsParameter": "arn-1234",
"domain": "corp.fullyqualified.com",
```

Dans un autre exemple, une tâche a accès au dossier`App02`à l'aide d'une information d'identification enregistrée dans le Secrets Manager. Son ARN est 6789.

```
"rootDirectory": "\\path\\to\\my\\data\App02",
"credentialsParameter": "arn-6789",
"domain": "corp.fullyqualified.com",
```

## Cas d'utilisation

Les conteneurs ne sont pas conçus pour persister les données. Toutefois, certaines applications .NET conteneurisées peuvent nécessiter des dossiers locaux en tant que stockage persistant pour enregistrer les sorties de l'application. Amazon FSx for Windows File Server propose un dossier local dans le conteneur. Cela permet à plusieurs conteneurs de lecture-écriture sur le même système de fichiers qui est soutenu par un partage SMB.

# Bonnes pratiques de sécurité

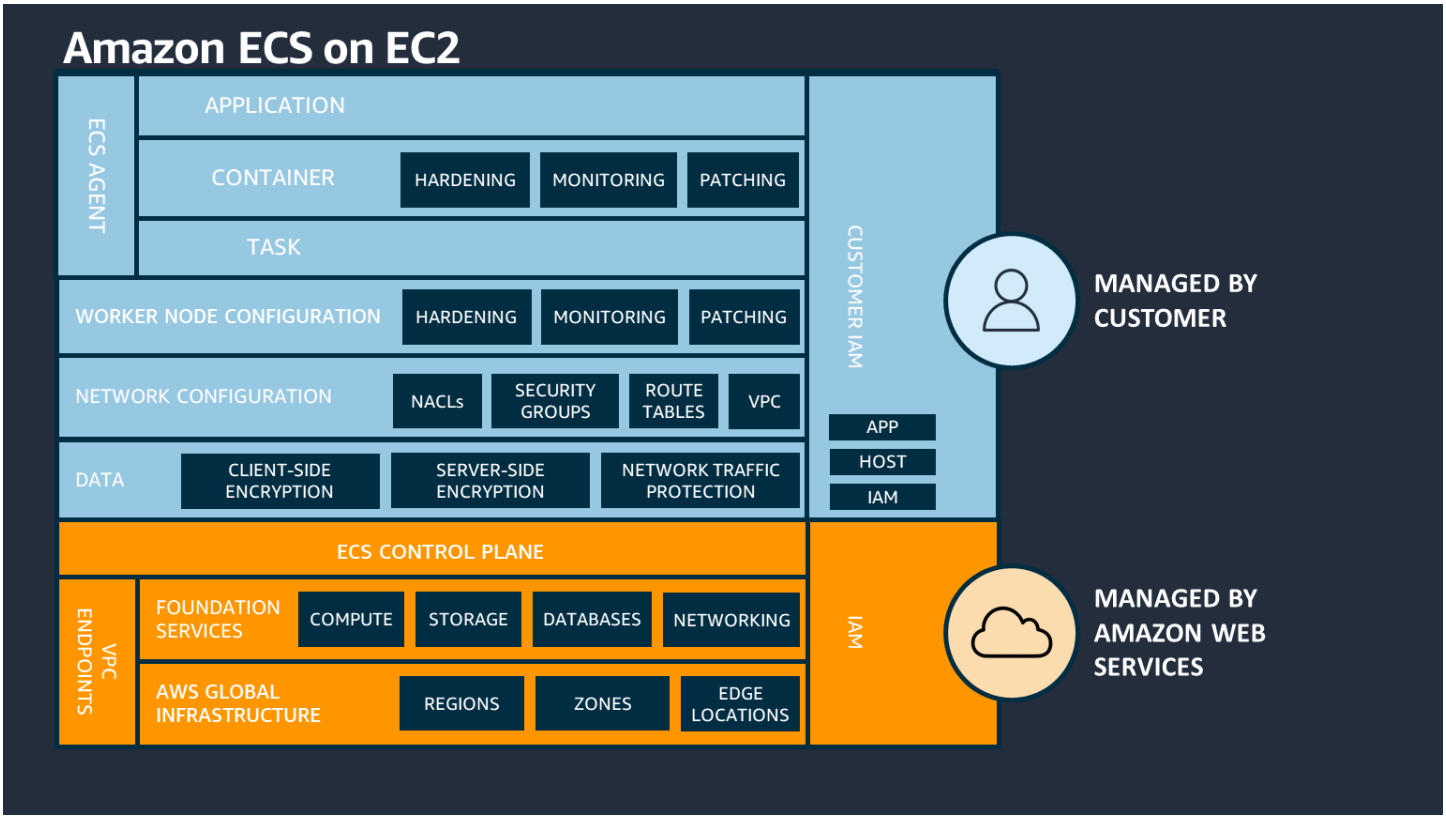
Ce guide fournit des recommandations de sécurité et de conformité pour protéger vos informations, systèmes et autres ressources qui dépendent d'Amazon ECS. Il introduit également des évaluations des risques et des stratégies d'atténuation que vous pouvez utiliser pour mieux maîtriser les contrôles de sécurité conçus pour les clusters Amazon ECS et les charges de travail qu'ils prennent en charge. Chaque rubrique de ce guide commence par un bref aperçu, suivi d'une liste de recommandations et de meilleures pratiques que vous pouvez utiliser pour sécuriser vos clusters Amazon ECS.

Rubriques

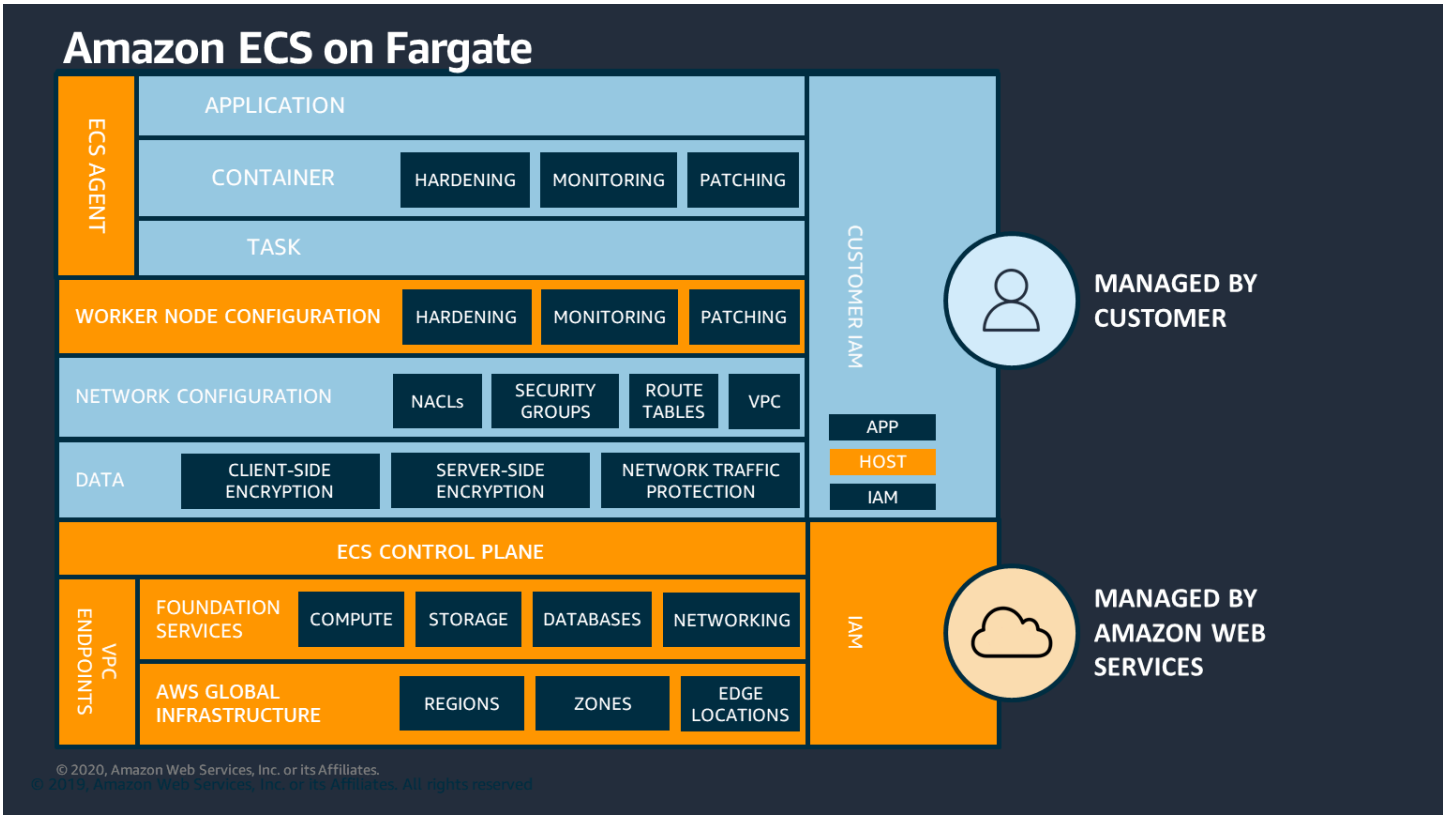


## Modèle de responsabilité partagée

La sécurité et la conformité d'un service géré comme Amazon ECS sont une resposibilité partagée entre vous etAWS. D'une manière générale,AWSest responsable de la sécurité « du » cloud alors que vous, le client, êtes responsable de la sécurité « dans » le cloud.AWSest responsable de la gestion du plan de contrôle Amazon ECS, y compris de l'infrastructure nécessaire pour fournir un service sécurisé et fiable. Et vous êtes en grande partie responsable des sujets abordés dans ce guide. Cela inclut la sécurité des données, du réseau et de l'exécution, ainsi que la journalisation et la surveillance.




En ce qui concerne la sécurité des infrastructures,AWSassume plus de responsabilité pourAWS Fargateque pour d'autres instances autogérées. Avec Fargate,AWSgère la sécurité de l'instance sous-jacente dans le cloud et le moteur d'exécution utilisé pour exécuter vos tâches. Fargate met également à l'échelle automatiquement votre infrastructure en votre nom.




Avant d'étendre vos services au cloud, vous devez comprendre les aspects de la sécurité et de la conformité dont vous êtes responsable.

Pour plus d'informations sur le modèle de responsabilité partagée, consultezModèle de responsabilité partagée.

## AWS Identity and Access Management

Vous pouvez utiliserAWS Identity and Access Management(IAM) pour gérer et contrôler l'accès à votreAWSservices et ressources au moyen de stratégies basées sur des règles à des fins d'authentification et d'autorisation. Plus précisément, à travers ce service, vous contrôlez l'accès à votreAWSà l'aide de stratégies appliquées aux utilisateurs, groupes ou rôles IAM. Parmi ces trois utilisateurs, les utilisateurs IAM sont des comptes qui peuvent avoir accès à vos ressources. Et, un rôle IAM est un ensemble d'autorisations qui peuvent être assumées par une identité authentifiée, qui n'est pas associée à une identité particulière en dehors d'IAM. Pour de plus amples informations, veuillez consulterStratégies et autorisations dans IAM ?.

# Gestion de l'accès à Amazon ECS

Vous pouvez contrôler l'accès à Amazon ECS en créant et en appliquant des stratégies IAM. Ces stratégies sont composées d'un ensemble d'actions qui s'appliquent à un ensemble spécifique de ressources. L'action d'une stratégie définit la liste des opérations (telles que les API Amazon ECS) autorisées ou refusées, tandis que la ressource contrôle quels sont les objets Amazon ECS auxquels l'action s'applique. Des conditions peuvent être ajoutées à une politique pour en restreindre la portée. Par exemple, une stratégie peut être écrite pour autoriser uniquement une action sur des tâches comportant un ensemble particulier de balises. Pour de plus amples informations, veuillez consulterFonctionnement d'Amazon ECS avec IAMdans leGuide du développeur Amazon Elastic Container Service.

## Recommendations

Nous vous recommandons d'effectuer les opérations suivantes lors de la configuration de vos rôles et stratégies IAM.

### Suivez la politique d'accès le moins privilégié

Créez des stratégies qui sont étendues pour permettre aux utilisateurs d'effectuer les tâches prescrites. Par exemple, si un développeur doit arrêter périodiquement une tâche, créez une stratégie qui autorise uniquement cette action particulière. L'exemple suivant permet uniquement à un utilisateur d'arrêter une tâche qui appartient à `untask_family`sur un cluster avec un Amazon Resource Name (ARN) spécifique. La référence à un ARN dans une condition est également un exemple d'utilisation des autorisations au niveau des ressources. Vous pouvez utiliser des autorisations au niveau des ressources pour spécifier la ressource à laquelle s'applique une action.

> **Note**
>
> Lorsque vous référencez un ARN dans une stratégie, utilisez le nouveau format ARN plus long. Pour de plus amples informations, veuillez consulterAmazon Resource Names (ARN) et IDdans leGuide du développeur Amazon Elastic Container Service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "ecs:StopTask"
      ],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"
        }
      },
      "Resource": [
        "arn:aws:ecs:<region>:<aws_account_id>:task-definition/<task_family>:*"
      ]
    }
  ]
}
```

## Laissez la ressource de cluster servir de limite administrative

Les politiques dont la portée est trop restreinte peuvent provoquer une prolifération des rôles et augmenter les frais administratifs. Plutôt que de créer des rôles qui sont limités à des tâches ou des services particuliers uniquement, créez des rôles qui sont étendus à des clusters et utilisez le cluster comme limite administrative principale.

## Isoler les utilisateurs finaux de l'API Amazon ECS en créant des pipelines automatisés

Vous pouvez limiter les actions que les utilisateurs peuvent utiliser en créant des pipelines qui empaquetent et déploient automatiquement des applications sur des clusters Amazon ECS. Cela délègue efficacement le travail de création, de mise à jour et de suppression de tâches au pipeline. Pour de plus amples informations, veuillez consulterDidacticiel : Déploiement standard Amazon ECS avec CodePipelinedans leAWS CodePipelineGuide de l'utilisateur.

## Utilisation des conditions de stratégie pour une couche de sécurité supplémentaire

Lorsque vous avez besoin d'une couche de sécurité supplémentaire, ajoutez une condition à votre stratégie. Cela peut être utile si vous effectuez une opération privilégiée ou lorsque vous devez restreindre l'ensemble des actions qui peuvent être effectuées sur des ressources particulières. L'exemple de stratégie suivant nécessite une autorisation multifacteur lors de la suppression d'un cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DeleteCluster"
      ],
      "Condition": {
        "Bool": {
          "aws:MultiFactorAuthPresent": "true"
        }
      },
    "Resource": ["*"]
    }
  ]
}
```

Les balises appliquées aux services sont propagées à toutes les tâches qui font partie de ce service. Pour cette raison, vous pouvez créer des rôles qui sont étendus aux ressources Amazon ECS avec des balises spécifiques. Dans la stratégie suivante, une entité IAM démarre et arrête toutes les tâches avec une clé de baliseDepartmentet une valeur de baliseAccounting.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:RunTask"
            ],
            "Resource": "arn:aws:ecs:*",
            "Condition": {
                "StringEquals": {"ecs:ResourceTag/Department": "Accounting"}
            }
        }
    ]
}
```

## Vérification périodique de l'accès aux API Amazon ECS

Un utilisateur peut changer de rôle. Après avoir modifié les rôles, les autorisations qui leur étaient précédemment accordées peuvent ne plus s'appliquer. Assurez-vous de vérifier qui a accès aux

API Amazon ECS et si cet accès est toujours justifié. Envisagez d'intégrer IAM à une solution de gestion du cycle de vie des utilisateurs qui révoque automatiquement l'accès lorsqu'un utilisateur quitte l'organisation. Pour de plus amples informations, veuillez consulterConsignes pour les audits de sécurité Amazon ECdans leRéférence générale Amazon Web Services.

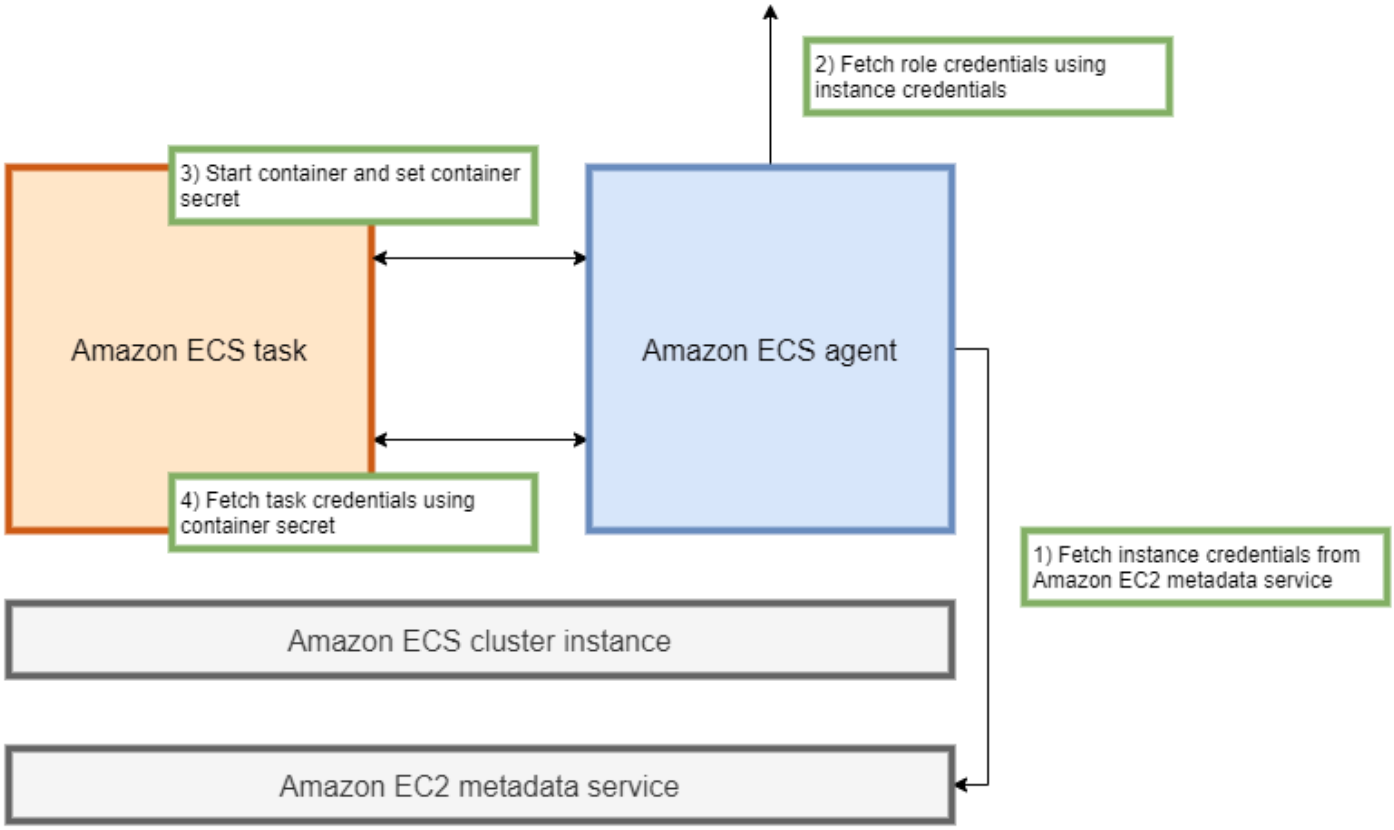
## Utilisation de rôles IAM avec les tâches Amazon ECS

Nous vous recommandons d'attribuer à une tâche un rôle IAM. Son rôle peut être distingué du rôle de l'instance Amazon EC2 sur laquelle elle s'exécute. L'attribution d'un rôle à chaque tâche est conforme au principe de l'accès le moins privilégié et permet un contrôle plus granulaire des actions et des ressources.

Lorsque vous attribuez des rôles IAM à une tâche, vous devez utiliser la stratégie d'approbation de suivi afin que chacune de vos tâches puisse assumer un rôle IAM différent de celui utilisé par votre instance EC2. De cette façon, votre tâche n'hérite pas du rôle de votre instance EC2.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Lorsque vous ajoutez un rôle de tâche à une définition de tâche, l'agent de conteneur Amazon ECS crée automatiquement un jeton avec un ID d'identification unique (par exemple,`12345678-90ab-cdef-1234-567890abcdef`) pour la tâche. Ce jeton et les informations d'identification du rôle sont ensuite ajoutés au cache interne de l'agent. L'agent remplit la variable d'environnement`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`dans le conteneur avec l'URI de l'ID d'identification (par exemple,`/v2/credentials/12345678-90ab-cdef-1234-567890abcdef`).




Vous pouvez récupérer manuellement les informations d'identification de rôle temporaires à partir d'un conteneur en ajoutant la variable d'environnement à l'adresse IP de l'agent de conteneur Amazon ECS et en exécutant la commande`curl`sur la chaîne résultante.

```
curl 192.0.2.0$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

La sortie attendue est la suivante :

```
{
 "RoleArn": "arn:aws:iam::123456789012:role/SSMTaskRole-SSMFargateTaskIAMRole-
DASWWSF2WGD6",
 "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
 "Token": "IQoJb3JpZ2luX2VjEEM/Example==",
 "Expiration": "2021-01-16T00:51:53Z"
}
```

Les versions plus récentes duAWSLes SDK récupèrent automatiquement ces informations d'identification à partir de la`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`environnement variable lors de la fabricationAWSAppels d'API.

La sortie inclut une paire de clés d'accès composée d'un ID de clé d'accès secrète et d'une clé secrète à laquelle votre application utilise pour accéderAWSAWS. Elle inclut également un jeton quiAWSutilise pour vérifier que les informations d'identification sont valides. Par défaut, les informations d'identification attribuées aux tâches à l'aide des rôles de tâche sont valides pendant six heures. Après cela, ils sont automatiquement tournés par l'agent de conteneur Amazon ECS.

## Rôle d'exécution de tâche

Le rôle d'exécution de tâche est utilisé pour accorder à l'agent de conteneur Amazon ECS l'autorisation d'appeler desAWSActions d'API en votre nom. Par exemple, lorsque vous utilisezAWS Fargate, Fargate a besoin d'un rôle IAM qui lui permet d'extraire des images d'Amazon ECR et d'écrire des journaux dans CloudWatch Logs. Un rôle IAM est également requis lorsqu'une tâche fait référence à un secret stocké dansAWS Secrets Manager, comme un secret d'extraction d'image.

> **Note**
>
> Si vous tirez des images en tant qu'utilisateur authentifié, vous êtes moins susceptible d'être affecté par les modifications apportées àLimites de taux de traction du Docker Hub. Pour plus d'informations, consultez .Authentification de registre privé pour les instances de.
> En utilisant Amazon ECR et Amazon ECR Public, vous pouvez éviter les limites imposées par Docker. Si vous extrayez des images depuis Amazon ECR, cela permet également de raccourcir les temps d'extraction du réseau et de réduire les changements de transfert de données lorsque le trafic quitte votre VPC.

> **Important**
>
> Lorsque vous utilisez Fargate, vous devez vous authentifier dans un registre d'images privées en utilisant`repositoryCredentials`. Il n'est pas possible de définir les variables d'environnement de l'agent de conteneur Amazon ECS`ECS_ENGINE_AUTH_TYPE`ou`ECS_ENGINE_AUTH_DATA`ou modifiez le`ecs.config`pour les tâches hébergées sur Fargate. Pour de plus amples informations, veuillez consulterAuthentification d'un registre privé pour.

# Rôle d'instance de conteneur Amazon EC2

L'agent de conteneur Amazon ECS est un conteneur qui s'exécute sur chaque instance Amazon EC2 dans un cluster Amazon ECS. Il est initialisé en dehors d'Amazon ECS à l'aide de l'outil`init`qui est disponible sur le système d'exploitation. Par conséquent, il ne peut pas être accordé d'autorisations via un rôle de tâche. Au lieu de cela, les autorisations doivent être attribuées aux instances Amazon EC2 sur lesquelles les agents s'exécutent. La liste des actions dans l'exemple`AmazonEC2ContainerServiceforEC2Role`doivent être accordées à la`ecsInstanceRole`. Si vous ne le faites pas, vos instances ne peuvent pas rejoindre le cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTags",
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:UpdateContainerInstancesState",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

Dans cette politique, le`ecr`and`logs`API permettent aux conteneurs qui s'exécutent sur vos instances d'extraire des images d'Amazon ECR et d'écrire des journaux sur Amazon CloudWatch. La .`ecs`permettent à l'agent d'enregistrer et de supprimer des instances et de communiquer avec le plan de contrôle Amazon ECS. Parmi ceux-ci, le`ecs:CreateCluster`est facultative.

## Rôles liés à un service

Vous pouvez utiliser le rôle lié à un service pour Amazon ECS afin d'accorder au service Amazon ECS l'autorisation d'appeler d'autres API de service en votre nom. Amazon ECS a besoin des autorisations pour créer et supprimer des interfaces réseau, enregistrer et annuler l'enregistrement des cibles auprès d'un groupe cible. Il a également besoin des autorisations nécessaires pour créer et supprimer des stratégies de mise à l'échelle. Ces autorisations sont accordées via le rôle lié à un service. Ce rôle est créé en votre nom la première fois que vous utilisez le service.

> **Note**
>
> Si vous supprimez par inadvertance le rôle lié à un service, vous pouvez le recréer. Pour obtenir des instructions, consultezCréation du rôle lié à un service.

## Recommendations

Nous vous recommandons d'effectuer les opérations suivantes lors de la configuration de vos rôles et stratégies IAM de tâche.

### Bloquer l'accès aux métadonnées Amazon EC2

Lorsque vous exécutez vos tâches sur des instances Amazon EC2, nous vous recommandons fortement de bloquer l'accès aux métadonnées Amazon EC2 afin d'empêcher vos conteneurs d'hériter du rôle attribué à ces instances. Si vos applications doivent appeler unAWS, utilisez les rôles IAM pour les tâches à la place.

Pour empêcher l'exécution de tâches danspontpour accéder aux métadonnées Amazon EC2, exécutez la commande suivante ou mettez à jour les données utilisateur de l'instance. Pour plus d'instructions sur la mise à jour des données utilisateur d'une instance, consultez cetteAWSArticle de support. Pour plus d'informations sur le mode pont de définition de tâche, consultezdéfinition de tâche mode réseau.

```
sudo yum install -y iptables-services; sudo iptables --insert FORWARD 1 --in-interface
 docker+ --destination 192.0.2.0/32 --jump DROP
```

Pour que cette modification persiste après un redémarrage, exécutez la commande suivante spécifique à votre Amazon Machine Image (AMI) :

- Amazon Linux 2

```
sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now
 iptables
```

- Amazon Linux

```
sudo service iptables save
```

Pour les tâches qui utilisentawsvpcmode réseau, définissez la variable d'environnementECS_AWSVPC_BLOCK_IMDSsurtruedans le/etc/ecs/ecs.configdans le fichier.

Vous devez définir leECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOSTvariable àfalsedans le fichier de configuration ecs-agent pour empêcher les conteneurs qui s'exécutent dans lehostd'accéder aux métadonnées Amazon EC2.

## Utiliser**awsvpc**Mode réseau

Utiliser le réseauawsvpcpour limiter le flux de trafic entre différentes tâches ou entre vos tâches et d'autres services qui s'exécutent au sein de votre Amazon VPC. L'ajout d'une couche de sécurité supplémentaire. La .awsvpcfournit l'isolation réseau au niveau des tâches pour les tâches qui s'exécutent sur Amazon EC2. C'est le mode par défaut surAWS Fargate. C'est le seul mode réseau que vous pouvez utiliser pour affecter un groupe de sécurité à des tâches.

## Utiliser IAM Access Advisor pour affiner les rôles

Nous vous recommandons de supprimer toutes les actions qui n'ont jamais été utilisées ou qui n'ont pas été utilisées depuis un certain temps. Cela empêche l'accès indésirable de se produire. Pour ce faire, consultez les résultats produits par IAM Access Advisor, puis supprimez les actions qui n'ont jamais été utilisées ou qui n'ont pas été utilisées récemment. Pour cela, vous pouvez effectuer cette opération en procédant comme suit.

Exécutez la commande suivante pour générer un rapport affichant les dernières informations d'accès pour la stratégie référencée :

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

Utilisation du`JobId`qui était dans la sortie pour exécuter la commande suivante. Une fois que vous avez terminé, vous pouvez afficher les résultats du rapport.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

Pour de plus amples informations, veuillez consulterIAM Access Advisor.

## ContrôleAWS CloudTrailpour les activités suspectes

Vous pouvez surveillerAWS CloudTrailpour toute activité suspecte. La plupart desAWSLes appels d'API sont journalisés àAWS CloudTrailcomme des événements. Ils sont analysés parAWS CloudTrailInsights, et vous êtes averti de tout comportement suspect associé à`write`Appels d'API. Cela peut inclure un pic dans le volume d'appel. Ces alertes incluent des informations telles que l'heure à laquelle l'activité inhabituelle s'est produite et l'ARN d'identité supérieure qui a contribué aux API.

Vous pouvez identifier les actions effectuées par des tâches avec un rôle IAM dansAWS CloudTrailen regardant le`userIdentity`Propriété. Dans l'exemple suivant, le kita`arn`comprend le nom du rôle assumé,`s3-write-go-bucket-role`, suivi du nom de la tâche,`7e9894e088ad416eb5cab92afExample`.

```
"userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA36C6WWEJ2YEXAMPLE:7e9894e088ad416eb5cab92afExample",
    "arn": "arn:aws:sts::123456789012:assumed-role/s3-write-go-bucket-
role/7e9894e088ad416eb5cab92afExample",
    ...
}
```

> **Note**
>
> Lorsque des tâches qui assument un rôle sont exécutées sur des instances de conteneur Amazon EC2, une demande est consignée par l'agent de conteneur Amazon ECS dans le journal d'audit de l'agent situé à une adresse dans la boîte de dialogue`/var/log/ecs/audit.log.YYYY-MM-DD-HH`. Pour de plus amples informations, veuillez consulterJournal des rôles IAM de la tâcheandJournalisation des événements Insights pour les journaux de suivi.

# Sécurité du réseau

La sécurité du réseau est un sujet large qui englobe plusieurs sous-thèmes. Il s'agit notamment du chiffrement en transit, de la segmentation et de l'isolement du réseau, du pare-feu, du routage du trafic et de l'observabilité.

## Chiffrement en transit

Le chiffrement du trafic réseau empêche les utilisateurs non autorisés d'intercepter et de lire des données lorsque ces données sont transmises sur un réseau. Avec Amazon ECS, le chiffrement réseau peut être implémenté de l'une des façons suivantes.

- Avec un maillage de service (TLS) :

  avecAWS App Mesh, vous pouvez configurer des connexions TLS entre les proxys Envoy qui sont déployés avec des points de terminaison maillés. Deux exemples sont les nœuds virtuels et les passerelles virtuelles. Les certificats TLS peuvent provenir deAWS Certificate Manager(ACM). Ou, il peut provenir de votre propre autorité de certification privée.

  - Activation de la sécurité de la couche de transport (TLS)
  - Activer le chiffrement du trafic entre les services dansAWS App Meshutilisation de certificats ACM ou de certificats fournis par le client
  - Procédure pas à pas TLS ACM
  - Procédure de présentation du fichier TLS
  - Envoy
- Utilisation des instances Nitro :

  Par défaut, le trafic est automatiquement chiffré entre les types d'instance Nitro suivants : C5n, P3dn, P3dn, P3dn, P3dn, P3dn, P3dn, R5n et R5n. Le trafic n'est pas chiffré lorsqu'il est acheminé via une passerelle de transit, un équilibreur de charge ou un intermédiaire similaire.

  - Chiffrement en transit
  - Nouveautés en matière de comptabilité à partir de 2019
  - Cette conférence de Re:inForce 2019
- Utilisation d'une extension SNI (Server Name Indication) avec un équilibreur de charge d'application :

L'Application Load Balancer (ALB) et l'Network Load Balancer (NLB) prennent en charge l'indication de nom de serveur (SNI). En utilisant SNI, vous pouvez placer plusieurs applications sécurisées derrière un seul écouteur. Pour cela, chacun a son propre certificat TLS. Nous vous recommandons de provisionner des certificats pour l'équilibreur de charge à l'aide deAWS Certificate Manager(ACM), puis ajoutez-les à la liste des certificats du processus d'écoute. La .AWSL'équilibreur de charge utilise un algorithme de sélection de certificat intelligent avec SNI. Si le nom d'hôte fourni par un client correspond à un seul certificat de la liste de certificats, l'équilibreur de charge choisit ce certificat. Si un nom d'hôte fourni par un client correspond à plusieurs certificats de la liste, l'équilibreur de charge sélectionne un certificat que le client peut prendre en charge. Par exemple, un certificat auto-signé ou un certificat généré par le biais du MCA.

  - SNI avec Application Load Balancer
  - SNI avec l'Network Load Balancer

- Chiffrement de bout en bout avec des certificats TLS :

  Cela implique le déploiement d'un certificat TLS avec la tâche. Il peut s'agir d'un certificat auto-signé ou d'un certificat d'une autorité de certification approuvée. Vous pouvez obtenir le certificat en référençant un secret pour le certificat. Sinon, vous pouvez choisir d'exécuter un conteneur qui émet une demande de signature de certificat (CSR) à ACM, puis monte le secret résultant sur un volume partagé.

  - Maintenir la sécurité de la couche de transport jusqu'à vos conteneurs à l'aide de l'Network Load Balancer avec Amazon ECS partie 1
  - Maintien de la sécurité de la couche de transport (TLS) jusqu'à la partie 2 de votre conteneur : Utilisation de AWS Private Certificate Authority

## Mise en réseau des tâches

Les recommandations suivantes tiennent compte du fonctionnement d'Amazon ECS. Amazon ECS n'utilise pas de réseau de superposition. Au lieu de cela, les tâches sont configurées pour fonctionner dans différents modes réseau. Par exemple, les tâches configurées pour utiliser`bridge`acquérir une adresse IP non routable à partir d'un réseau Docker qui s'exécute sur chaque hôte. Tâches configurées pour utiliser l'outil`awsvpc`Le mode réseau acquiert une adresse IP auprès du sous-réseau de l'hôte. Tâches configurées avec`host`utilisent l'interface réseau de l'hôte.`awsvpc`est le mode réseau préféré. C'est parce que c'est le seul mode que vous pouvez utiliser pour affecter des

groupes de sécurité aux tâches. C'est aussi le seul mode disponible pourAWS Fargatesur Amazon ECS.

### Groupes de sécurité pour les tâches

Nous vous recommandons de configurer vos tâches pour utiliser l'`awsvpc`Mode réseau. Après avoir configuré votre tâche pour utiliser ce mode, l'agent Amazon ECS provisionne automatiquement et attache une interface réseau Elastic (ENI) à la tâche. Lorsque l'ENI est provisionné, la tâche est inscrite dans unAWSgroupe de sécurité. Le groupe de sécurité agit en tant que pare-feu virtuel que vous pouvez utiliser pour contrôler le trafic entrant et sortant.

## Mesh de service et sécurité de la couche de transport mutuelle (MTL)

Vous pouvez utiliser un maillage de service tel queAWS App Meshpour contrôler le trafic réseau. Par défaut, un nœud virtuel ne peut communiquer qu'avec ses backends de service configurés, tels que les services virtuels avec lesquels le nœud virtuel communiquera. Si un nœud virtuel doit communiquer avec un service en dehors du maillage, vous pouvez utiliser l'outil`ALLOW_ALL`ou en créant un nœud virtuel à l'intérieur du maillage pour le service externe. Pour de plus amples informations, veuillez consulterProcédure à suivre pour Kubernetes Egress.

App Mesh vous donne également la possibilité d'utiliser Mutual Transport Layer Security (MTL) où le client et le serveur sont mutuellement authentifiés à l'aide de certificats. Les communications ultérieures entre le client et le serveur sont ensuite cryptées à l'aide de TLS. En exigeant des MTL entre services dans un maillage, vous pouvez vérifier que le trafic provient d'une source fiable. Pour de plus amples informations, consultez les rubriques suivantes :

- Authentification MTL
- Procédure pas à pas du Service de découverte secrète (SDS) MTLS
- Procédure pas à pas

## AWS PrivateLink

AWS PrivateLinkest une technologie de mise en réseau qui vous permet de créer des points de terminaison privés pour différentsAWS, y compris Amazon ECS. Les points de terminaison sont requis dans les environnements sandbox où il n'y a pas de Internet Gateway (IGW) connecté au VPC Amazon et aucune autre route vers Internet. UtiliserAWS PrivateLinkgarantit que les appels vers le service Amazon ECS restent dans le VPC Amazon et ne traversent pas Internet. Pour savoir

comment créerAWS PrivateLinkpour Amazon ECS et d'autres services associés, consultezInterface Amazon ECS Endpoints de terminaison Amazon VPC.

> **Important**
>
> AWS Fargatene nécessitent pas deAWS PrivateLinkpour Amazon ECS.

Amazon ECR et Amazon ECS prennent toutes les deux en charge les politiques relatives aux terminaux. Ces stratégies vous permettent d'affiner l'accès aux API d'un service. Par exemple, vous pouvez créer une stratégie de point de terminaison pour Amazon ECR qui permet uniquement d'envoyer des images dans les registres, en particulierAWS. Une telle stratégie pourrait être utilisée pour empêcher l'exfiltration des données via des images de conteneur tout en permettant aux utilisateurs de pousser vers des registres Amazon ECR autorisés. Pour de plus amples informations, veuillez consulterUtilisation des stratégies de point de terminaison de VPC.

La stratégie suivante autorise tous lesAWSde votre compte pour effectuer toutes les actions contre vos référentiels Amazon ECR uniquement :

```
{
  "Statement": [
    {
      "Sid": "LimitECRAccess",
      "Principal": "*",
      "Action": "*",
      "Effect": "Allow",
      "Resource": "arn:aws:ecr:region:your_account_id:repository/*"
    },
  ]
}
```

Vous pouvez l'améliorer davantage en définissant une condition qui utilise le nouveau`PrincipalOrgID`Propriété. Cela empêche de pousser et de tirer des images par un principal IAM qui ne fait pas partie de votreAWS Organizations. Pour de plus amples informations, veuillez consulteraws:PrincipalOrgID.

Nous avons recommandé d'appliquer la même politique à la fois`com.amazonaws.region.ecr.dkr`et l'`com.amazonaws.region.ecr.api`Points de terminaison .

# Paramètres d'agent de conteneur Amazon ECS

Le fichier de configuration de l'agent de conteneur Amazon ECS inclut plusieurs variables d'environnement liées à la sécurité du réseau.`ECS_AWSVPC_BLOCK_IMDS`and`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST`sont utilisés pour bloquer l'accès d'une tâche aux métadonnées Amazon EC2.`HTTP_PROXY`est utilisé pour configurer l'agent pour acheminer via un proxy HTTP pour se connecter à Internet. Pour obtenir des instructions sur la configuration de l'agent et du moteur d'exécution Docker afin qu'ils acheminent à travers un proxy, consultezConfiguration d'un proxy HTTP.

> **Important**
>
> Ces paramètres ne sont pas disponibles lorsque vous utilisezAWS Fargate.

## Recommendations

Nous vous recommandons d'effectuer les opérations suivantes lors de la configuration de votre VPC Amazon, de vos équilibreurs de charge et de votre réseau.

### Utiliser le chiffrement réseau, le cas échéant

Vous devez utiliser le chiffrement réseau le cas échéant. Certains programmes de conformité, tels que PCI DSS, exigent que vous cryptiez les données en transit si les données contiennent des données de titulaire de carte. Si votre charge de travail a des exigences similaires, configurez le chiffrement réseau.

Les navigateurs modernes avertissent les utilisateurs lors de la connexion à des sites non sécurisés. Si votre service est dirigé par un équilibreur de charge public, utilisez TLS/SSL pour chiffrer le trafic du navigateur du client vers l'équilibreur de charge et le chiffrer à nouveau vers le backend si nécessaire.

### Utiliser**`awsvpc`**mode réseau et groupes de sécurité lorsque vous devez contrôler le trafic entre les tâches ou entre les tâches et d'autres ressources réseau

Vous devez utiliser`awsvpc`le mode réseau et les groupes de sécurité lorsque vous devez contrôler le trafic entre les tâches et entre les tâches et d'autres ressources réseau. Si votre service est derrière un ALB, utilisez des groupes de sécurité pour autoriser uniquement le trafic entrant provenant

d'autres ressources réseau utilisant le même groupe de sécurité que votre ALB. Si votre application se trouve derrière une NLB, configurez le groupe de sécurité de la tâche de manière à autoriser uniquement le trafic entrant provenant de la plage CIDR Amazon VPC et les adresses IP statiques affectées à l'NLB.

Les groupes de sécurité doivent également être utilisés pour contrôler le trafic entre les tâches et d'autres ressources au sein d'Amazon VPC, telles que les bases de données Amazon RDS.

## Créer des clusters dans des VPC Amazon distincts lorsque le trafic réseau doit être strictement isolé

Vous devez créer des clusters dans des VPC Amazon distincts lorsque le trafic réseau doit être strictement isolé. Évitez d'exécuter des charges de travail qui ont des exigences de sécurité strictes sur les clusters dont les charges de travail n'ont pas à respecter ces exigences. Lorsque l'isolement strict du réseau est obligatoire, créez des clusters dans des VPC Amazon distincts et exposez de manière sélective les services à d'autres VPC Amazon à l'aide de points de terminaison Amazon VPC. Pour de plus amples informations, veuillez consulterPoints de terminaison Amazon VPC.

## ConfigurationAWS PrivateLinkpoints de terminaison lorsque cela est justifié

Vous devez configurerAWS PrivateLinkpoints de terminaison lorsque cela est justifié. Si votre stratégie de sécurité vous empêche d'attacher une Internet Gateway (IGW) à vos VPC Amazon, configurezAWS PrivateLinkpour Amazon ECS et d'autres services tels qu'Amazon ECR,AWS Secrets Manageret Amazon CloudWatch.

## Utilisez les journaux de flux Amazon VPC pour analyser le trafic à destination et en provenance des tâches de longue durée

Vous devez utiliser les journaux de flux Amazon VPC pour analyser le trafic à destination et en provenance des tâches de longue durée. Tâches qui utilisent`awsvpc`mode réseau obtenir leur propre ENI. Pour ce faire, vous pouvez surveiller le trafic qui va vers et depuis des tâches individuelles à l'aide des journaux de flux Amazon VPC. Une mise à jour récente des journaux de flux Amazon VPC (v3) enrichit les journaux avec des métadonnées de trafic, y compris l'ID vpc, l'ID de sous-réseau et l'ID d'instance. Ces métadonnées peuvent être utilisées pour limiter une enquête. Pour de plus amples informations, veuillez consulterJournaux de flux Amazon VPC.

> **Note**
>
> En raison de la nature temporaire des conteneurs, les journaux de flux peuvent ne pas toujours être un moyen efficace d'analyser les schémas de trafic entre différents conteneurs ou conteneurs et d'autres ressources réseau.

# Gestion des secrets

Les secrets, tels que les clés API et les informations d'identification de base de données, sont fréquemment utilisés par les applications pour accéder à d'autres systèmes. Ils se composent souvent d'un nom d'utilisateur et d'un mot de passe, d'un certificat ou d'une clé d'API. L'accès à ces secrets doit être limité aux entités IAM spécifiques qui utilisent IAM et injectées dans des conteneurs au moment de l'exécution.

Les secrets peuvent être injectés de manière transparente dans les conteneurs deAWS Secrets Manageret Amazon EC2 Systems Manager. Ces secrets peuvent être référencés dans votre tâche comme l'un des éléments suivants.

1. Ils sont référencés en tant que variables d'environnement qui utilisent la méthode`secrets`définition de conteneur.
2. Ils sont référencés comme`secretOptions`si votre plateforme de journalisation nécessite une authentification. Pour de plus amples informations, veuillez consulterOptions de configuration de journalisation.
3. Ils sont référencés comme des secrets tirés par des images qui utilisent le`repositoryCredentials`paramètre de définition de conteneur si le registre d'où le conteneur est extrait nécessite une authentification. Utilisez cette méthode pour extraire des images depuis Docker Hub. Pour de plus amples informations, veuillez consulterAuthentification d'un registre privé pour.

## Recommendations

Nous vous recommandons de faire ce qui suit lors de la mise en place de la gestion des secrets.

## UtiliserAWS Secrets Managerou Amazon EC2 Systems Manager pour stocker des matériaux secrets

Vous devez stocker en toute sécurité les clés d'API, les informations d'identification de base de données et d'autres documents secrets dansAWS Secrets Managerou en tant que paramètre chiffré dans le magasin de paramètres Amazon EC2 Systems Manager. Ces services sont similaires car ils sont tous deux des magasins de valeur clé gérés qui utilisentAWS KMSpour chiffrer les données sensibles.AWS Secrets Manager, cependant, inclut également la possibilité de faire pivoter automatiquement des secrets, de générer des secrets aléatoires et de partager des secrets dansAWS. Si vous considérez ces fonctionnalités importantes, utilisezAWS Secrets Managersinon utilisez des paramètres chiffrés.

> **Note**
>
> Tâches qui font référence à un secret deAWS Secrets Managerou le magasin de paramètres Amazon EC2 Systems Manager nécessitent unRôle d'exécution de tâcheavec une politique qui accorde à Amazon ECS l'accès au secret désiré et, le cas échéant, auAWS KMSutilisée pour chiffrer et déchiffrer ce secret.

> **Important**
>
> Les secrets référencés dans les tâches ne sont pas tournés automatiquement. Si votre secret change, vous devez forcer un nouveau déploiement ou lancer une nouvelle tâche pour récupérer la dernière valeur secrète. Pour de plus amples informations, consultez les rubriques suivantes :
>
> - AWS Secrets Manager : Injecter des données en tant que variables d'environnement
> - Magasin de paramètres Amazon EC2 Systems Manager : Injecter des données en tant que variables d'environnement

## Récupération de données à partir d'un compartiment Amazon S3 chiffré

Parce que la valeur des variables d'environnement peut fuir par inadvertance dans les journaux et sont révélées lors de l'exécution`docker inspect`, vous devez stocker des secrets dans un compartiment Amazon S3 chiffré et utiliser des rôles de tâche pour restreindre l'accès à ces secrets. Lorsque vous effectuez cette opération, votre application doit être écrite pour lire le secret du

compartiment Amazon S3. Pour obtenir des instructions, consultezDéfinition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

### Monter le secret sur un volume à l'aide d'un conteneur side-car

Étant donné qu'il existe un risque élevé de fuite de données avec des variables d'environnement, vous devez exécuter un conteneur sidecar qui lit vos secrets deAWS Secrets Manageret les écrire sur un volume partagé. Ce conteneur peut s'exécuter et quitter avant le conteneur d'application en utilisantCommande de conteneur Amazon ECS. Lorsque vous faites cela, le conteneur d'application est ensuite monter le volume où le secret a été écrit. Comme la méthode du compartiment Amazon S3, votre application doit être écrite pour lire le secret du volume partagé. Étant donné que le volume est étendu à la tâche, le volume est automatiquement supprimé après l'arrêt de la tâche. Pour obtenir un exemple de conteneur side-car, consultez la sectioninjecteur aws-sécrét-side-car-.

> **Note**
>
> Sur Amazon EC2, le volume sur lequel le secret est écrit peut être chiffré avec unAWS KMSClé gérée par le client. SurAWS Fargate, le stockage en volume est automatiquement chiffré à l'aide d'une clé gérée par le service.

## Ressources supplémentaires

- Transmettre des secrets aux conteneurs dans une tâche Amazon ECS
- Chambreest un wrapper pour stocker des secrets dans le magasin de paramètres Amazon EC2 Systems Manager

# Compliance

Votre responsabilité de conformité lors de l'utilisation d'Amazon ECS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables.

AWSfournit les ressources suivantes pour faciliter la conformité :

- Guides de démarrage rapide sur la sécurité et la conformité : Ces guides de déploiement proposent des considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence centrés sur la sécurité et la conformité dansAWS.

- Livre blanc Architecting for HIPAA Security and Compliance Livre blanc : Ce livre blanc décrit comment les entreprises peuvent utiliserAWSpour créer des applications conformes à la loi HIPAA.
- AWSServices visés par le programme de conformité : La liste contient le kitAWSServices dans le champ d'application de certains programmes de conformité. Pour de plus amples informations, veuillez consulterAWSProgrammes de conformité.

## Normes de sécurité des données de l'industrie des cartes de paiement (PCI DSS)

Il est important que vous compreniez le flux complet des données des titulaires de carte (CHD) dans l'environnement lorsque vous respectez PCI DSS. Le flux CHD détermine l'applicabilité du DSS PCI, définit les limites et les composantes d'un environnement de données de titulaire de carte (CDE) et, par conséquent, la portée d'une évaluation PCI DSS. La détermination précise de la portée de la norme PCI DSS est essentielle à la définition de la posture de sécurité et, en fin de compte, à une évaluation réussie. Les clients doivent disposer d'une procédure de détermination de la portée qui assure son exhaustivité et détecte les changements ou les écarts par rapport à la portée.

La nature temporaire des applications conteneurisées offre des complexités supplémentaires lors de l'audit des configurations. Par conséquent, les clients doivent rester conscients de tous les paramètres de configuration des conteneurs pour s'assurer que les exigences de conformité sont respectées à toutes les phases du cycle de vie d'un conteneur.

Pour plus d'informations sur la conformité PCI DSS sur Amazon ECS, reportez-vous aux livres blancs suivants.

- Architecting on Amazon ECS pour la conformité PCI DSS
- Architecture pour la définition et la segmentation PCI DSS sur AWS

## HIPAA (Health Insurance Portability and Accountability Act)

L'utilisation d'Amazon ECS avec des charges de travail qui traitent les informations d'intégrité protégées (PHI) ne nécessite aucune configuration supplémentaire. Amazon ECS agit comme un service d'orchestration qui coordonne le lancement de conteneurs sur Amazon EC2. Il ne fonctionne pas avec ou sur les données de la charge de travail orchestrée. Conformément à la réglementation HIPAA et à laAWSAddendum Business Associate, PHI doit être chiffré en transit et au repos lorsqu'il est accessible par des conteneurs lancés avec Amazon ECS.

Différents mécanismes de cryptage au repos sont disponibles avec chaqueAWS, comme Amazon S3, Amazon EBS etAWS KMS. Vous pouvez déployer un réseau de superposition (tel que VNS3 ou Weave Net) pour assurer le cryptage complet des données PHI transférées entre les conteneurs ou pour fournir une couche de chiffrement redondante. La journalisation complète doit également être activée et tous les journaux des conteneurs doivent être dirigés vers Amazon CloudWatch. Pour de plus amples informations, veuillez consulterArchitecting for HIPAA Security and Compliance.

### Recommendations

Vous devez engager tôt les responsables du programme de conformité au sein de votre entreprise et utiliser leAWSModèle de responsabilité partagéepour déterminer la propriété du contrôle de la conformité pour réussir avec les programmes de conformité pertinents.

## Journalisation et surveillance

La journalisation et la surveillance sont un aspect important du maintien de la fiabilité, de la disponibilité et des performances d'Amazon ECS et de votreAWSsolutions.AWSfournit plusieurs outils pour surveiller vos ressources Amazon ECS et répondre aux éventuels incidents.

- Alarmes Amazon CloudWatch
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CloudTrail Journaux

Vous pouvez configurer les conteneurs de vos tâches afin qu'ils envoient des informations de journaux aux journaux Amazon CloudWatch Logs. Si vous utilisez leAWS FargatePour vos tâches, vous pouvez afficher les journaux à partir de vos conteneurs. Si vous utilisez le type de lancement Amazon EC2, vous pouvez afficher différents journaux de vos conteneurs dans un emplacement pratique. Cela empêche également que vos journaux de conteneur occupent de l'espace disque sur vos instances de conteneur.

Pour plus d'informations sur Amazon CloudWatch Logs, consultezSurveillance des journaux des instances Amazon EC2dans leGuide de l'utilisateur Amazon CloudWatch.. Pour obtenir des instructions sur l'envoi de journaux de conteneur à partir de vos tâches vers Amazon CloudWatch Logs, consultezUtilisation du kit`awslogs`pilote de journal.

## Enregistrement des conteneurs avec Fluent Bit

AWSfournit une image Fluent Bit avec des plug-ins pour Amazon CloudWatch Logs et Amazon Kinesis Data Firehose. Cette image permet d'acheminer les journaux vers les destinations Amazon CloudWatch et Amazon Kinesis Data Firehose (qui incluent Amazon S3, Amazon Elasticsearch Service et Amazon Redshift). Nous vous recommandons d'utiliser Fluent Bit comme routeur de journal car il a un taux d'utilisation des ressources inférieur à Fluentd. Pour de plus amples informations, veuillez consulterAmazon CloudWatch Logs pour Fluent BitandAmazon Kinesis Data Firehose pour Fluent Bit.

La .AWSpour Fluent Bit est disponible sur :

- Galerie publique Amazon ECR sur Amazon ECR
- Dépôt Amazon ECR(dans la plupart des régions de haute disponibilité)
- Docker Hub

L'exemple suivant montre la syntaxe à utiliser pour l'interface de ligne de commande Docker.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:tag
```

Par exemple, vous pouvez extraire le dernierAWSpour l'image Fluent Bit à l'aide de cette commande Docker CLI :

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:latest
```

Consultez également les billets de blog suivants pour trouver d'autres informations sur Fluent Bit et les fonctionnalités connexes :

- Bit Fluent pour Amazon EKS sur AWS Fargate
- Journalisation centralisée des conteneurs avec le bit Fluent
- Création d'un agrégateur de solution de journal évolutif avecAWS Fargate, Fluentd et Amazon Kinesis Data Firehose

## Routage personnalisé des journaux - FireLens pour Amazon ECS

Avec FireLens pour Amazon ECS, vous pouvez utiliser des paramètres de définition de tâche pour acheminer des journaux vers unAWSservice ouAWSDestination Partner Network (APN) pour le

stockage et l'analyse des journaux. FireLens fonctionne avec Fluentd et Fluent Bit. Nous fournissons leAWSpour l'image Fluent Bit. Vous pouvez également utiliser votre propre image Fluentd ou Fluent Bit.

Vous devez tenir compte des conditions et considérations suivantes lors de l'utilisation de FireLens pour Amazon ECS :

- FireLens pour Amazon ECS est pris en charge pour les tâches hébergées à la fois surAWS Fargateet Amazon EC2.
- FireLens pour Amazon ECS est pris en charge dansAWS CloudFormationModèles. Pour de plus amples informations, veuillez consulterAWS::ECS::TaskDefinition FirelensConfigurationdans leAWS CloudFormationGuide de l'utilisateur.
- Pour les tâches qui utilisent le`bridge`En mode réseau, les conteneurs avec la configuration FireLens doivent démarrer avant que n'importe lequel des conteneurs d'application qui l'utilisent démarre. Pour contrôler l'ordre dans lequel vos conteneurs commencent, utilisez des conditions de dépendance dans la définition de tâche. Pour de plus amples informations, veuillez consulterDépendances de.

# Sécurité AWS Fargate

Nous vous recommandons de tenir compte des bonnes pratiques suivantes lorsque vous utilisezAWS Fargate.

## UtiliserAWS KMSpour chiffrer le stockage éphémère

Vous devriez avoir votre stockage éphémère crypté parAWS KMS. Pour les tâches Amazon ECS hébergées surAWS FargateUtilisation de la version de plateforme`1.4.0`ou une version ultérieure, chaque tâche reçoit 20 Go de stockage éphémère. La quantité de stockage n'est pas réglable. Pour de telles tâches lancées le 28 mai 2020 ou plus tard, le stockage éphémère est chiffré à l'aide d'un algorithme de chiffrement AES-256 à l'aide d'une clé de chiffrement gérée parAWS Fargate.

Exemple : Lancement d'une tâche Amazon ECS surAWS Fargatela version 1.4.0 de la plate-forme avec chiffrement de stockage éphémère

La commande suivante lancera une tâche Amazon ECS surAWS Fargateversion 1.4 de la plateforme. Étant donné que cette tâche est lancée dans le cadre du cluster Amazon ECS, elle utilise les 20 Go de stockage éphémère qui sont automatiquement cryptés.

```
aws ecs run-task --cluster clustername \
  --task-definition taskdefinition:version \
  --count 1
  --launch-type "FARGATE" \
  --platform-version 1.4.0 \
  --network-configuration
 "awsvpcConfiguration={subnets=[subnetid],securityGroups=[securitygroupid]}" \
  --region region
```

## Capacité SYS_PTRACE pour le suivi du système du noyau

La configuration par défaut des fonctionnalités Linux ajoutées ou supprimées de votre conteneur est fournie par Docker. Pour plus d'informations sur les fonctionnalités disponibles, consultezPrivilège d'exécution et fonctionnalités Linuxdans leExécution du Docker.

Tâches lancées surAWS Fargateprend uniquement en charge l'ajout de`SYS_PTRACE`Capacité du noyau.

Reportez-vous au didacticiel vidéo ci-dessous qui montre comment utiliser cette fonctionnalité via le système SysdigFalco.

#ContainersFromTheCouch - Dépannage de votreAWS FargateTâche utilisant la fonctionnalité SYS_PTRACE

Le code discuté dans la vidéo précédente peut être trouvé sur GitHubIci.

# Sécurité des tâches et des conteneurs

Vous devriez considérer l'image de conteneur comme la première ligne de défense contre une attaque. Une image peu sécurisée et mal construite peut permettre à un attaquant d'échapper aux limites du conteneur et d'accéder à l'hôte. Vous devez faire ce qui suit pour atténuer le risque que cela se produise.

## Recommendations

Nous vous recommandons d'effectuer les opérations suivantes lors de la configuration de vos tâches et conteneurs.

## Créer un minimum ou utiliser des images sans distroless

Commencez par supprimer tous les fichiers binaires superflus de l'image du conteneur. Si vous utilisez une image inconnue de Docker Hub, inspectez l'image pour faire référence au contenu de chacune des couches du conteneur. Vous pouvez utiliser une application telle quePlongéePour

Vous pouvez également utiliserdistrolessqui incluent uniquement votre application et ses dépendances d'exécution. Ils ne contiennent pas de gestionnaires de paquets ou d'interpréteurs de commandes. Les images sans distroless améliorent le « signal au bruit des scanners et réduisent le fardeau d'établir la provenance à ce dont vous avez besoin ». Pour plus d'informations, consultez la Documentation GitHub surdistroless.

Docker dispose d'un mécanisme permettant de créer des images à partir d'une image minimale réservée appeléegratter. Informations Formore, consultezCréation d'une image parent simple à l'aide degratterdans la documentation Docker. Avec des langages comme Go, vous pouvez créer un binaire lié statique et le référencer dans votre Dockerfile. L'exemple suivant montre comment effectuer cette opération.

```
############################
# STEP 1 build executable binary
############################
FROM golang:alpine AS builder
# Install git.
# Git is required for fetching the dependencies.
RUN apk update && apk add --no-cache git
WORKDIR $GOPATH/src/mypackage/myapp/
COPY . .
# Fetch dependencies.
# Using go get.
RUN go get -d -v
# Build the binary.
RUN go build -o /go/bin/hello
############################
# STEP 2 build a small image
############################
FROM scratch
# Copy our static executable.
COPY --from=builder /go/bin/hello /go/bin/hello
# Run the hello binary.
ENTRYPOINT ["/go/bin/hello"]
```

```
This creates a container image that consists of your application and nothing else,
 making it extremely secure.
```

L'exemple précédent est également un exemple de construction en plusieurs étapes. Ces types de builds sont attrayants du point de vue de la sécurité, car vous pouvez les utiliser pour minimiser la taille de l'image finale envoyée dans votre registre de conteneurs. Les images de conteneur dépourvues d'outils de construction et d'autres fichiers binaires externes améliorent votre posture de sécurité en réduisant la surface d'attaque de l'image. Pour plus d'informations sur les builds multi-étapes, consultezcréation de builds en plusieurs étapes.

## Analyser vos images à la recherche de vulnérabilités

À l'instar de leurs homologues de machines virtuelles, les images de conteneur peuvent contenir des binaires et des bibliothèques d'applications présentant des vulnérabilités ou développer des vulnérabilités au fil du temps. La meilleure façon de vous protéger contre les exploits est de numériser régulièrement vos images à l'aide d'un scanner d'images. Les images stockées dans Amazon ECR peuvent être numérisées en mode push ou à la demande (une fois toutes les 24 heures). Amazon ECR utilise actuellementClair, une solution open source de numérisation d'images. Une fois une image numérisée, les résultats sont enregistrés dans le flux d'événements Amazon ECR dans Amazon EventBridge. Vous pouvez également voir les résultats d'une analyse à partir de la console Amazon ECR ou en appelant leDescribeImageScanFactsAPI. Images avec un`HIGH`ou`CRITICAL`doit être supprimée ou reconstruite. Si une image déployée développe une vulnérabilité, elle doit être remplacée dès que possible.

Docker Desktop Edge version 2.3.6.0ou version ultérieure peutscanimages locales. Les scans sont alimentés parSnyk, un service de sécurité des applications. Lorsque des vulnérabilités sont découvertes, Snyk identifie les couches et les dépendances avec la vulnérabilité dans Dockerfile. Il recommande également des alternatives sûres comme l'utilisation d'une image de base plus mince avec moins de vulnérabilités ou la mise à niveau d'un paquet particulier vers une version plus récente. En utilisant Docker scan, les développeurs peuvent résoudre des problèmes de sécurité potentiels avant de transférer leurs images dans le registre.

- Automatisation de la conformité des images à l'aide d'Amazon ECR etAWS Security Hubexplique comment afficher les informations de vulnérabilité d'Amazon ECR dansAWS Security Hubet automatisez la correction en bloquant l'accès aux images vulnérables.

## Supprimer les autorisations spéciales de vos images

Les drapeaux des droits d'accès`setuid`and`setgid`permettent d'exécuter un exécutable avec les autorisations du propriétaire ou du groupe de l'exécutable. Supprimez tous les fichiers binaires dotés de ces droits d'accès de votre image car ces fichiers binaires peuvent être utilisés pour augmenter les privilèges. Envisagez de supprimer tous les shells et utilitaires commencand`curl`qui peuvent être utilisés à des fins malveillantes. Vous pouvez trouver les fichiers avec`setuid`and`setgid`À l'aide de la commande suivante.

```
find / -perm /6000 -type f -exec ls -ld {} \;
```

Pour supprimer ces autorisations spéciales de ces fichiers, ajoutez la directive suivante à votre image de conteneur.

```
RUN find / -xdev -perm /6000 -type f -exec chmod a-s {} \; || true
```

## Créer un ensemble d'images sélectionnées

Plutôt que de permettre aux développeurs de créer leurs propres images, créez un ensemble d'images vérifiées pour les différentes piles d'applications de votre organisation. Ce faisant, les développeurs peuvent renoncer à apprendre à composer Dockerfiles et se concentrer sur l'écriture de code. Au fur et à mesure que les modifications sont fusionnées dans votre base de code, un pipeline CI/CD peut compiler automatiquement l'actif, puis le stocker dans un référentiel d'artefacts. Enfin, copiez l'artefact dans l'image appropriée avant de le pousser dans un registre Docker tel qu'Amazon ECR. Au moins, vous devriez créer un ensemble d'images de base que les développeurs de chapeau peuvent créer leurs propres Dockerfiles à partir de. Évitez de tirer des images depuis Docker Hub. Vous ne savez pas toujours ce qu'il y a dans l'image et environ un cinquième des 1000 premières images présentent des vulnérabilités. Vous trouverez une liste de ces images et de leurs vulnérabilités à l'adressehttps://vulnerablecontainers.org/.

## Analyser les paquets d'applications et les bibliothèques pour détecter les vulnérabilités

L'utilisation de bibliothèques open source est maintenant courante. Comme pour les systèmes d'exploitation et les packages de systèmes d'exploitation, ces bibliothèques peuvent présenter des vulnérabilités. Dans le cadre du cycle de développement, ces bibliothèques doivent être analysées et mises à jour lorsque des vulnérabilités critiques sont détectées.

Docker Desktop effectue des analyses locales à l'aide de Snyk. Il peut également être utilisé pour trouver des vulnérabilités et des problèmes potentiels de licence dans les bibliothèques open source. Il peut être intégré directement dans les flux de travail des développeurs, ce qui vous permet d'atténuer les risques posés par les bibliothèques open source. Pour de plus amples informations, consultez les rubriques suivantes :

- Outils de sécurité des applications Open Sourceinclut une liste d'outils permettant de détecter les vulnérabilités dans les applications.
- Feuille de triche de numérisation Docker

## Effectuer une analyse de code statique

Vous devez effectuer une analyse de code statique avant de créer une image de conteneur. Il est exécuté avec votre code source et est utilisé pour identifier les erreurs de codage et le code qui pourraient être exploités par un acteur malveillant, comme les injections de défauts. SonarQubeest une option populaire pour les tests de sécurité des applications statiques (SAST), avec la prise en charge d'une variété de langages de programmation différents.

## Exécuter des conteneurs en tant qu'utilisateur non root

Vous devez exécuter des conteneurs en tant qu'utilisateur non-racine. Par défaut, les conteneurs s'exécutent en tant que`root`à moins que l'utilisateur`USER`est incluse dans votre Dockerfile. Les fonctionnalités Linux par défaut attribuées par Docker restreignent les actions qui peuvent être exécutées en tant que`root`, mais seulement marginalement. Par exemple, un conteneur s'exécutant en tant que`root`n'est toujours pas autorisé à accéder aux appareils.

Dans le cadre de votre pipeline CI/CD, vous devriez pelucher Dockerfiles pour rechercher le`USER`et échoue la construction si elle est manquante. Pour de plus amples informations, consultez les rubriques suivantes :

- Dockerfileest un outil open source de RedHat qui peut être utilisé pour vérifier si le fichier est conforme aux meilleures pratiques.
- Hadolintest un autre outil permettant de créer des images Docker conformes aux meilleures pratiques.

## Utiliser un système de fichiers racine en lecture seule

Vous devez utiliser un système de fichiers racine en lecture seule. Le système de fichiers racine d'un conteneur est accessible en écriture par défaut. Lorsque vous configurez un conteneur avec unR0(lecture seule), il vous oblige à définir explicitement où les données peuvent être conservées. Cela réduit votre surface d'attaque car le système de fichiers du conteneur ne peut pas être écrit à moins que des autorisations ne soient spécifiquement accordées.

> **Note**
>
> Le fait d'avoir un système de fichiers racine en lecture seule peut provoquer des problèmes avec certains packages du système d'exploitation qui s'attendent à pouvoir écrire sur le système de fichiers. Si vous envisagez d'utiliser des systèmes de fichiers racine en lecture seule, testez soigneusement au préalable.

## Configurer les tâches avec des limites de CPU et de mémoire (Amazon EC2)

Vous devez configurer les tâches avec des limites de CPU et de mémoire pour minimiser les risques suivants. Les limites de ressources d'une tâche définissent une limite supérieure pour la quantité de CPU et de mémoire qui peut être réservée par tous les conteneurs d'une tâche. Si aucune limite n'est définie, les tâches ont accès au processeur et à la mémoire de l'hôte. Cela peut provoquer des problèmes dans lesquels les tâches déployées sur un hôte partagé peuvent affamer d'autres tâches de ressources système.

> **Note**
>
> Amazon ECS surAWS Fargatenécessitent que vous spécifiez des limites de CPU et de mémoire car il utilise ces valeurs à des fins de facturation. Une tâche qui bloque toutes les ressources système n'est pas un problème pour Amazon ECS Fargate, car chaque tâche est exécutée sur sa propre instance dédiée. Si vous ne spécifiez pas de limite de mémoire, Amazon ECS alloue au moins 4 Mo à chaque conteneur. De même, si aucune limite de CPU n'est définie pour la tâche, l'agent de conteneur Amazon ECS lui attribue un minimum de 2 CPU.

## Utiliser des balises immuables avec Amazon ECR

Avec Amazon ECR, vous pouvez et devez utiliser configurer des images avec des balises immuables. Cela empêche d'envoyer une version modifiée ou mise à jour d'une image dans votre référentiel d'images avec une balise identique. Cela protège contre un attaquant qui pousse une version compromise d'une image sur votre image avec la même balise. En utilisant des balises immuables, vous vous forcez effectivement à pousser une nouvelle image avec une balise différente pour chaque changement.

## Évitez d'utiliser des conteneurs comme privilégiés (Amazon EC2)

Vous devez éviter d'exécuter des conteneurs comme privilégiés. En arrière-plan, les conteneurs s'exécutent en tant que`privileged`sont exécutés avec des privilèges étendus sur l'hôte. Cela signifie que le conteneur hérite de toutes les fonctionnalités Linux assignées à`root`sur l'hôte. Son utilisation devrait être strictement restreinte ou interdite. Nous vous recommandons de définir la variable d'environnement Amazon ECS Container Agent`ECS_DISABLE_PRIVILEGED`sur`true`pour empêcher les conteneurs de s'exécuter en tant que`privileged`sur des hôtes particuliers si`privileged`n'est pas nécessaire. Vous pouvez également utiliserAWS Lambdapour analyser vos définitions de tâches en vue de l'utilisation de l'outil`privileged` Paramètre .

> **Note**
>
> Exécuter un conteneur en tant que`privileged`n'est pas pris en charge sur Amazon ECS surAWS Fargate.

## Suppression des fonctionnalités Linux inutiles du conteneur

Voici une liste des fonctionnalités Linux par défaut attribuées aux conteneurs Docker. Pour plus d'informations concernant chaque fonctionnalité, consultezPrésentation des capacités Linux.

```
CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_FOWNER, CAP_FSETID, CAP_KILL,
CAP_SETGID, CAP_SETUID, CAP_SETPCAP, CAP_NET_BIND_SERVICE,
CAP_NET_RAW, CAP_SYS_CHROOT, CAP_MKNOD, CAP_AUDIT_WRITE,
CAP_SETFCAP
```

Si un conteneur ne nécessite pas toutes les capacités kernales Docker répertoriées ci-dessus, envisagez de les retirer du conteneur. Pour plus d'informations sur chaque capacité Kernal Docker,

consultezCapacités du noyau. Vous pouvez déterminer quelles fonctionnalités sont utilisées en procédant comme suit :

- Installez le package OSlibcap-nget exécutez le`pscap`pour répertorier les fonctionnalités que chaque processus utilise.
- Vous pouvez également utiliserscapshpour déchiffrer les capacités qu'un processus utilise.
- Reportez-vous àCapacités de Linux 101Pour plus d'informations, consultez.

### Utiliser une clé gérée par le client (CMK) pour chiffrer les images transmises à Amazon ECR

Vous devez utiliser une clé gérée par le client (CMK) pour encruster les images qui sont transmises à Amazon ECR. Les images qui sont transmises à Amazon ECR sont automatiquement cryptées au repos avec unAWS Key Management Service(AWS KMS). Si vous préférez utiliser votre propre clé, Amazon ECR prend désormais en chargeAWS KMSchiffrement avec les clés gérées par le client (CMK). Avant d'activer le chiffrement côté serveur avec un CMK, consultez les considérations répertoriées dans la documentation surChiffrement au repos.

## Sécurité d'exécution

La sécurité d'exécution fournit une protection active à vos conteneurs pendant qu'ils s'exécutent. L'idée est de détecter et d'empêcher les activités malveillantes de se produire dans vos conteneurs.

Avec l'informatique sécurisée (`seccomp`), vous pouvez empêcher une application conteneurisée d'effectuer certains systèmes dans le noyau du système d'exploitation hôte sous-jacent. Bien que le système d'exploitation Linux ait quelques centaines d'appels système, la plupart d'entre eux ne sont pas nécessaires pour exécuter des conteneurs. En limitant les systèmes pouvant être réalisés par un conteneur, vous pouvez réduire efficacement la surface d'attaque de votre application.

Pour commencer avec seccomp, vous pouvez utiliser`strace`pour générer une trace de pile pour voir les appels système que votre application effectue. Vous pouvez utiliser un outil tel qu'`syscall2seccomp`pour créer un profil seccomp à partir des données collectées à partir de la trace de la pile. Pour de plus amples informations, veuillez consulterstraceandsyscall2seccomp.

Contrairement au module de sécurité SELinux, seccomp ne peut pas isoler les conteneurs les uns des autres. Cependant, il protège le noyau hôte contre les appels système non autorisés. Il fonctionne en interceptant les systèmes et en permettant seulement à ceux qui ont été autorisés de

passer. Docker dispose d'unpar défautprofil seccomp adapté à la majorité des charges de travail à usage général.

> **Note**
>
> Il est également possible de créer vos propres profils pour des éléments qui nécessitent des privilèges supplémentaires.

AppArmor est un module de sécurité Linux similaire à seccomp, mais il limite les capacités d'un conteneur, y compris l'accès à certaines parties du système de fichiers. Il peut être exécuté dans`enforcement`ou`complain`Mode. Dans la mesure où la création de profils AppArmor peut être difficile, nous vous recommandons d'utiliser un outil tel quefléau. Pour plus d'informations sur AppArmor, consultez leAppArmor.

> **Important**
>
> AppArmor n'est disponible que pour les distributions Ubuntu et Debian de Linux.

## Recommendations

Nous vous recommandons de prendre les actions de folowing lors de la configuration de votre sécurité d'exécution.

### Utilisation d'une solution tierce pour la défense d'exécution

Utilisez une solution tierce pour la défense de l'exécution. Si vous êtes familier avec le fonctionnement de la sécurité Linux, créez et gérez des profils seccomp et AppArmor. Les deux sont des projets open source. Sinon, envisagez d'utiliser un autre service tiers. La plupart utilisent l'apprentissage automatique pour bloquer ou alerter les activités suspectes. Pour obtenir la liste des solutions tierces disponibles, consultezAWS Marketplacepour Conteneurs.

### Ajouter ou supprimer des fonctionnalités Linux à l'aide de stratégies seccomp

Utilisez seccomp pour avoir un meilleur contrôle sur les fonctionnalités Linux et pour éviter les erreurs de vérification syscall. Seccomp fonctionne comme un filtre syscall qui révoque l'autorisation d'exécuter certains systèmes ou d'utiliser des agruments spécifiques.

# AWSPartenaires

Vous pouvez utiliser l'un desAWSProduits partenaires pour ajouter des fonctionnalités et des fonctionnalités supplémentaires à vos charges de travail Amazon ECS. Pour de plus amples informations, veuillez consulterPartenaires Amazon ECS.

Aqua Security

Vous pouvez utiliserAqua Securitypour sécuriser vos applications cloud natives du développement à la production. Aqua Cloud Native Security Platform s'intègre à vos ressources et outils d'orchestration natifs pour fournir une sécurité transparente et automatisée. Il peut prévenir les activités suspectes et les attaques en temps réel, et aider à appliquer les politiques et à simplifier la conformité réglementaire.

Palo Alto Networks

Palo Alto Networksassure la sécurité et la protection de vos hôtes, conteneurs et infrastructure sans serveur dans le cloud et tout au long du cycle de vie du développement et du logiciel.

Twistlock est fourni par Palo Alto Networks et peut être intégré avec Amazon ECS FireLens. Avec elle, vous avez accès à des journaux et des incidents de sécurité haute fidélité qui sont regroupés de façon transparente dans plusieursAWSServices . Il s'agit notamment d'Amazon CloudWatch, d'Amazon Athena et d'Amazon Kinesis. Twistlock sécurise les charges de travail déployées surAWSServices de conteneurs.

Sysdig

Vous pouvez utiliserSysdigpour exécuter des charges de travail cloud natives sécurisées et conformes dans des scénarios de production. La plate-forme Sysdig Secure DevOps intègre des fonctionnalités de sécurité et de conformité pour protéger vos charges de travail natives dans le cloud. Elle offre également une évolutivité, des performances et une personnalisation de niveau entreprise.

# Historique du document pour le Guide des bonnes pratiques Amazon ECS

Le tableau suivant décrit les versions de documentation du Guide des bonnes pratiques Amazon ECS.

| update-history-change | update-history-description | update-history-date |
| --- | --- | --- |
| Bonnes pratiques de sécurité | Ajout de meilleures pratiques pour la gestion de la sécurité pour les charges de travail Amazon ECS. | 26 mai 2021 |
| Meilleures pratiques de mise à l'échelle automatique et de gestion des capacités | Ajout de meilleures pratiques pour la mise à l'échelle automatique et la gestion de la capacité pour les charges de travail Amazon ECS. | 14 mai 2021 |
| Bonnes pratiques de stockage persistant | Ajout de meilleures pratiques pour le stockage persistan t pour les charges de travail Amazon ECS. | 7 mai 2021 |
| Bonnes pratiques de réseau | Ajout de meilleures pratiques pour la gestion des réseaux pour les charges de travail Amazon ECS. | 6 avril 2021 |
| Première version | Première version du Guide des bonnes pratiques Amazon ECS | 6 avril 2021 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.