aws

Guide du développeur

Modèles d’AWS Blockchain

Modèles d’AWS Blockchain: Guide du développeur

Table of Contents

AWS Blockchain Templates a été abandonné le 30 avril 2019. Aucune autre mise à jour de ce service ou de cette documentation justificative ne sera apportée. Pour une expérience optimale de Managed BlockchainAWS, nous vous recommandons d'utiliser Amazon Managed Blockchain (AMB). Pour en savoir plus sur les premiers pas avec Amazon Managed Blockchain, consultez notre atelier sur Hyperledger Fabric ou notre blog sur le déploiement d'un nœud Ethereum. Si vous avez des questions sur AMB ou si vous avez besoin d'une assistance supplémentaire, contactez AWS Support l'équipe chargée de votre AWS compte.

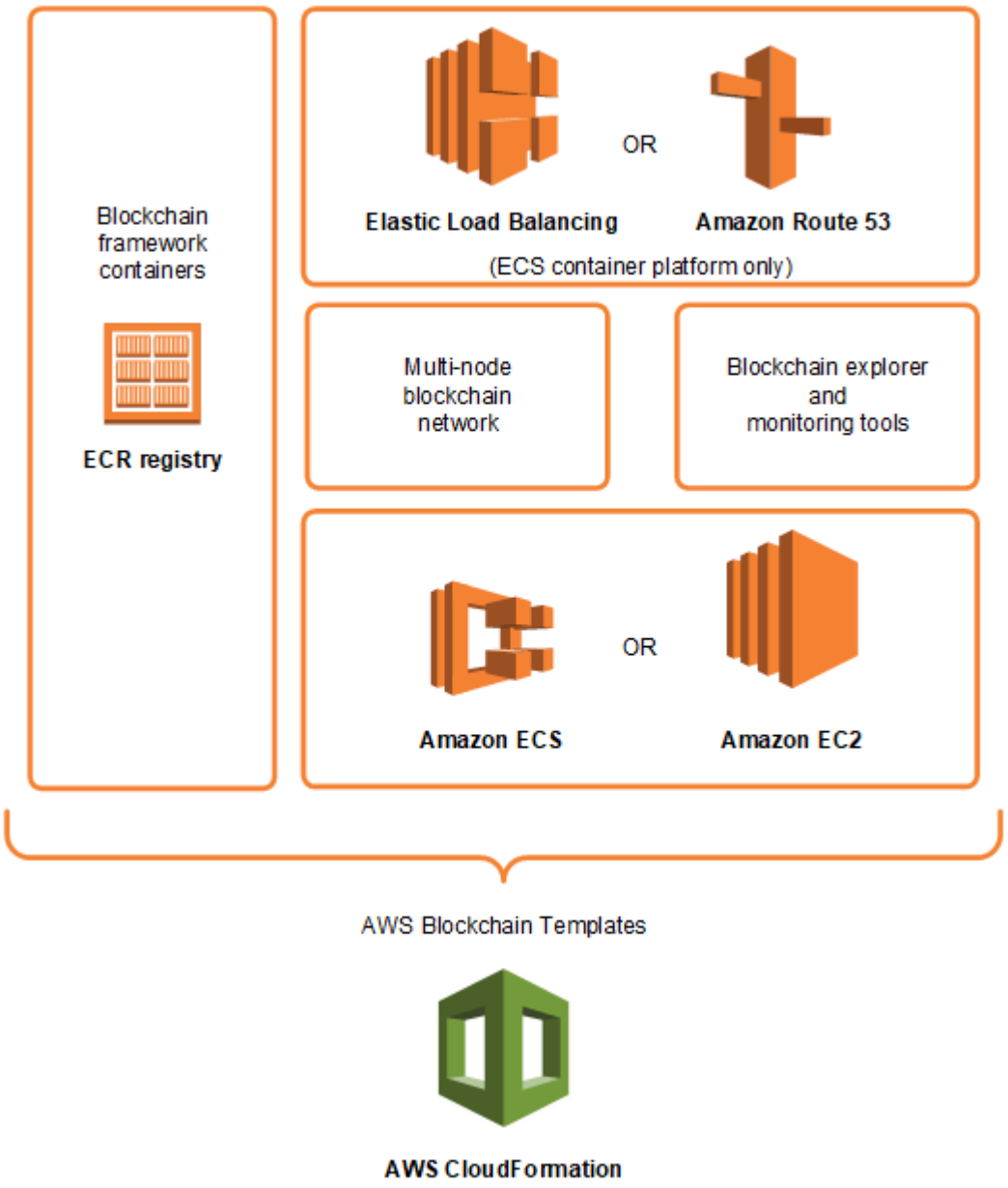
Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.

Qu'est-ce qu'AWS Blockchain Templates ?

AWS Blockchain Templates vous aide à créer et à déployer rapidement des réseaux de chaînes de blocs à l'AWSaide de différents frameworks de chaînes de blocs. La blockchain (chaîne de blocs) est une technologie de base de données décentralisée qui gère un ensemble en augmentation constante de transactions et de contrats intelligents renforcés contre les falsifications et les modifications à l'aide du chiffrement.

Un réseau blockchain est un peer-to-peer réseau qui améliore l'efficacité et l'immuabilité des transactions pour les processus commerciaux tels que les paiements internationaux, la gestion de la chaîne d'approvisionnement, l'enregistrement foncier, le financement participatif, la gouvernance, les transactions financières, etc. Cela permet à des personnes et des organisations qui ne se connaissent peut-être pas d'approuver et de vérifier de manière indépendante l'enregistrement des transactions.

Vous utilisez AWS Blockchain Templates pour configurer et lancer des AWS CloudFormation piles afin de créer des réseaux de chaînes de blocs. Les AWS ressources et les services que vous utilisez dépendent du modèle AWS Blockchain que vous choisissez et des options que vous spécifiez. Pour plus d'informations sur les modèles disponibles et leurs fonctionnalités, consultezModèles et fonctionnalités d'AWS Blockchain. Les composants fondamentaux d'un réseau de chaînes de blocs AWS créé à l'aide des modèles AWS Blockchain sont présentés dans le schéma suivant.



Comment démarrer

Le meilleur point de départ dépend de votre niveau d'expertise en matière de blockchain et, AWS en particulier, en ce qui concerne les services liés aux modèles AWS Blockchain Templates.

Je suis compétent avec AWS et blockchain

Commencez par la rubrique Modèles et fonctionnalités d'AWS Blockchain relative au framework que vous souhaitez utiliser. Utilisez les liens pour lancer le modèle AWS Blockchain et configurer le réseau blockchain, ou téléchargez les modèles pour les consulter par vous-même.

Je suis compétent avec AWS et je débute avec blockchain

Commencez par le Commencer à utiliser les modèles AWS Blockchain didacticiel. Cela vous explique comment créer un réseau de blockchain Ethereum d'introduction avec les paramètres par défaut. Lorsque vous aurez terminé, consultez Modèles et fonctionnalités d'AWS Blockchain pour obtenir une présentation des infrastructures blockchain et des liens vous permettant d'en savoir plus sur les options de configuration et les fonctions.

Je débute avec AWS et je suis compétent avec blockchain

Commencez parConfiguration des modèles de Blockchain AWS. Vous pourrez acquérir des notions fondamentales sur AWS, comme un compte et un profil utilisateur. Ensuite, parcourez le Commencer à utiliser les modèles AWS Blockchain didacticiel. Ce didacticiel vous explique comment créer un réseau de blockchain Ethereum d'introduction. Même si vous n'utiliserez finalement pas Ethereum, vous acquerrez une expérience pratique de la mise en place de services connexes. Cette expérience est utile pour tous les frameworks de blockchain. Enfin, consultez la rubrique dans la section Modèles et fonctionnalités d'AWS Blockchain de votre infrastructure.

Je débute avec AWS et blockchain

Commencez parConfiguration des modèles de Blockchain AWS. Vous pourrez acquérir des notions fondamentales sur AWS, comme un compte et un profil utilisateur. Parcourez ensuite le Commencer à utiliser les modèles AWS Blockchain didacticiel. Ce didacticiel vous explique comment créer un réseau de blockchain Ethereum d'introduction. Prenez le temps de découvrir les liens pour en savoir plus sur les services AWS et Ethereum.

Services connexes

Selon les options que vous sélectionnez, AWS Blockchain Templates peut utiliser les AWS services suivants pour déployer la blockchain :

- Amazon EC2 —Fournit une capacité de calcul pour votre réseau de chaînes de blocs. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
- Amazon ECS —Orchestre le déploiement de conteneurs entre les instances EC2 d'un cluster pour votre réseau blockchain, si vous choisissez de l'utiliser. Pour plus d'informations, consultez le Guide du développeur Amazon Elastic Container Service.

- Amazon VPC —Fournit un accès réseau pour les ressources Ethereum que vous créez. Vous pouvez personnaliser la configuration pour des raisons d'accessibilité et de sécurité. Pour plus d'informations, consultez le manuel Amazon VPC Developer Guide.
- Équilibrage de charge des applications : sert de point de contact unique pour accéder aux interfaces utilisateur disponibles et pour la découverte des services internes lors de l'utilisation d'Amazon ECS comme plate-forme de conteneurs. Pour plus d'informations, voir Qu'est-ce qu'un Application Load Balancer ? dans le guide de l'utilisateur des équilibreurs de charge d'application. .

Configuration des modèles de Blockchain AWS

Avant de commencer à utiliser les modèles AWS Blockchain, effectuez les tâches suivantes :

- Inscrivez-vous à AWS
- Créer un utilisateur IAM
- Créer une paire de clés

Ce sont des prérequis fondamentaux pour toutes les configurations de blockchain. En outre, le réseau blockchain que vous choisissez peut avoir des prérequis, qui varient en fonction de l'environnement et des choix de configuration souhaités. Pour plus d'informations, consultez la section correspondant à votre modèle de blockchain dans Modèles et fonctionnalités d'AWS Blockchain.

Pour step-by-step obtenir des instructions sur la configuration des prérequis pour un réseau Ethereum privé utilisant un cluster Amazon ECS, consultezCommencer à utiliser les modèles AWS Blockchain.

Inscrivez-vous à AWS

Lorsque vous créez un compte AWS, votre compte AWS est automatiquement inscrit à tous les services. Seuls les services que vous utilisez vous sont facturés.

Si vous possédez déjà un compte AWS, passez à la prochaine tâche. Si vous n'avez pas de compte AWS, observez la procédure suivante pour en créer un.

Créer un compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous souscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur root a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un

utilisateur administratif, et à uniquement utiliser l'utilisateur root pour effectuer tâches nécessitant un accès utilisateur root.

Notez votre numéro de compte AWS. Vous en aurez besoin lorsque vous créerez un utilisateur IAM dans le cadre de la tâche suivante.

Créer un utilisateur IAM

Pour accéder à un service AWS, vous devez fournir vos informations d'identification afin que le service puisse déterminer si vous êtes autorisé à accéder à ses ressources. La console exige votre mot de passe. Vous pouvez créer des clés d'accès pour votre compte AWS afin d'accéder à l'interface ligne de commande ou à l'API. Cependant, il est déconseillé d'accéder à AWS à l'aide des informations d'identification de votre compte AWS ; utilisez plutôt AWS Identity and Access Management (IAM). Créez un utilisateur IAM, puis ajoutez-le à un groupe IAM disposant des autorisations administratives et octroyez-lui ces autorisations. Vous pourrez alors accéder à AWS à l'aide d'une URL spéciale et des informations d'identification de l'utilisateur IAM.

Si vous êtes inscrit à AWS, mais que vous n'avez pas créé d'utilisateur IAM pour vous-même, vous pourrez en créer un à l'aide de la console IAM. Si vous avez déjà un utilisateur IAM, vous pouvez ignorer cette étape.

Afin de créer un utilisateur administrateur, choisissez l'une des options suivantes :

Choisissez un moyen de gérer votre administrateur	Pour	Par	Vous pouvez également
Dans IAM Identity Center (Recommandé)	Utiliser des identifiants à court terme pour accéder à AWS. Telles sont les meilleures pratiques en matière de	Suivre les instructions de la section Mise en route dans le AWS IAM Identity Center Guide de l'utilisateur.	Configuration de l'accès par programmation en Configurant le AWS CLI à utiliser AWS IAM Identity Centerdans le AWS Command Line Interface Guide de l'utilisateur.

Choisissez un moyen de gérer votre administrateur	Pour	Par	Vous pouvez également
	sécurité. Pour plus d'informations sur les bonnes pratiques, veuillez consulter Security best practices in IAM (français non garanti) dans le Guide de l'utilisateur IAM.		
Dans IAM (Non recommandé)	Utiliser des identifiants à long terme pour accéder à AWS.	Suivre les instructions relatives à la Création de votre premier groupe utilisateur administrateur et utilisateur IAM dans le Guide de l'utilisateur IAM.	Configuration de l'accès par programmation via la Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Pour vous connecter en tant que nouvel utilisateur IAM, déconnectez-vous de AWS Management Console, puis utilisez l'URL suivante, où your_aws_account_id désigne votre numéro de compte AWS sans les traits d'union (par exemple, si votre numéro de compte AWS est `1234-5678-9012`, votre ID de compte AWS est `123456789012`) :

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Saisissez le nom utilisateur et le mot de passe IAM que vous venez de créer. Lorsque vous êtes connecté, la barre de navigation affiche « votre_nom_utilisateur @ votre_id_de_compte_aws ».

Si vous ne voulez pas que l'URL de votre page de connexion contienne votre ID de compte AWS, vous pouvez créer un alias de compte. Dans le tableau de bord IAM, choisissez Créer un alias de

compte et entrez un alias, tel que le nom de votre entreprise. Pour vous connecter après avoir créé un alias de compte, utilisez l'URL suivante :

```
https://your_account_alias.signin.aws.amazon.com/console/
```

Pour vérifier le lien de connexion des utilisateurs IAM pour votre compte, ouvrez la console IAM et vérifiez le lien sous IAM users sign-in link (Lien de connexion des utilisateurs IAM) dans le tableau de bord.

Pour plus d'informations, consultez le Guide de l'utilisateur d'AWS Identity and Access Management.

Créer une paire de clés

AWS utilise le chiffrement de clé publique pour sécuriser les informations de connexion pour les instances de votre réseau blockchain. Vous spécifiez le nom de la paire de clés lorsque vous utilisez chaque modèle de blockchain AWS. Vous pouvez ensuite utiliser la paire de clés pour accéder aux instances directement, par exemple, afin de vous connecter à l'aide de SSH.

Si vous possédez déjà une paire de clés dans la bonne région, vous pouvez ignorer cette étape. Si vous n'avez pas encore créé de paire de clés, vous pouvez le faire à l'aide de la console Amazon EC2. Créez la paire de clés dans la même région que celle que vous utilisez pour lancer le réseau Ethereum. Pour de plus amples 'informations, veuillez consulter Régions et zones de disponibilité dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Création d'une paire de clés

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans la barre de navigation, sélectionnez une région pour la paire de clés. Vous pouvez sélectionner n'importe quelle région disponible, quel que soit votre lieu de résidence, mais les paires de clés sont spécifiques à une région. Par exemple, si vous envisagez de lancer une instance dans la région USA Est (Ohio), vous devez créer une paire de clés pour l'instance dans la même région.
3. Dans le panneau de navigation, choisissez Paires de clés, puis Créer une paire de clés.
4. Pour Key pair name (Nom de la paire de clés), saisissez un nom pour la nouvelle paire de clés. Choisissez un nom facile à retenir, tel que votre nom d'utilisateur IAM, suivi de `-key-pair`, plus le nom de la région. Par exemple, moi-key-pair-useast2. Choisissez Créer.

5. Le fichier de clé privée est automatiquement téléchargé dans votre navigateur. Le nom de base du fichier est le nom que vous avez spécifié comme nom de votre paire de clés et l'extension du nom de fichier est `.pem`.pem. Enregistrez le fichier de clé privée en lieu sûr.

   > **Important**
   >
   > C'est votre seule occasion d'enregistrer le fichier de clé privée. Vous fournissez le nom de votre paire de clés quand vous lancez le réseau Ethereum.

Pour de plus amples informations, veuillez consulter Paires de clés Amazon EC2 dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux. Pour plus d'informations sur la connexion aux instances EC2 à l'aide de la paire de clés, consultez Connect to Your Linux Instance dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

Commencer à utiliser les modèles AWS Blockchain

Ce didacticiel explique comment utiliser le modèle AWS Blockchain pour Ethereum afin de créer un réseau de blockchain privé sur AWS ThroughAWS CloudFormation. Le réseau que vous créez possède deux clients Ethereum et un mineur exécutés sur des instances Amazon EC2 dans un cluster Amazon ECS. Amazon ECS exécute ces services dans des conteneurs Docker extraits d'Amazon ECR. Avant de commencer ce didacticiel, vous trouverez utile d'en savoir plus sur les réseaux blockchain et les services AWS impliqués, mais ce n'est pas obligatoire.

Ce didacticiel part du principe que vous avez configuré les prérequis généraux décrits dansConfiguration des modèles de Blockchain AWS. En outre, vous devez configurer certaines AWS ressources, telles qu'un réseau Amazon VPC et des autorisations spécifiques pour les rôles IAM, avant d'utiliser le modèle.

Le didacticiel explique comment configurer ces prérequis. Nous avons fait des choix de configuration, mais ils ne sont pas normatifs. Tant que vous respectez les prérequis, vous pouvez faire d'autres choix de configuration en fonction des besoins de votre application et de votre environnement. Pour de plus amples informations sur les fonctions et les prérequis généraux de chaque modèle, et pour télécharger des modèles ou les lancer directement dans AWS CloudFormation, consultez Modèles et fonctionnalités d'AWS Blockchain.

Tout au long de ce didacticiel, des exemples utilisent la région USA Ouest (Oregon) (us-west-2), mais vous pouvez utiliser n'importe quelle région qui prend en charge les modèles AWS Blockchain Templates :

- Région USA Ouest (Oregon) (us-west-2)
- Région USA Est (Virginie du Nord) (us-east-1)
- Région USA Est (Ohio) (us-east-2)

> **Note**
>
> L'exécution d'un modèle dans une région non répertoriée ci-dessus permet de lancer des ressources dans la région USA Est (Virginie du Nord) (us-east-1).

Le modèle AWS Blockchain pour Ethereum que vous configurez à l'aide de ce didacticiel crée les ressources suivantes :

- Instances EC2 à la demande du type et du numéro que vous spécifiez. Le didacticiel utilise le type d'instance t2.medium par défaut.
- Un équilibreur de charge d'application interne.

Après le tutoriel, des étapes sont fournies pour nettoyer les ressources que vous créez.

Rubriques

Configuration des prérequis

Le modèle AWS Blockchain pour la configuration Ethereum que vous spécifiez dans ce didacticiel nécessite que vous fassiez ce qui suit :

Créer un VPC et des sous-réseaux

Le modèle AWS Blockchain pour Ethereum lance des ressources dans un réseau virtuel que vous définissez à l'aide d'Amazon Virtual Private Cloud (Amazon VPC). La configuration que vous spécifiez dans ce didacticiel crée un équilibreur de charge d'application qui nécessite deux sous-réseaux publics dans des zones de disponibilité distinctes. En outre, un sous-réseau privé est requis pour les instances de conteneur et le sous-réseau doit se trouver dans la même zone de disponibilité que l'équilibreur de charge d'application. Vous utilisez d'abord l'Assistant VPC pour créer un sous-réseau public et un sous-réseau privé dans la même zone de disponibilité. Vous créez ensuite un second sous-réseau public dans ce VPC dans une autre zone de disponibilité.

Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon VPC ? dans le Guide de l'utilisateur Amazon VPC.

Utilisez la console Amazon VPC (https://console.aws.amazon.com/vpc/) pour créer l'adresse IP élastique, le VPC et le sous-réseau, comme décrit ci-dessous.

Pour créer une adresse IP Elastic

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Choisissez Elastic IPs (Adresses IP Elastic, Allocate new address (Allouer une nouvelle adresse, Allocate (Allouer).
3. Notez l'adresse IP Elastic (EIP) que vous créez, puis choisissez Close (Fermer).
4. Dans la liste des adresses IP Elastic (EIP), recherchez l'Allocation ID (ID d'allocation) de l'adresse IP Elastic (EIP) créée précédemment. Vous l'utiliserez lorsque vous créerez le VPC.

Pour créer le VPC

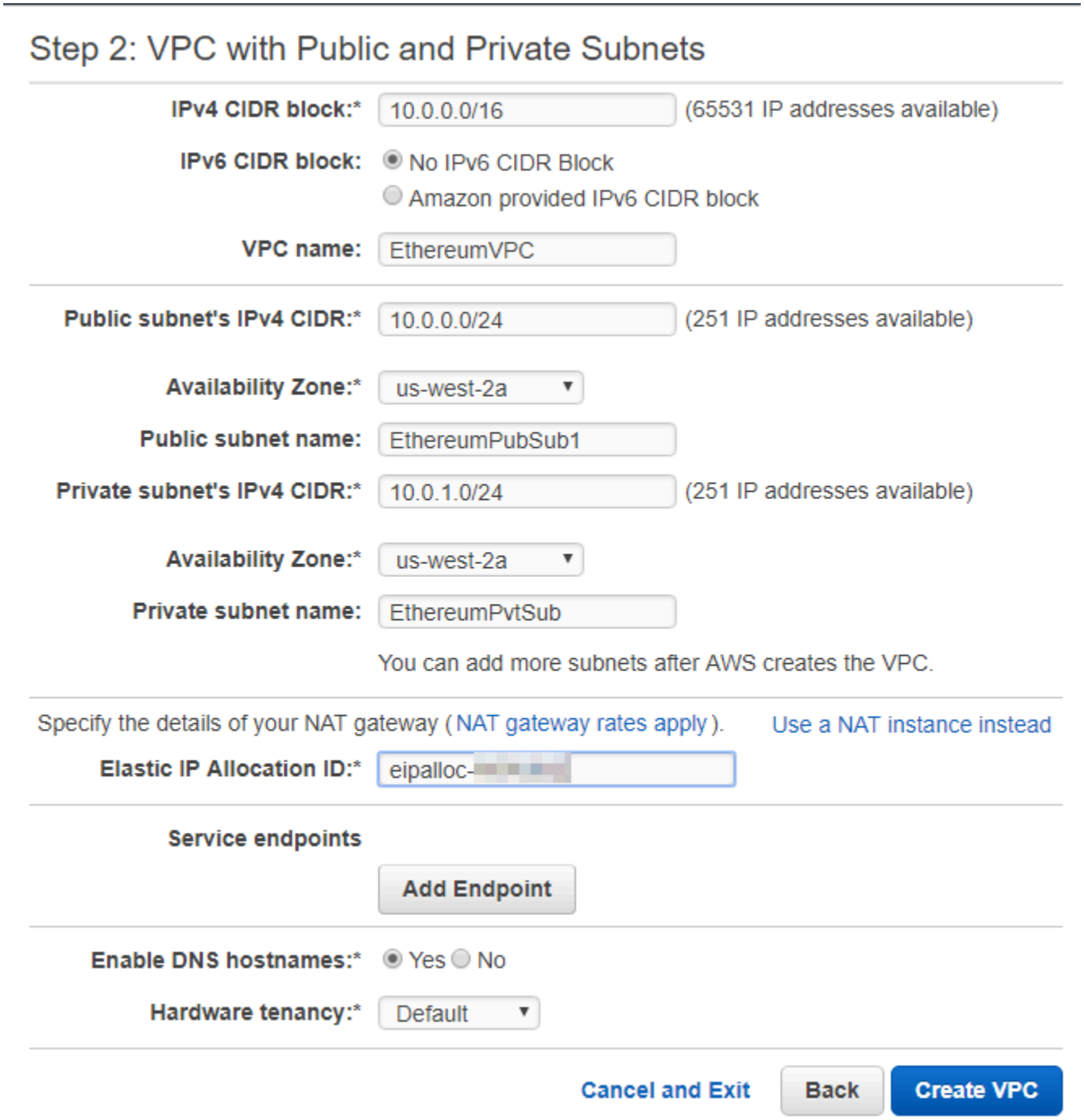
1. Dans la barre de navigation, sélectionnez une région pour le VPC. Les VPC étant spécifiques à une région, sélectionnez la même région dans laquelle vous avez créé votre paire de clés et dans laquelle vous lancez la pile Ethereum. Pour plus d'informations, consultez Créer une paire de clés.
2. Dans le tableau de bord VPC, choisissez Démarrer l'assistant VPC.
3. Sur la page Step 1: Select a VPC Configuration (Étape 1 : Sélectionner une configuration de VPC), choisissez VPC with Public and Private Subnets (VPC avec sous-réseaux publics et privés), puis Select (Sélectionner).
4. Sur la page Step 2: VPC with Public and Private Subnets (Étape 2 : VPC avec des sous-réseaux publics et privés), laisser le IPv4 CIDR block (Bloc d'adresse CIDR IPv4) et le IPv6 CIDR block (Bloc d'adresse CIDR IPv6) à leurs valeurs par défaut. Pour VPC name (Nom du VPC), saisissez un nom convivial.
5. Pour Public subnet's IPv4 CIDR (Bloc d'adresse CIDR IPv4 du sous-réseau public), conservez la valeur par défaut. Pour Availability Zone (Zone de disponibilité), choisissez une zone. Pour Public subnet name (Nom du sous-réseau public), saisissez un nom convivial.

   Vous spécifiez ce sous-réseau en tant que le premier sous-réseau des deux sous-réseaux de l'équilibreur de charge d'application lorsque vous utilisez le modèle.

   Notez la zone de disponibilité de ce sous-réseau, car vous sélectionnez la même zone de disponibilité pour le sous-réseau privé et une différente pour l'autre sous-réseau public.

6. Pour Private subnet’s IPv4 CIDR (Bloc d’adresse CIDR IPv4 du sous-réseau privé), conservez la valeur par défaut. Pour Availability Zone (Zone de disponibilité), sélectionnez la même zone de disponibilité que lors de l'étape précédente. Pour Private subnet name (Nom du sous-réseau privé), saisissez un nom convivial.
7. Dans le champ Elastic IP Allocation ID (ID d'allocation d'adresses IP Elastic), choisissez l'ID de l'adresse IP Elastic que vous avez créée précédemment.
8. Conservez les valeurs par défaut pour les autres paramètres.
9. Sélectionnez Create VPC (Créer un VPC).

   L'exemple ci-dessous montre un VPC EthereumNetworkVPC avec un sous-réseau public EthereumPubSub1 et un sous-réseau privé 1. EthereumPvtSub Le sous-réseau public utilise la zone de disponibilité us-west-2a.

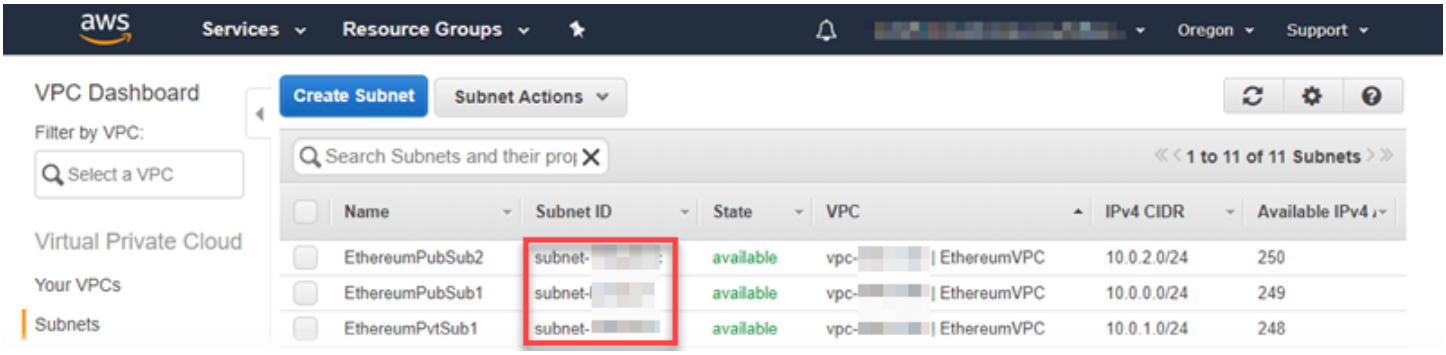


Pour créer le second sous-réseau public dans une autre zone de disponibilité

1. Choisissez Subnets (Sous-réseaux), puis sélectionnez le sous-réseau public que vous avez créé précédemment dans la liste. Sélectionnez l'onglet Route Table (Table de routage) et notez l'ID de la Route table (Table de routage. Vous spécifiez cette même table de routage pour le deuxième sous-réseau public ci-dessous.

2. Choisissez Create Subnet.
3. Pour Name tag (Balise de nom), entrez un nom pour le sous-réseau. Vous utilisez ce nom plus tard lorsque vous créez l'hôte bastion dans ce réseau.
4. Dans le champ VPC, sélectionnez le VPC que vous avez créé précédemment.
5. Pour Availability Zone (Zone de disponibilité), sélectionnez une autre zone à partir de la zone choisie pour le premier sous-réseau public.
6. Pour IPv4 CIDR block (Bloc d'adresse CIDR IPv4),saisissez 10.0.2.0/24.
7. Choisissez Yes, Create. Le sous-réseau est ajouté à la liste des sous-réseaux.
8. Une fois le sous-réseau sélectionné dans la liste, choisissez Subnet Actions (Actions de sous-réseau), Modify auto assign IP settings (Modifier l'affectation automatique des paramètres IP). Sélectionnez Auto-assign IPs (Affecter automatiquement des adresses IP), Save (Enregistrer), Close (Fermer). Cela permet à l'hôte bastion d'obtenir une adresse IP publique lorsque vous le créez dans ce sous-réseau.
9. Sous l'onglet Route Table (Table de routage), choisissez Edit (Modifier). Sous Change to (Remplacer par), sélectionnez l'ID de la table de routage que vous avez noté précédemment et choisissez Save (Enregistrer).

Vous devriez maintenant voir trois sous-réseaux pour le VPC que vous avez créé précédemment. Notez les noms et les ID des sous-réseaux pour pouvoir les spécifier à l'aide du modèle.



Créer des groupes de sécurité

Les groupes de sécurité agissent comme des pare-feux, contrôlant le trafic entrant et sortant vers les ressources. Lorsque vous utilisez le modèle pour créer un réseau Ethererum sur un cluster Amazon ECS, vous spécifiez deux groupes de sécurité :

- Un groupe de sécurité pour les instances EC2 qui contrôle le trafic de et vers les instances EC2 dans le cluster

- Groupe de sécurité pour l'équilibreur de charge d'application qui contrôle le trafic entre ce dernier, les instances EC2 et l'hôte bastion. Vous associez également ce groupe de sécurité à l'hôte bastion.

Chaque groupe de sécurité possède des règles qui autorisent la communication entre l'Application Load Balancer et les instances EC2, ainsi que d'autres règles minimales. Cela nécessite que les groupes de sécurité se réfèrent les uns aux autres. Pour cette raison, vous devez d'abord créer les groupes de sécurité, puis les mettre à jour avec les règles appropriées.

Pour créer deux groupes de sécurité

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans le volet de navigation, choisissez Security Groups, Create Security Group.
3. Pour Security group name (Nom du groupe de sécurité), saisissez un nom pour le groupe de sécurité facile à identifier et qui le différenciera des autres, par exemple EthereumEC2-SG ou EthereumALB-SG. Vous utiliserez ces noms plus tard. Sous Description, saisissez un résumé.
4. Dans le champ VPC, sélectionnez le VPC que vous avez créé précédemment.
5. Choisissez Créer.
6. Répétez les étapes ci-dessus pour créer un autre groupe de sécurité.

Ajouter des règles entrantes au groupe de sécurité pour les instances EC2

1. Sélectionner le groupe de sécurité pour les instances EC2 créées précédemment
2. Dans l'onglet Entrant, choisissez Modifier.
3. Pour Type, sélectionnez All traffic (Tout le trafic). Pour Source, laissez Custom (Personnalisé) sélectionné, puis choisissez le groupe de sécurité que vous modifiez actuellement dans la liste, par exemple, EthereumEC2-SG. Cela permet aux instances EC2 du groupe de sécurité de communiquer les unes avec les autres.
4. Choisissez Add Rule (Ajouter une règle).
5. Pour Type, sélectionnez Tout le trafic. Pour Source, laissez Custom (Personnalisé) sélectionné, puis choisissez le groupe de sécurité de l'équilibreur de charge d'application à partir de la liste, par exemple, EthereumALB-SG. Cela permet aux instances EC2 du groupe de sécurité de communiquer avec l'équilibreur de charge d'application.
6. Choisissez Enregistrer.

Ajouter des données entrantes et modifier des règles sortantes pour le groupe de sécurité de l'équilibreur de charge d'application

1. Sélectionner le groupe de sécurité des équilibreurs de charge d'application créés précédemment
2. Dans l'onglet Inbound (Entrant), choisissez Edit (Modifier), puis ajoutez les règles entrantes suivantes :

   a. Pour Type, sélectionnez All traffic (Tout le trafic). Pour Source, laissez Custom (Personnalisé) sélectionné, puis choisissez le groupe de sécurité que vous modifiez actuellement dans la liste, par exemple, EthereumALB-SG. Cela permet à l'équilibreur de charge d'application de communiquer avec lui-même et avec l'hôte bastion.

   b. Choisissez Add Rule (Ajouter une règle).

   c. Pour Type, sélectionnez Tout le trafic. Pour Source, laissez Custom (Personnalisé) sélectionné, puis choisissez le groupe de sécurité pour les instances EC2 de la liste, par exemple, EthereumEC2-SG. Cela permet aux instances EC2 du groupe de sécurité de communiquer avec l'équilibreur de charge d'application et avec l'hôte bastion.

   d. Choisissez Add Rule (Ajouter une règle).

   e. Pour Type, choisissez SSH. Dans Source, sélectionnez My IP (Mon adresse IP), qui détecte l'adresse IP CIDR de votre ordinateur et la fournit.

   > ⚠ Important
   >
   > Cette règle permet à l'hôte bastion d'accepter le trafic SSH depuis votre ordinateur. Ainsi, ce dernier peut utiliser l'hôte bastion pour afficher les interfaces web et se connecter aux instances EC2 sur le réseau Ethereum. Pour permettre à d'autres personnes de se connecter au réseau Ethereum, ajoutez-les en tant que sources dans cette règle. Autoriser uniquement le trafic entrant vers des sources approuvées.

   f. Choisissez Enregistrer.

3. Sous l'onglet Outbound (Sortant), choisissez Edit (Modifier) et supprimer la règle qui a été créée automatiquement pour autoriser le trafic sortant pour toutes les adresses IP.
4. Choisissez Add Rule (Ajouter une règle).
5. Pour Type, sélectionnez Tout le trafic. Pour Destination, laissez Custom (Personnalisé) sélectionné, puis choisissez le groupe de sécurité pour les instances EC2 de la liste. Cela

autorise les connexions sortantes de l'équilibreur de charge d'application et e l'hôte bastion pour les instances EC2 du réseau Ethereum.

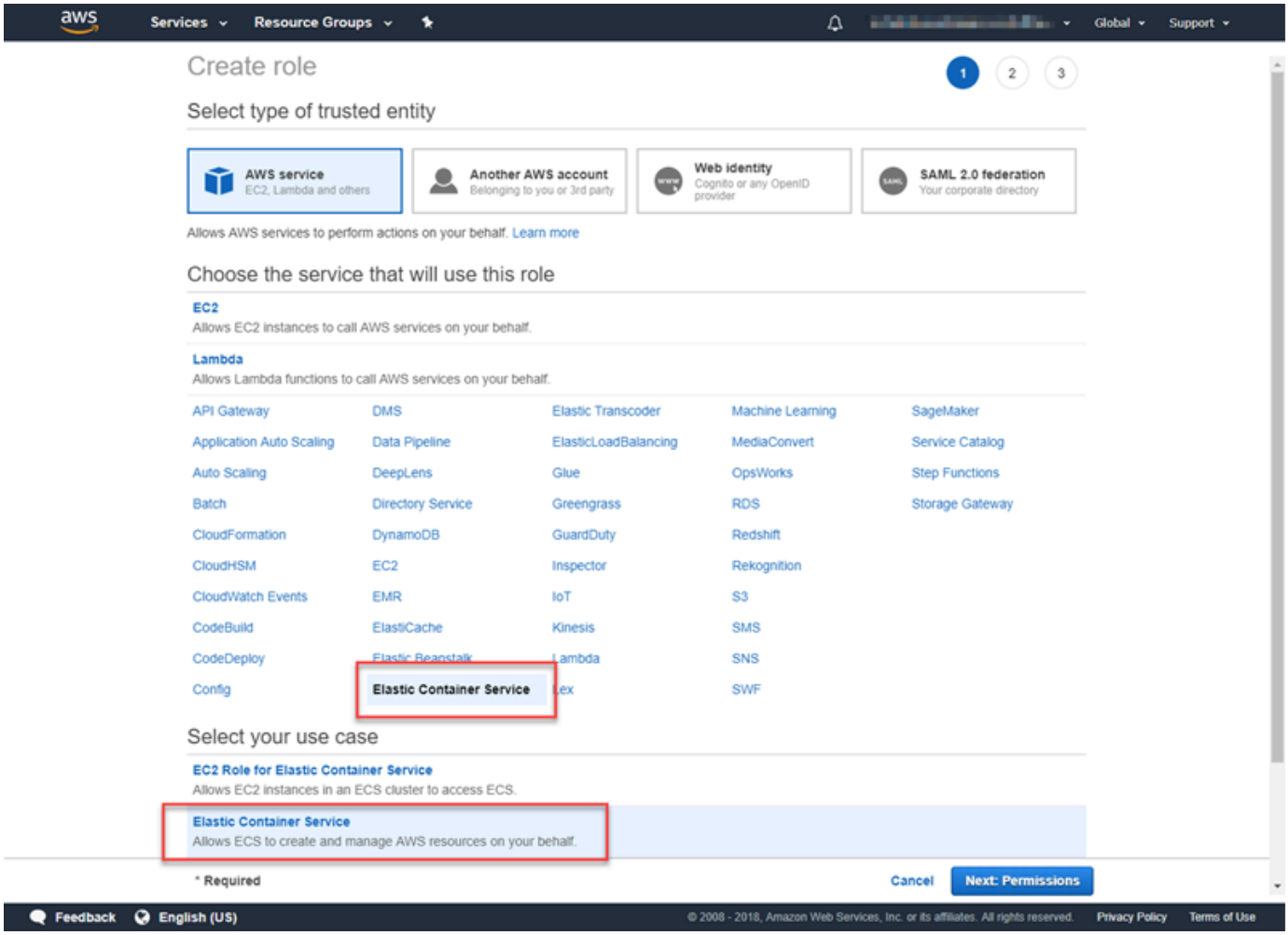
6. Choisissez Add Rule (Ajouter une règle).
7. Pour Type, sélectionnez Tout le trafic. Pour Destination, laissez Custom (Personnalisé) sélectionné, puis choisissez le groupe de sécurité que vous modifiez actuellement dans la liste, par exemple, EthereumALB-SG. Cela permet à l'équilibreur de charge d'application de communiquer avec lui-même et avec l'hôte bastion.
8. Choisissez Enregistrer.

Création d'un rôle IAM pour Amazon ECS et d'un profil d'instance EC2

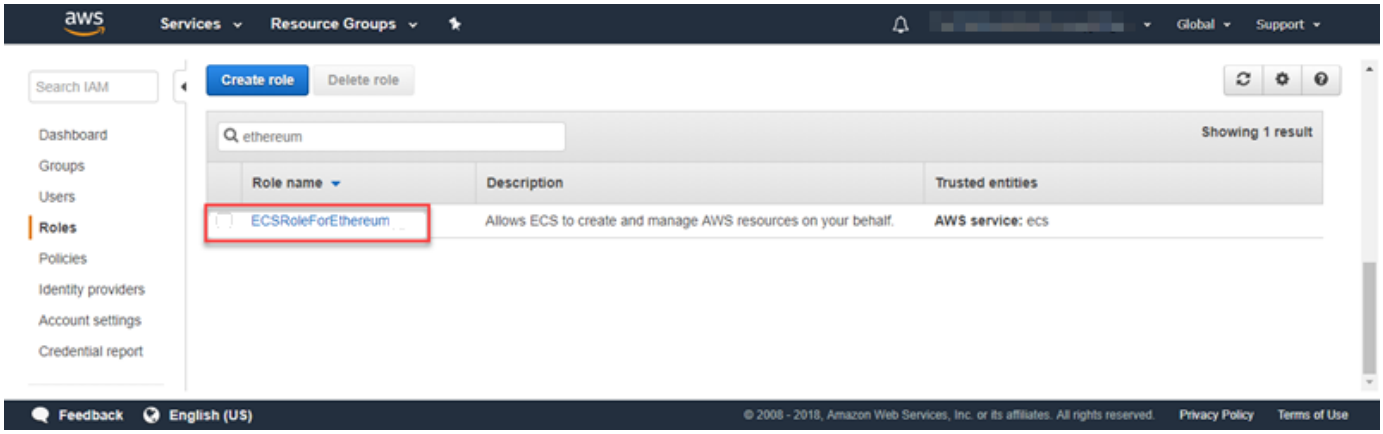
Lorsque vous utilisez ce modèle, vous spécifiez un rôle IAM pour Amazon ECS et un profil d'instance EC2. Les politiques d'autorisation associées à ces rôles permettent aux ressources et aux instances AWS de votre cluster d'interagir avec d'autres ressources AWS. Pour plus d'informations, veuillez consulter Rôles IAM dans le Guide de l'utilisateur IAM. Vous configurez le rôle IAM pour Amazon ECS et le profil d'instance EC2 à l'aide de la console IAM (https://console.aws.amazon.com/iam/).

Pour créer le rôle IAM pour Amazon ECS

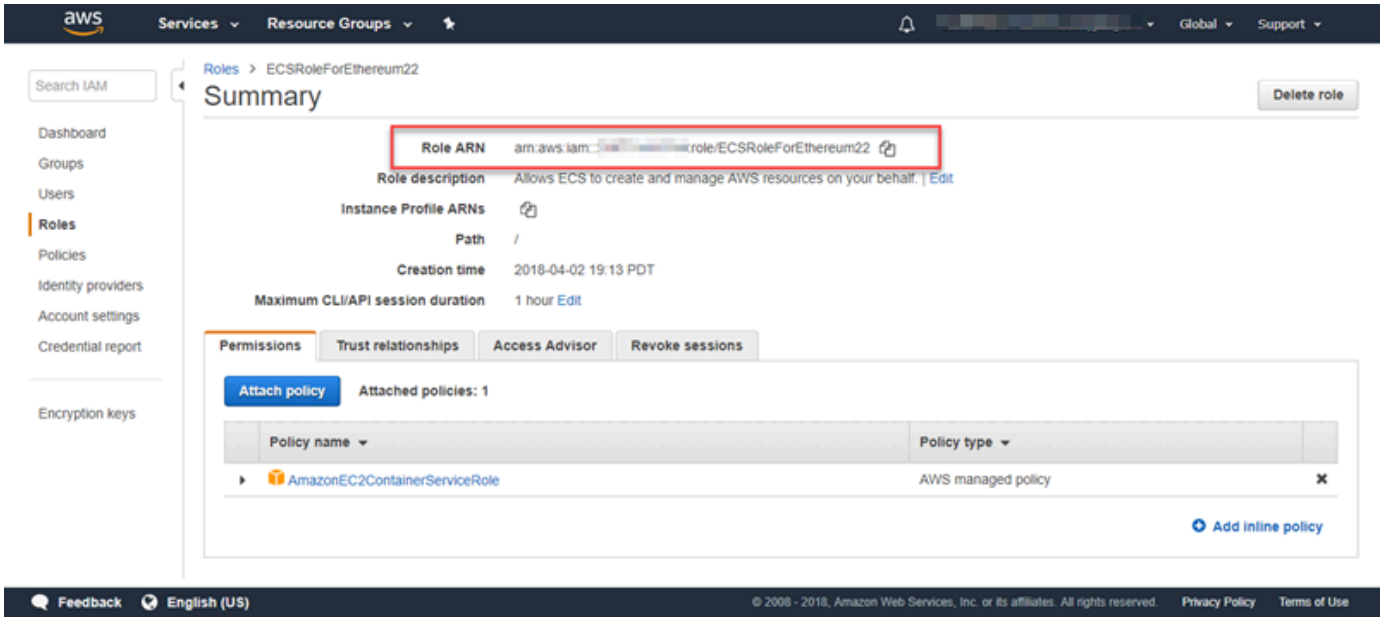
1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Rôles, puis Créer un rôle.
3. Sous Select type of trusted entity (Sélectionner le type d'entité de confiance), choisissez AWS service (Service AWS).
4. Sous Choisir le service qui utilisera ce rôle, choisissez Elastic Container Service.
5. Sous Select your use case (Sélectionnez votre cas d'utilisation), choisissez Elastic Container Service, Next:Permissions (Suivant : Autorisations).



6. Pour la politique d'autorisations, laissez la politique par défaut (AmazonEC2 ContainerServiceRole) sélectionnée, puis choisissez Next:Review.
7. Dans Nom du rôle, entrez une valeur qui vous aide à identifier le rôle, telle que ECS RoleForEthereum. Pour Role Description (Description du rôle), saisissez un résumé. Notez le nom du rôle pour plus tard.
8. Sélectionnez Créer un rôle.
9. Sélectionnez le rôle que vous venez de créer dans la liste. Si votre compte comporte de nombreux rôles, vous pouvez rechercher le nom du rôle.



10. Copiez la valeur Role ARN (ARN du rôle) et enregistrez-la afin de pouvoir la copier à nouveau. Vous aurez besoin de cet ARN lorsque vous créerez le réseau Ethereum.



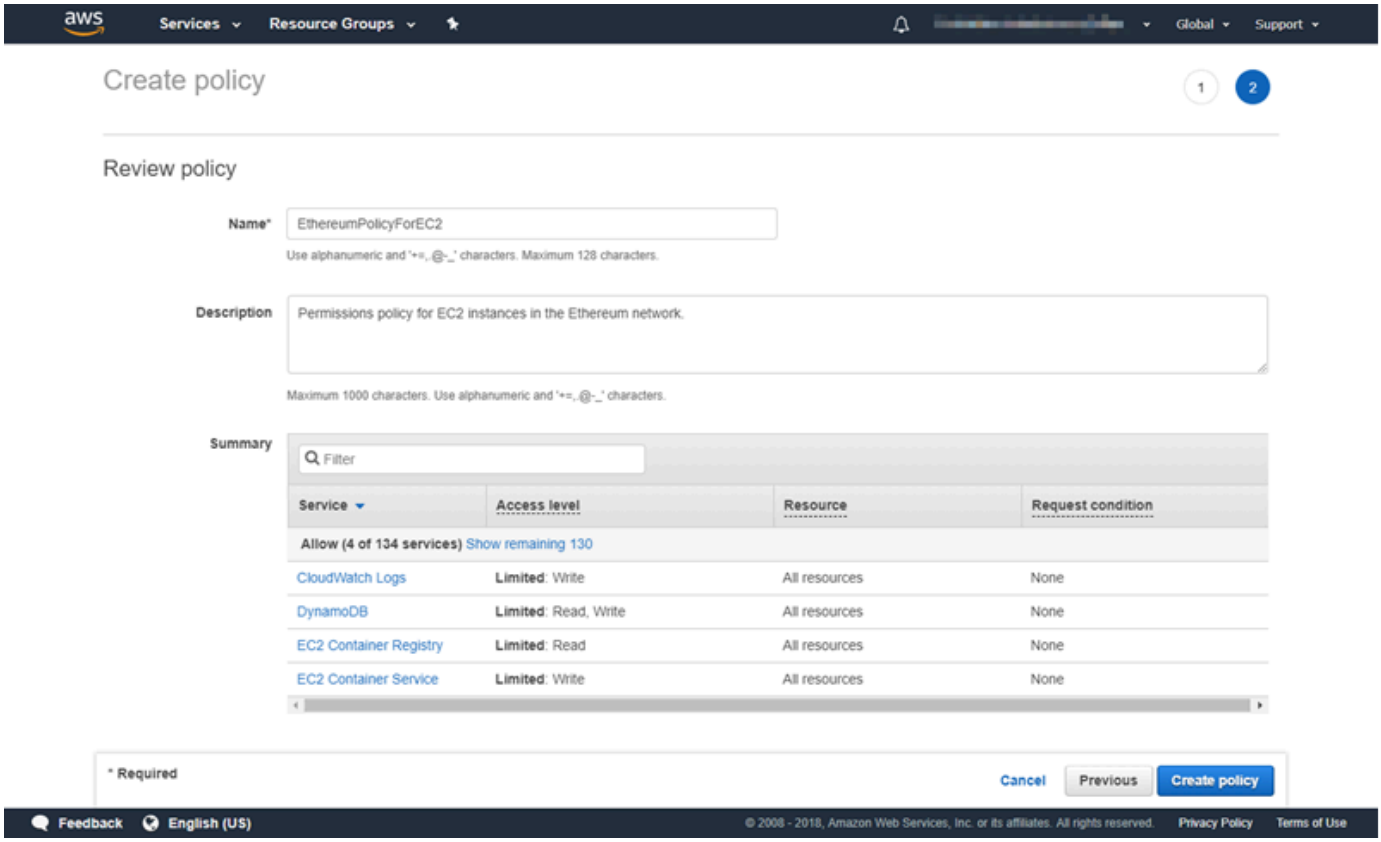
Le profil d'instance EC2 que vous spécifiez dans le modèle est assumé par les instances EC2 dans le réseau Ethereum comme interagissant avec d'autres services AWS. Vous créez une stratégie d'autorisation pour le rôle, puis créez le rôle (qui crée automatiquement un profil d'instance du même nom), et enfin vous attachez la stratégie d'autorisation au rôle.

Pour créer un profil d'instance EC2

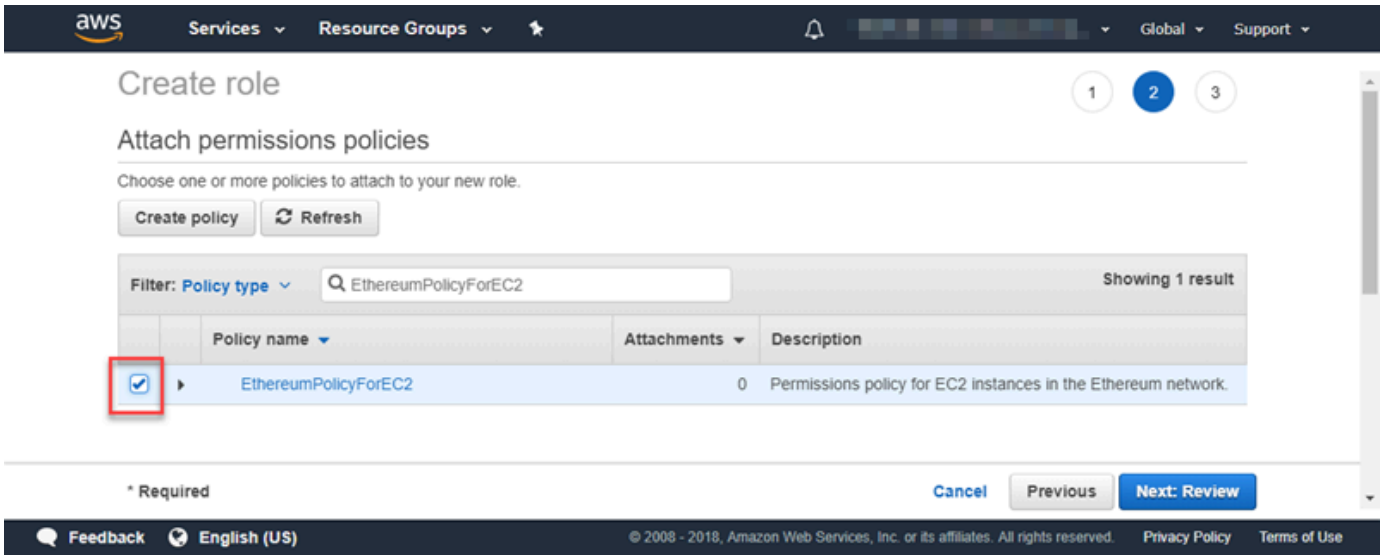
1. Dans le panneau de navigation, choisissez Policies (Politiques), puis Create policy (Créer une politique).
2. Choisissez JSON et remplacez la déclaration de stratégie par défaut par le code JSON suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "dynamodb:BatchGetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Choisissez Examiner une politique.
4. Dans Nom, entrez une valeur qui vous aide à identifier cette politique d'autorisation, par exemple EthereumPolicyForEC2. Sous Description, saisissez un résumé. Choisissez Créer une politique.

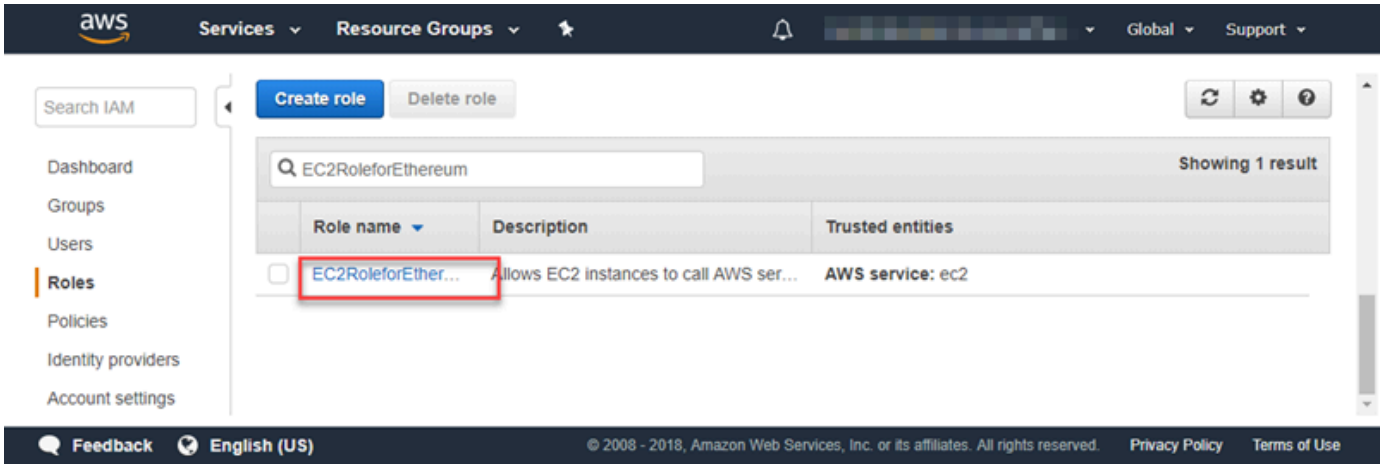


5. Choisissez Roles (Rôles), Create role (Créer un rôle).
6. Choisissez EC2, Next: Permissions (Suivant : Autorisations).
7. Dans le champ Rechercher, entrez le nom de la politique d'autorisation que vous avez créée précédemment, par exemple EthereumPolicyForEC2.
8. Sélectionnez la case à cocher de la stratégie que vous avez créée précédemment, puis choisissez Next: Review (Suivant : Vérification).

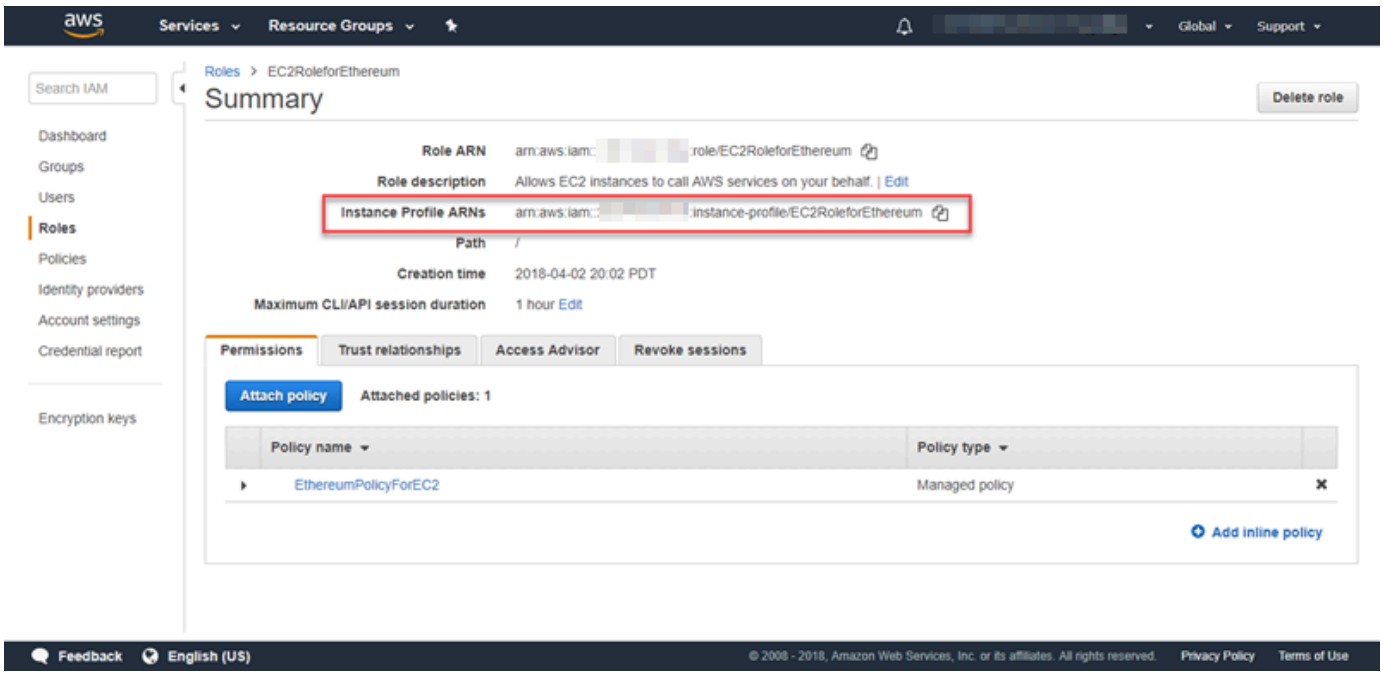


9. Dans Nom du rôle, entrez une valeur qui vous aide à identifier le rôle, par exemple EC2. RoleForEthereum Pour Role description (Description du rôle, saisissez un résumé. Choisissez Create role (Créer un rôle).

10. Sélectionnez le rôle que vous venez de créer dans la liste. Si votre compte possède de nombreux rôles, vous pouvez saisir le nom du rôle dans le champ Search (Recherche).



11. Copiez la valeur de l'Instance Profile ARN (ARN de profil d'instance) et enregistrez-le afin de pouvoir le copier à nouveau. Vous aurez besoin de cet ARN lorsque vous créerez le réseau Ethereum.



Création d'un hôte bastion

Dans ce didacticiel, vous créez un bastion host (hôte bastion). Il s'agit d'une instance EC2 que vous utilisez pour vous connecter aux interfaces Web et aux instances de votre réseau Ethereum. Son seul but est de transférer le trafic SSH de clients approuvés à l'extérieur du VPC, afin qu'ils puissent accéder aux ressources du réseau Ethereum.

Vous configurez l'hôte bastion, car l'équilibreur de charge d'application créé par le modèle est interne, ce qui signifie qu'il achemine uniquement les adresses IP internes. L'hôte bastion :

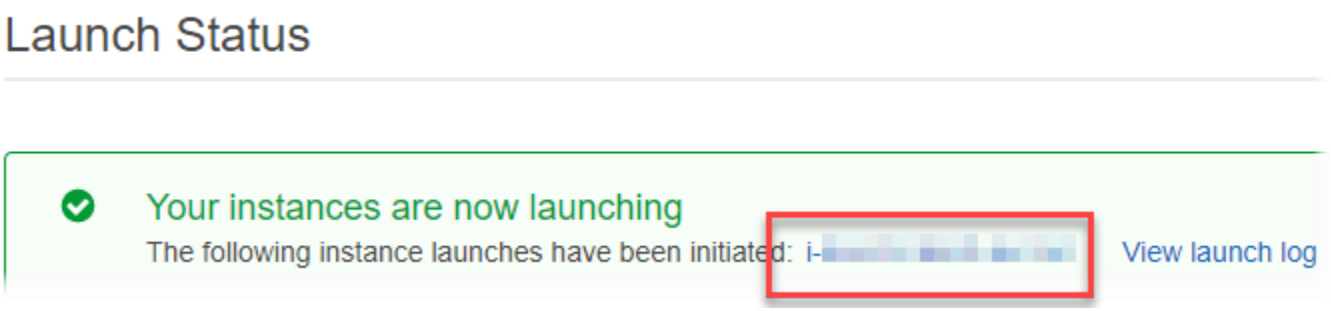
- Dispose d'une adresse IP interne que l'équilibreur de charge d'application reconnaît, car vous la lancez dans le deuxième sous-réseau public que vous avez créé précédemment.
- Dispose d'une adresse IP publique que le sous-réseau attribue, accessible par des sources approuvées hors du VPC.
- Est associé au groupe de sécurité de l'équilibreur de charge d'application que vous avez créé précédemment, qui dispose d'une règle entrante qui autorise le trafic SSH (port 22) à partir de clients approuvés.

Pour pouvoir accéder au réseau Ethereum, les clients de approuvés doivent être configurés pour se connecter via l'hôte bastion. Pour plus d'informations, consultez Se connecter à l'hôte Bastion EthStats et l' EthExplorer utiliser. Un hôte bastion est une approche. Vous pouvez utiliser n'importe quelle approche qui fournit l'accès des clients approuvés aux ressources privées dans un VPC.

Création d'un hôte bastion

1. Suivez les cinq premières étapes pour lancer une instance dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.
2. Choisissez Modifier les détails de l'instance. Pour Network (Réseau), choisissez le VPC que vous avez créé précédemment. Pour Subnet (Sous-réseau), sélectionnez le deuxième sous-réseau public que vous avez créé précédemment. Conservez les valeurs par défaut de tous les autres paramètres.
3. Confirmez la modification lorsque vous y êtes invité, puis choisissez Review and Launch (Vérifier et lancer).
4. Sélectionnez Edit security groups (Modifier les groupes de sécurité). Pour Attribuer un groupe de sécurité, choisissez Select an existing security group (Sélectionner un groupe de sécurité existant).

5. Dans la liste des groupes de sécurité, sélectionnez le groupe de sécurité pour l'équilibreur de charge d'application que vous avez créé précédemment, puis choisissez Review and Launch (Vérifier et lancer).
6. Choisissez Lancer.
7. Notez l'ID de l'instance. Vous en aurez besoin plus tard quand vous Se connecter à l'hôte Bastion EthStats et l' EthExplorer utiliser.



Créer le réseau Ethereum

Le réseau Ethereum que vous spécifiez à l'aide du modèle décrit dans cette rubrique lance une AWS CloudFormation pile qui crée un cluster Amazon ECS d'instances EC2 pour le réseau Ethereum. Le modèle s'appuie sur les ressources créées précédemment dans Configuration des prérequis.

Lorsque vous lancez la pile AWS CloudFormation à l'aide du modèle, celui-ci crée des piles imbriquées pour certaines tâches. Une fois celles-ci terminées, vous pouvez vous connecter aux ressources servies par le biais de l'équilibreur de charge d'application du réseau à travers l'hôte bastion, pour vérifier que votre réseau Ethereum est en cours d'exécution et accessible.

Pour créer le réseau Ethereum à l'aide du modèle AWS Blockchain pour Ethereum

1. Consultez Getting Started with AWS Blockchain Templates et ouvrez le dernier modèle de blockchain AWS pour Ethereum dans la AWS CloudFormation console à l'aide des liens rapides correspondant à votre région AWS.
2. Entrez les valeurs en fonction des consignes suivantes :

   - Pour Stack name (Nom de pile), entrez un nom facile à identifier. Ce nom est utilisé dans les noms des ressources créées par la pile.

- Sous Ethereum Network Parameters (Paramètres réseau Ethereum) et Private Ethereum Network Parameters (Paramètres du réseau privé Ethereum), conservez les paramètres par défaut.

  > **Warning**
  >
  > Utilisez les comptes par défaut et la phrase mnémonique associée à des fins de test uniquement. N'envoyez pas de véritable ether à l'aide de l'ensemble de comptes par défaut, car quiconque ayant accès à la phrase mnémonique peut y accéder, ou le voler, à partir des comptes. Spécifiez plutôt des comptes personnalisés à des fins de production. La phrase mnémonique associée au compte par défaut est `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

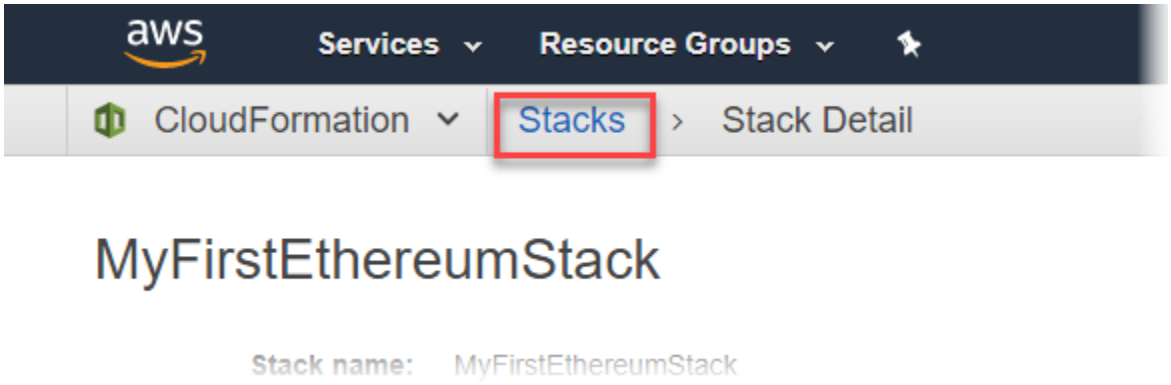
- Sous Configuration de la plate-forme, conservez les paramètres par défaut, ce qui crée un cluster Amazon ECS d'instances EC2. L'autre option, docker-local crée un réseau Ethereum à l'aide d'une seule instance EC2.
- Sous EC2 configuration (Configuration EC2), sélectionnez les options selon les instructions suivantes :
  - Pour EC2 Key Pair (Paire de clés EC2), sélectionnez une paire de clés. Pour plus d'informations sur la création d'une paire de clés, consultez Créer une paire de clés.
  - Pour EC2 Security Group (Groupe de sécurité EC2), sélectionnez le groupe de sécurité créé précédemment dans Créer des groupes de sécurité.
  - Pour EC2 Instance Profil ARN (ARN de profil d'instance EC2), saisissez l'ARN du profil d'instance créé précédemment dans Création d'un rôle IAM pour Amazon ECS et d'un profil d'instance EC2.
- Sous VPC network configuration (Configuration réseau VPC), sélectionnez les options selon les instructions suivantes :
  - Pour VPC ID (ID du VPC), sélectionnez le VPC que vous avez créé précédemment dans Créer un VPC et des sous-réseaux.
  - Pour Ethereum Network Subnet IDs (ID de sous-réseau de réseau Ethereum), sélectionnez le seul sous-réseau privé que vous avez créé précédemment au cours de la procédure To create the VPC.
- Sous ECS cluster configuration (Configuration du cluster ECS), conservez les valeurs par défaut. Cela crée un cluster ECS de trois instances EC2.

- Sous Application Load Balancer (ECS only) (Configuration de l'équilibreur de charge d'application (ECS uniquement (ECS uniquement)), sélectionnez les options selon les instructions suivantes :
  - Pour les Application Load Balancer Subnet IDs (ID de sous-réseau de l'équilibreur de charge d'application), sélectionnez deux sous-réseaux publics parmi les list of subnets que vous avez notés précédemment.
  - Pour Application Load Balancer Security Group (Groupe de sécurité de l'équilibreur de charge d'application), sélectionnez le groupe de sécurité de l'équilibreur de charge d'application créé précédemment dans Créer des groupes de sécurité.
  - Pour le rôle IAM, entrez l'ARN du rôle ECS que vous avez créé précédemment. Création d'un rôle IAM pour Amazon ECS et d'un profil d'instance EC2
- Sous EthStats, sélectionnez les options conformément aux instructions suivantes :
  - Pour Deploy EthStats, conservez le paramètre par défaut, qui est vrai.
  - Pour EthStats Connection Secret, entrez une valeur arbitraire d'au moins six caractères.
- Sous EthExplorer, laissez le paramètre par défaut pour Deploy EthExplorer, qui est vrai.
- Sous Other parameters (Autres paramètres), conservez la valeur par défaut pour le Nested Template S3 URL Prefix (Préfixe d'URL du modèle imbriqué S3) et notez cette valeur. C'est là que vous pouvez trouver des modèles imbriqués.

3. Conservez les valeurs par défaut pour tous les autres paramètres, activez la case à cocher de confirmation, puis choisissez Create (Créer).

   La page Stack Detail (Détails de la pile) pour la pile racine lancée par AWS CloudFormation s'affiche.

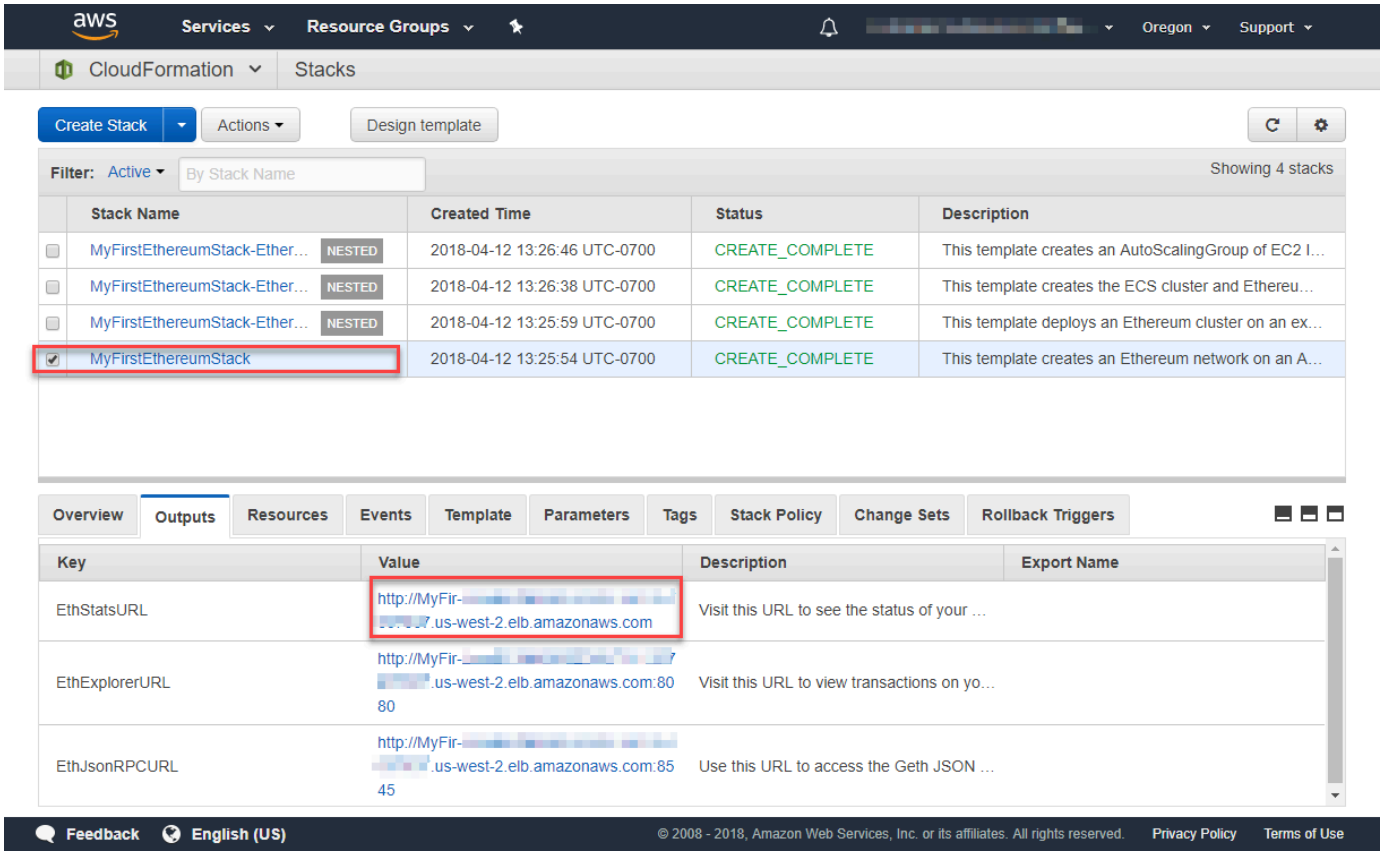
4. Pour surveiller la progression de la pile racine et des piles imbriquées, choisissez Stacks (Piles).



5. Lorsque toutes les piles affichent CREATE_COMPLETE pour Status, vous pouvez vous connecter aux interfaces utilisateur d'Ethereum pour vérifier que le réseau fonctionne et est

accessible. Lorsque vous utilisez la plate-forme de conteneurs ECS, les URL de connexion et de EthStats EthJson RPC via l'Application Load Balancer sont disponibles dans l'onglet Outputs de la pile racine. EthExplorer

> **Important**
>
> Vous ne pourrez pas vous connecter directement à ces URL ou SSH directement tant que vous n'aurez pas configuré une connexion proxy via l'hôte bastion sur votre ordinateur client. Pour plus d'informations, consultez Se connecter à l'hôte Bastion EthStats et l' EthExplorer utiliser.



Se connecter à l'hôte Bastion EthStats et l' EthExplorer utiliser

Pour vous connecter aux ressources Ethereum dans ce didacticiel, vous configurez le réacheminement de port SSH (tunnelisation SSH) via l'hôte bastion. Les instructions suivantes montrent comment procéder pour que vous puissiez vous connecter à un navigateur EthStats et

accéder à des EthExplorer URL à l'aide d'un navigateur. Dans les instructions ci-dessous, vous configurez d'abord un proxy SOCKS sur un port local. Vous utilisez ensuite une extension de navigateur pour utiliser ce port transféré pour les URL de votre réseau Ethereum. FoxyProxy

Sous Mac OS ou Linux, utilisez un client SSH pour configurer la connexion proxy SOCKS à l'hôte bastion. Sous Windows, utilisez PuTTY. Avant de vous connecter, vérifiez que l'ordinateur client que vous utilisez est spécifié comme source autorisée pour le trafic SSH entrant dans le groupe de sécurité de l'équilibreur de charge d'application que vous avez configuré précédemment.

Pour se connecter à l'hôte bastion avec le réacheminement de port SSH à l'aide de SSH

- Suivez les procédures décrites dans la section Connexion à votre instance Linux à l'aide de SSH dans le guide de l'utilisateur Amazon EC2 pour les instances Linux. Pour l'étape 4 de la procédure de connexion à votre instance Linux, ajoutez `-D 9001` à la commande SSH, spécifiez la même paire de clés que celle que vous avez spécifiée dans le modèle de blockchain AWS pour la configuration Ethereum et spécifiez le nom DNS de l'hôte bastion.

  ```
  ssh -i /path/my-template-key-pair.pem ec2-user@bastion-host-dns -D 9001
  ```

Pour se connecter à l'hôte bastion avec le réacheminement de port SSH à l'aide de PuTTY (Windows)

1. Suivez les procédures décrites dans la section Connexion à votre instance Linux depuis Windows à l'aide de PuTTY du guide de l'utilisateur Amazon EC2 pour les instances Linux jusqu'à l'étape 7 de la procédure de démarrage d'une session PuTTY, en utilisant la même paire de clés que celle que vous avez spécifiée dans le modèle de blockchain AWS pour la configuration d'Ethereum.
2. Dans PuTTY, sous Category (Catégorie), choisissez Connection (Connexion), SSH, Tunnels.
3. Pour le Port forwarding (Réacheminement de port), choisissez Local ports accept connections from other hosts (Ports locaux acceptent les connexions d'autres hôtes).
4. Sous Add new forwarded port (Ajouter un nouveau port réacheminé) :

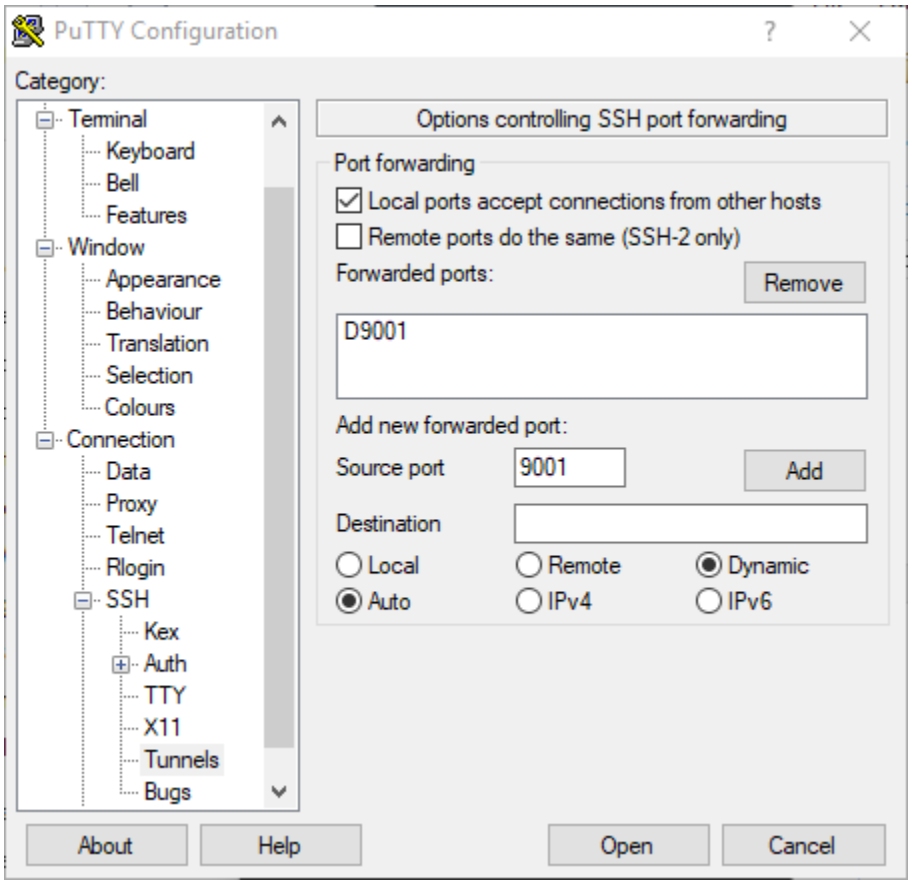
   a. Pour Source port (Port source), saisissez 9001. C'est un port arbitraire non utilisé que nous avons choisi, et vous pouvez en choisir un autre si nécessaire.

   b. Laissez Destination vide.

   c. Sélectionnez Dynamic (Dynamique).

d. Choisissez Ajouter.

Pour Forwarded ports (Ports réacheminés), D9001 doit apparaître comme indiqué ci-dessous.



5. Choisissez Open (Ouvrir), puis authentifiez-vous auprès de l'hôte bastion tel que requis par votre configuration de clé. Laissez la connexion ouverte.

Avec la connexion PuTTY ouverte, vous configurez maintenant votre système ou une extension du navigateur pour utiliser le port réacheminé pour vos URL du réseau Ethereum. Les instructions suivantes sont basées sur l'utilisation de FoxyProxy Standard pour transférer les connexions en fonction du modèle EthStats d' EthExplorer URL du port 9001, que vous avez défini précédemment comme port transféré, mais vous pouvez utiliser la méthode de votre choix.

FoxyProxy Pour configurer l'utilisation du tunnel SSH pour les URL du réseau Ethereum

Cette procédure a été écrite sur la base de Chrome. Si vous utilisez un autre navigateur, traduisez les paramètres et la séquence dans la version de FoxyProxy pour ce navigateur.

1. Téléchargez et installez l'extension de navigateur FoxyProxy standard, puis ouvrez les options conformément aux instructions de votre navigateur.
2. Choisissez Add New Proxy (Ajouter un nouveau proxy).
3. Sous l'onglet General (Général), assurez-vous que le proxy est Enabled (Activé) et saisissez un Proxy Name (Nom de proxy) et des Proxy Notes (Notes de proxy), qui vous aident à identifier cette configuration de proxy.
4. Sous l'onglet Proxy Details (Détails du proxy), choisissez Manual Proxy Configuration (Configuration manuelle du proxy). Pour Host or IP Adress (Hôte ou adresse IP) (ou Server or IP Address (Serveur ou adresse IP) dans certaines versions), saisissez localhost. Pour Port, saisissez 9001. Sélectionnez SOCKS proxy? (Proxy SOCKS ?).
5. Sous l'onglet URL Pattern (Modèle d'URL) choisissez Add New Pattern (Ajouter un nouveau modèle).
6. Pour Nom du modèle, entrez un nom facile à identifier, et pour Modèle d'URL, entrez un modèle qui correspond à toutes les URL de ressources Ethereum que vous avez créées avec le modèle, par exemple http://internal - MyUser -LoadB-*. Pour de plus amples informations sur l'affichage des URL, consultez Ethereum URLs.
7. Laissez les sélections par défaut pour les autres paramètres et choisissez Save (Enregistrer).

Vous pouvez désormais vous connecter aux URL Ethereum, qui sont disponibles sur CloudFormation console à l'aide de l'onglet Sorties de la pile racine que vous avez créée avec le modèle.

Nettoyage des ressources

AWS CloudFormation facilite pour le nettoyage des ressources créées par la pile. Lorsque vous supprimez la pile, toutes les ressources créées par celle-ci sont supprimées.

Pour supprimer les ressources créées par le modèle

- Ouvrez la console AWS CloudFormation, sélectionnez la pile racine créée précédemment, choisissez Actions, Delete (Supprimer).

  Le Status (Statut) de la pile racine créée précédemment et les piles imbriquées associées passe à DELETE_IN_PROGRESS.

Vous pouvez choisir de supprimer les prérequis créés pour le réseau Ethereum.

Supprimer le VPC

- Ouvrez la console Amazon VPC, sélectionnez le VPC que vous avez créé précédemment, puis choisissez Actions, Supprimer le VPC. Cette opération supprime également les sous-réseaux, les groupes de sécurité, et la passerelle NAT associée au VPC.

Supprimer le rôle IAM et le profil d'instance EC2

- Ouvrez la console IAM et choisissez Rôles. Sélectionnez le rôle pour ECS et le rôle pour EC2 créés précédemment, puis choisissez Delete (Supprimer).

Arrêtez l'instance EC2 pour l'hôte bastion

- Ouvrez le tableau de bord Amazon EC2, choisissez Running instances, sélectionnez l'instance EC2 que vous avez créée pour l'hôte Bastion, choisissez Actions, État de l'instance, Terminate.

Modèles et fonctionnalités d'AWS Blockchain

Cette section fournit des liens pour vous permettre de commencer à créer un réseau blockchain immédiatement, ainsi que des informations sur les options de configuration et les prérequis pour configurer le réseau sur AWS.

Les modèles suivants sont disponibles :

- Modèle AWS Blockchain pour Ethereum
- Modèle de blockchain AWS pour Hyperledger Fabric

AWS Blockchain Templates est disponible dans les régions suivantes :

- Région USA Ouest (Oregon) (us-west-2)
- Région USA Est (Virginie du Nord) (us-east-1)
- Région USA Est (Ohio) (us-east-2)

> **Note**
>
> L'exécution d'un modèle dans une région non répertoriée ci-dessus permet de lancer des ressources dans la région USA Est (Virginie du Nord) (us-east-1).

Utilisation du modèle AWS Blockchain pour Ethereum

Ethereum est un framework de blockchain qui exécute des contrats intelligents à l'aide de Solidity, un langage spécifique à Ethereum. Homestead est la version la plus récente d'Ethereum. Pour plus d'informations, consultez la documentation Ethereum Homestead et la documentation Solidity.

Liens pour le lancement

Consultez Getting Started with AWS Blockchain Templates pour obtenir des liens à lancer AWS CloudFormation dans des régions spécifiques à l'aide des modèles Ethereum.

Options Ethereum

Lorsque vous configurez le réseau Ethereum à l'aide du modèle, vous faites des choix qui déterminent les exigences suivantes :

- Choix de la plateforme de conteneur
- Choisir un réseau Ethereum privé ou public
- Modification des comptes par défaut et de la phrase mnémonique

Choix de la plateforme de conteneur

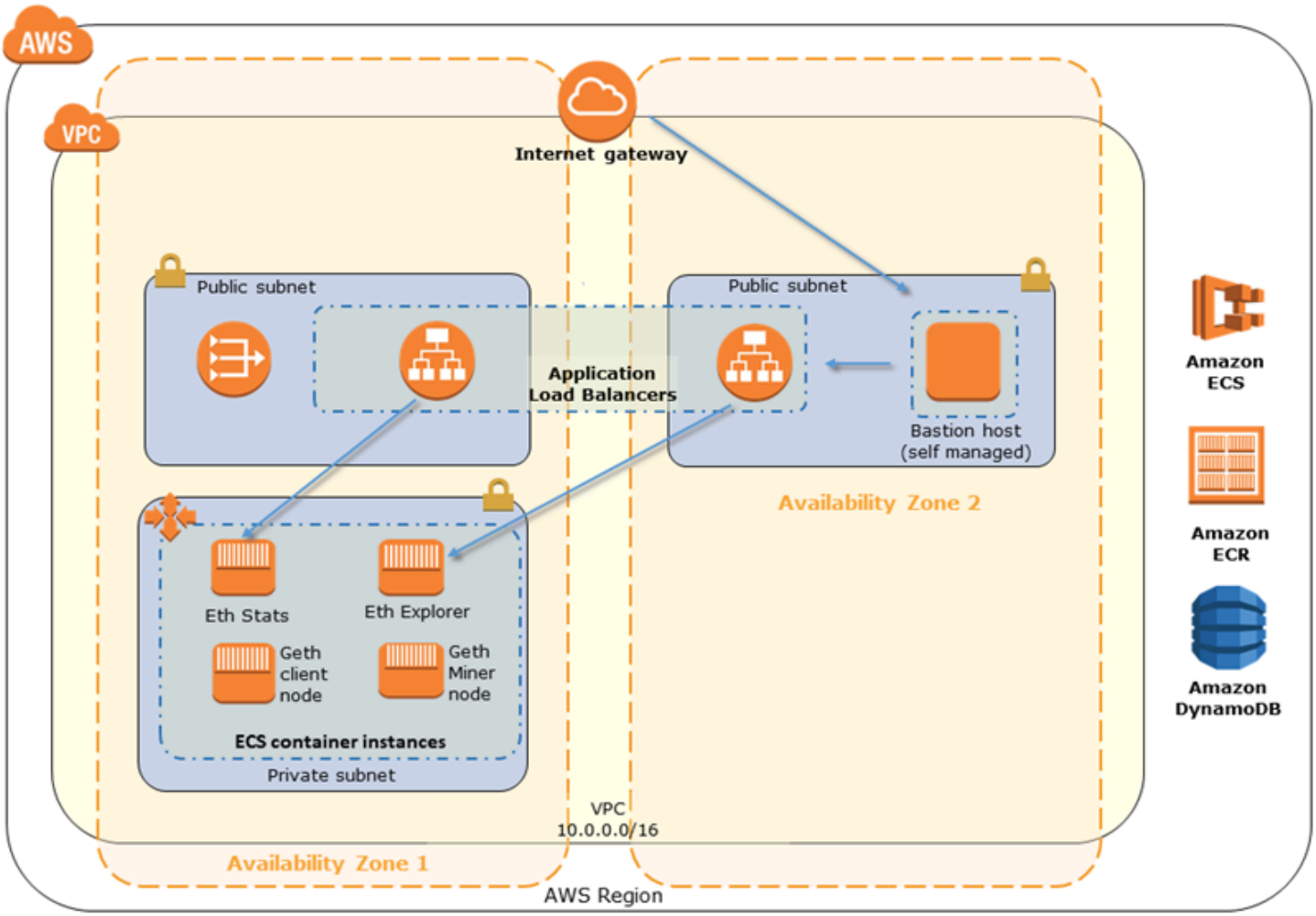
Les modèles AWS Blockchain Templates utilisent des conteneurs Docker stockés dans Amazon ECR pour déployer un logiciel de blockchain. Le modèle AWS Blockchain pour Ethereum propose deux options pour la plateforme de conteneurs :

- ecs —Spécifie qu'Ethereum s'exécute sur un cluster Amazon ECS d'instances Amazon EC2.
- docker-local —Spécifie qu'Ethereum s'exécute sur une seule instance EC2.

Utilisation de la plateforme de conteneurs Amazon ECS

Avec Amazon ECS, vous créez votre réseau Ethereum sur un cluster ECS composé de plusieurs instances EC2, avec un Application Load Balancer et les ressources associées. Pour plus d'informations sur l'utilisation de la configuration Amazon ECS, consultez le Commencer à utiliser les modèles AWS Blockchain didacticiel.

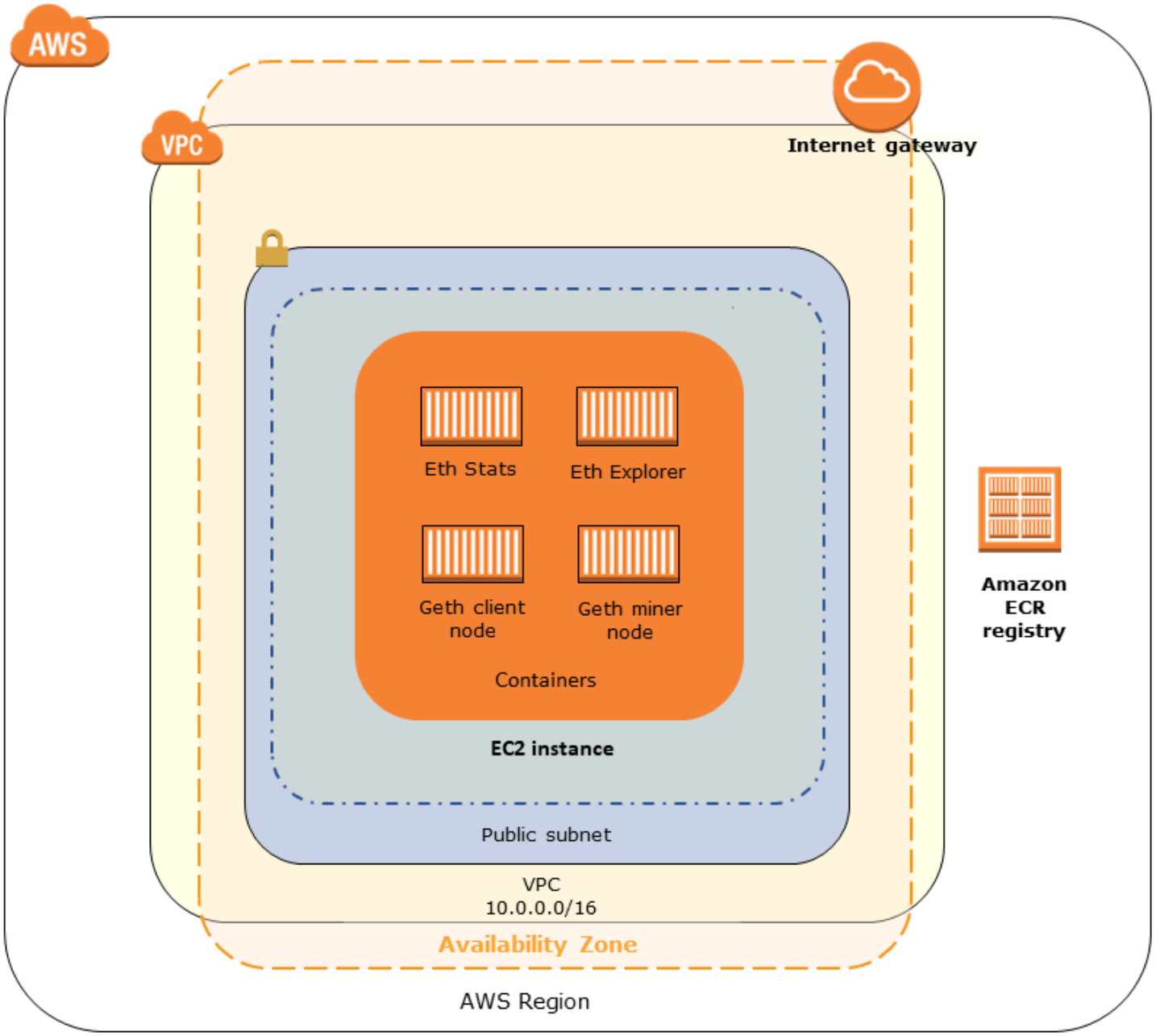
Le schéma suivant illustre un réseau Ethereum créé à l'aide du modèle avec l'option de plateforme de conteneur ECS :



Utilisation de la plateforme Docker-Local

Vous pouvez également lancer des conteneurs Ethereum au sein d'une seule instance Amazon EC2. Tous les conteneurs s'exécutent sur une seule instance EC2. Il s'agit d'une configuration simplifiée.

Le schéma suivant illustre un réseau Ethereum créé à l'aide du modèle avec l'option de plateforme de conteneur docker-local :



Choisir un réseau Ethereum privé ou public

Le choix d'une valeur d'ID réseau Ethereum autre que 1 à 4 crée des nœuds Ethereum privés qui s'exécutent dans un réseau que vous définissez, à l'aide de paramètres de réseau privé que vous spécifiez.

Lorsque vous choisissez un identifiant de réseau Ethereum compris entre 1 et 4, les nœuds Ethereum que vous créez sont joints au réseau public Ethereum. Vous pouvez ignorer les paramètres du réseau privé et leurs valeurs par défaut. Si vous choisissez de joindre des nœuds Ethereum au

réseau Ethereum public, assurez-vous que les services appropriées de votre réseau sont accessibles via Internet.

Modification des comptes par défaut et de la phrase mnémonique

Une phrase mnémonique est un ensemble aléatoire de mots que vous pouvez utiliser pour générer des portefeuilles Ethereum (c'est-à-dire, des paires de clés privées/publiques) pour les comptes associés sur n'importe quel réseau. La phrase mnémonique peut être utilisée par les comptes associés pour accéder à Ether. Nous avons créé une phrase mnémonique par défaut associée aux comptes par défaut qu'utilise le modèle Ethereum.

> **Warning**
>
> Utilisez les comptes par défaut et la phrase mnémonique associée à des fins de test uniquement. N'envoyez pas de véritable ether à l'aide de l'ensemble de comptes par défaut, car quiconque ayant accès à la phrase mnémonique peut y accéder, ou le voler, à partir des comptes. Spécifiez plutôt des comptes personnalisés à des fins de production. La phrase mnémonique associée au compte par défaut est `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

Prérequis

Lorsque vous configurez votre réseau Ethereum à l'aide du modèle AWS Blockchain pour Ethereum, les exigences minimales répertoriées ci-dessous doivent être satisfaites. Le modèle requiert les composants AWS répertoriés pour chacune des catégories suivantes :

Rubriques

- Conditions préalables pour accéder aux ressources Ethereum
- Conditions préalables à l'IAM
- Prérequis du groupe de sécurité
- Prérequis pour VPC
- Exemple d'autorisations IAM pour le profil d'instance EC2 et le rôle ECS

Conditions préalables pour accéder aux ressources Ethereum

Prérequis	Pour la plateforme ECS	Pour Docker-Local
Une paire de clés Amazon EC2 que vous pouvez utiliser pour accéder aux instances EC2. Cette clé doit être dans la même région que le cluster ECS et autres ressources.	✓	✓
Composant accessible sur Internet, tel qu'un hôte bastion ou un équilibreur de charge accessible sur Internet, avec une adresse interne à partir de laquelle le trafic est autorisé dans l'équilibreur de charge d'application. Ceci est requis avec la plateforme ECS, car le modèle crée un équilibre ur de charge interne pour des raisons de sécurité. Ceci est requis avec la plateforme docker-local lorsque l'instanc e EC2 se trouve dans un sous-réseau privé, ce que nous recommandons. Pour de plus amples informations sur la configuration d'un hôte bastion, consultez Création d'un hôte bastion.	✓	✓ (avec sous-réseau privé)

Conditions préalables à l'IAM

Prérequis	Pour la plateforme ECS	Pour Docker-Local
Un principal IAM (utilisateur ou groupe) autorisé à utiliser tous les services associés.	✓	✓
Un profil d'instance Amazon EC2 avec les autorisations appropriées permettant aux instances EC2 d'interagir avec d'autres services. Pour plus d'informations, consultez To create an EC2 instance profile.	✓	✓
Rôle IAM avec des autorisations permettant à Amazon ECS d'interagir avec d'autres services. Pour plus d'informations, consultez Création des rôles et autorisations ECS.	✓	

Prérequis du groupe de sécurité

Prérequis	Pour la plateforme ECS	Pour Docker-Local
Un groupe de sécurité pour les instances EC2, avec les conditions requises suivantes :	✓	✓
• Règles sortantes autorisant le trafic vers 0.0.0.0/0 (par défaut).	✓	✓
• Règle entrante qui autorise tout le trafic en provenanc	✓	✓

Prérequis	Pour la plateforme ECS	Pour Docker-Local
e d'elle-même (le même groupe de sécurité).		
• Règle entrante qui autorise tout le trafic en provenanc e du groupe de sécurité pour l'équilibreur de charge d'application.	✓	
• Règles entrantes qui autorisent le protocole HTTP (port 80) EthStats (servi sur le port 8080), le protocole JSON RPC sur HTTP (port 8545) et le protocole SSH (port 22) à partir de sources externes fiables, telles que le CIDR IP de votre ordinateur client.		✓

Prérequis	Pour la plateforme ECS	Pour Docker-Local
Un groupe de sécurité de l'équilibreur de charge d'application, avec les conditions requises suivantes : • Règle entrante qui autorise tout le trafic en provenance d'elle-même (le même groupe de sécurité). • Règle entrante autorisant tout le trafic depuis le groupe de sécurité pour les instances EC2. • Les règles sortantes qui autorisent tout le trafic uniquement vers le groupe de sécurité pour les instances EC2. Pour plus d'informations, consultez Créer des groupes de sécurité. • Si vous associez ce même groupe de sécurité à un hôte bastion, règle entrante qui autorise le trafic SSH (port 22) à partir de sources approuvées. • Si l'hôte bastion ou un autre composant accessible sur Internet se trouve dans un groupe de sécurité différent, règle entrante qui autorise le trafic à partir de ce composant.	✓	

Prérequis pour VPC

Prérequis	Pour la plateforme ECS	Pour Docker-Local
Une adresse IP élastique , utilisée pour accéder aux services Ethereum.	✓	✓
Un sous-réseau pour exécuter des instances EC2. Un sous-réseau privé est vivement recommandé.	✓	✓
Deux sous-réseaux publiquement accessibles. Chaque sous-réseau doit se trouver dans des zones de disponibilité différentes, dont l'un dans la même zone de disponibilité que le sous-réseau pour les instances EC2.	✓	

Exemple d'autorisations IAM pour le profil d'instance EC2 et le rôle ECS

Vous spécifiez un ARN de profil d'instance EC2 comme l'un des paramètres lorsque vous utilisez le modèle. Si vous utilisez la plate-forme de conteneur ECS, vous spécifiez également un ARN de rôle ECS. Les politiques d'autorisation associées à ces rôles permettent aux ressources et aux instances AWS de votre cluster d'interagir avec d'autres ressources AWS. Pour plus d'informations, veuillez consulter Rôles IAM dans le Guide de l'utilisateur IAM. Utilisez les instructions de stratégie et les procédures ci-dessous comme point de départ pour créer des autorisations.

Exemple de stratégie d'autorisations pour le profil d'instance EC2

La politique d'autorisation suivante décrit les actions autorisées pour le profil d'instance EC2 lorsque vous choisissez la plate-forme de conteneur ECS. Les mêmes instructions de stratégie peuvent être utilisées dans une plateforme de conteneur docker-local, avec les clés de contexte `ecs` supprimées pour limiter l'accès.

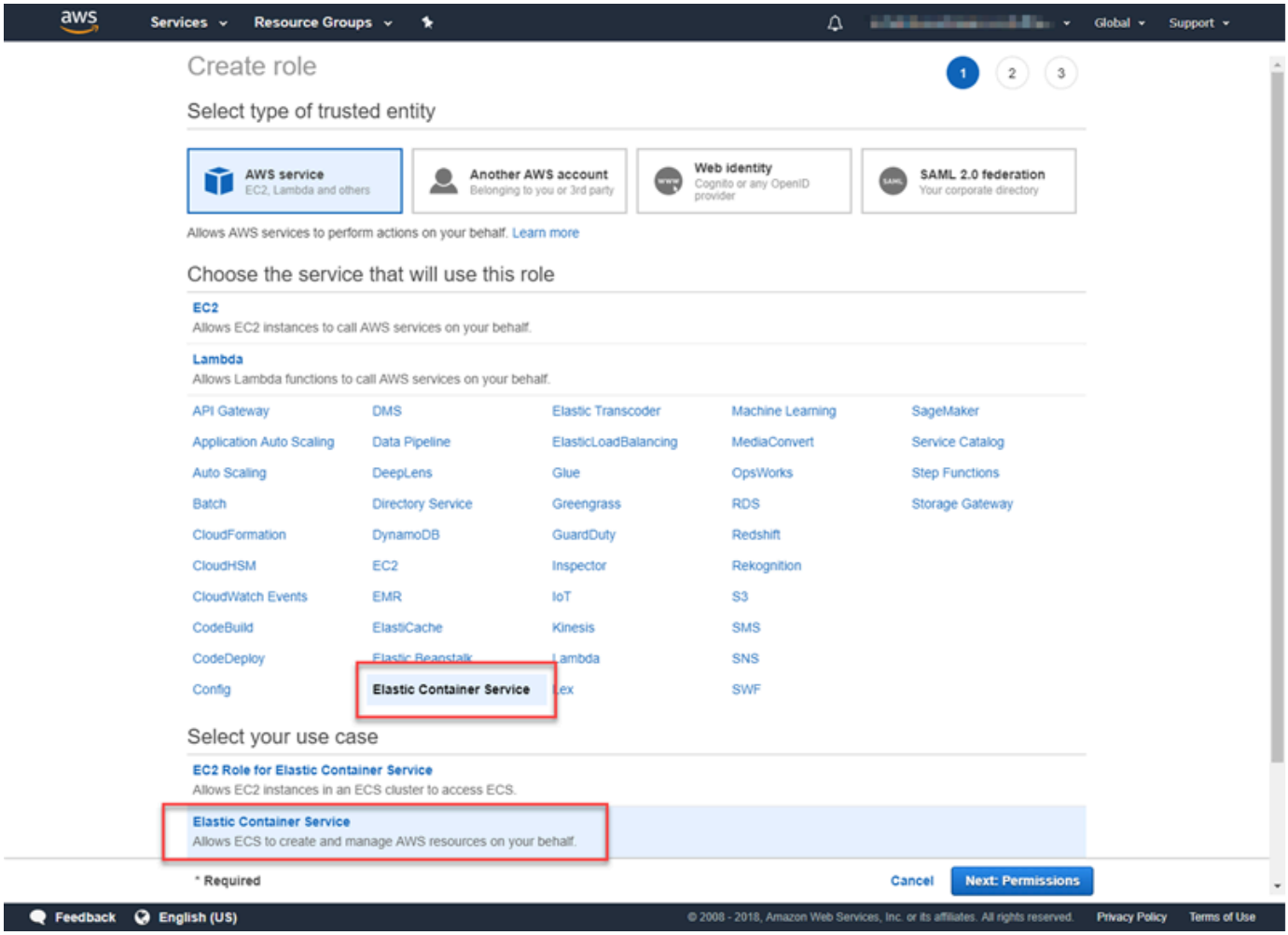
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "dynamodb:BatchGetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem"
            ],
            "Resource": "*"
        }
    ]
}
```

Création des rôles et autorisations ECS

Pour les autorisations associées au rôle ECS, nous vous recommandons de commencer par la politique d'ContainerServiceRoleautorisations AmazonEC2. Utilisez la procédure suivante pour créer un rôle et associer cette politique d'autorisation. Utilisez la console IAM pour afficher le plus grand nombre up-to-date d'autorisations de cette politique.

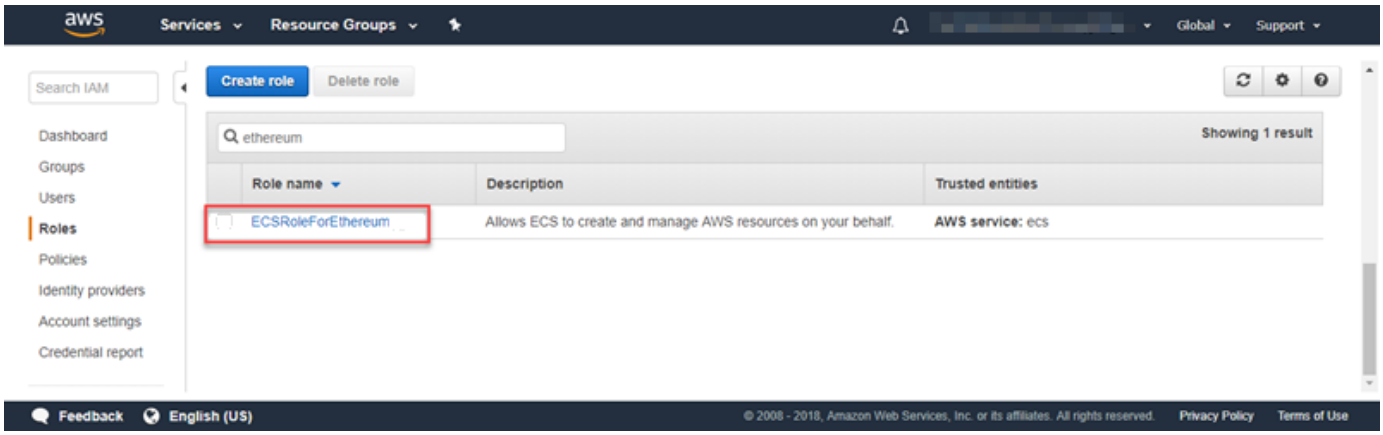
Pour créer le rôle IAM pour Amazon ECS

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Rôles, puis Créer un rôle.
3. Sous Select type of trusted entity (Sélectionner le type d'entité de confiance), choisissez AWS service (Service AWS).
4. Sous Choisir le service qui utilisera ce rôle, choisissez Elastic Container Service.
5. Sous Select your use case (Sélectionnez votre cas d'utilisation), choisissez Elastic Container Service, Next:Permissions (Suivant : Autorisations).

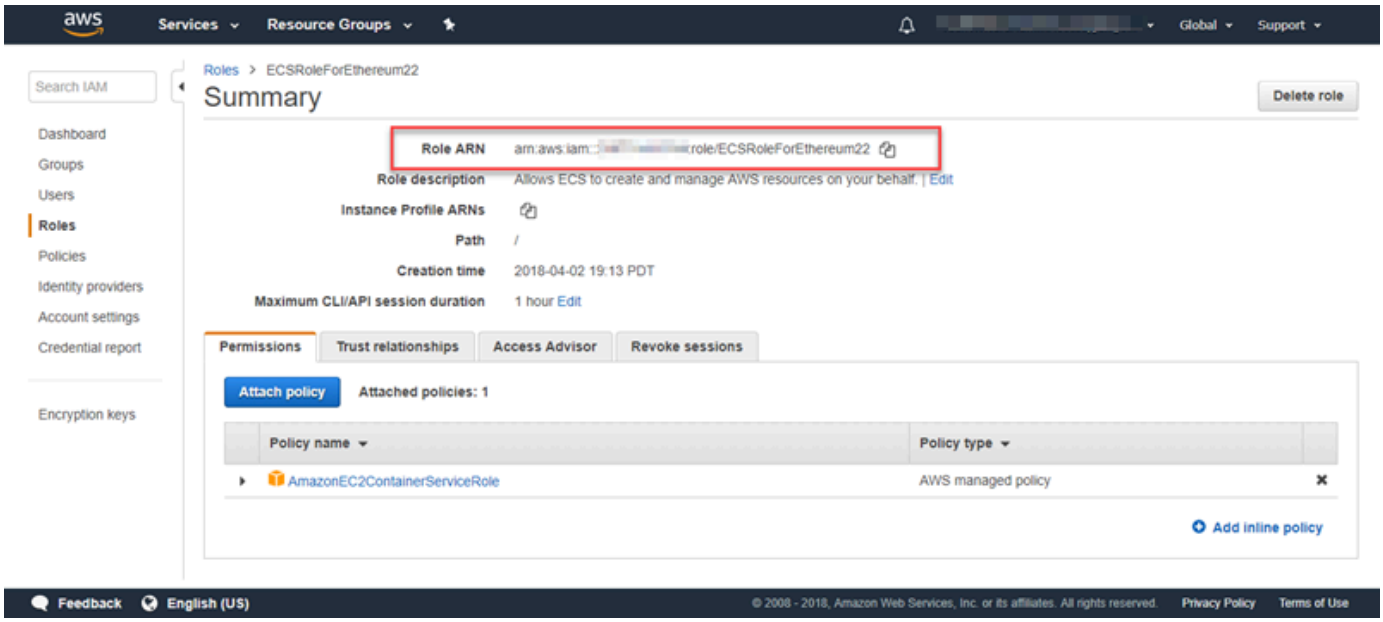


6. Pour la politique d'autorisations, laissez la politique par défaut (AmazonEC2 ContainerServiceRole) sélectionnée et choisissez Next:Review.
7. Dans Nom du rôle, entrez une valeur qui vous aide à identifier le rôle, telle que ECS RoleForEthereum. Pour Role Description (Description du rôle), saisissez un résumé. Notez le nom du rôle pour plus tard.

8. Sélectionnez Créer un rôle.
9. Sélectionnez le rôle que vous venez de créer dans la liste. Si votre compte comporte de nombreux rôles, vous pouvez rechercher le nom du rôle.



10. Copiez la valeur Role ARN (ARN du rôle) et enregistrez-la afin de pouvoir la copier à nouveau. Vous aurez besoin de cet ARN lorsque vous créerez le réseau Ethereum.



Connexion aux ressources Ethereum

Une fois que la pile racine que vous créez avec le modèle indique CREATE_COMPLETE, vous pouvez vous connecter aux ressources Ethereum à l'aide de la AWS CloudFormation console. Comment vous connecter dépend de la plateforme de conteneur que vous choisissez, ECS ou docker-local :

- ECS —L'onglet Output de la pile racine fournit des liens vers les services exécutés sur l'Application Load Balancer. Ces URL ne sont pas directement accessibles pour des raisons de sécurité. Pour vous connecter, vous pouvez configurer et utiliser un hôte bastion pour les connexions proxy avec celles-ci. Pour plus d'informations, consultez Connexions proxy à l'aide d'un hôte Bastion ci-dessous.
- docker-local —Vous vous connectez en utilisant l'adresse IP de l'instance EC2 hébergeant les services Ethereum, comme indiqué ci-dessous. Utilisez la console EC2 pour trouver l'*adresse-IP-ec2* de l'instance créée par le modèle.
  - EthStats–*Utilisez l'adresse IP HTTP ://EC2*
  - EthExplorer–*Utilisez http ://EC2-IP-Address:8080*
  - EthJsonRpc–*Utilisez http ://EC2-IP-Address:8545*

  Si vous avez spécifié un sous-réseau public pour l'Ethereum Network Subnet ID (ID sous-réseau Ethereum) (Liste de sous-réseaux VPC à utiliser dans le modèle), vous pouvez vous connecter directement. Votre client doit être une source fiable de trafic entrant pour SSH (port 22), ainsi que pour les ports répertoriés. Ceci est déterminé par le groupe de sécurité EC2 que vous avez spécifié à l'aide du modèle AWS Blockchain pour Ethereum.

  Si vous avez spécifié un sous-réseau privé, vous pouvez configurer et utiliser un hôte bastion pour les connexions proxy à ces adresses. Pour plus d'informations, consultez Connexions proxy à l'aide d'un hôte Bastion ci-dessous.

Connexions proxy à l'aide d'un hôte Bastion

Dans certaines configurations, les services Ethereum peuvent ne pas être accessibles au public. Dans ces cas, vous pouvez vous connecter aux ressources Ethereum via un hôte bastion. Pour plus d'informations sur les hôtes bastion, consultez Architecture d'hôtes bastion Linux dans le guide Quick Start de l'hôte bastion Linux.

L'hôte du bastion est une instance EC2. Assurez-vous que les conditions suivantes sont remplies :

- L'instance EC2 de l'hôte Bastion se trouve dans un sous-réseau public sur lequel l'attribution automatique d'une adresse IP publique est activée et qui possède une passerelle Internet.
- L'hôte Bastion possède la paire de clés qui autorise les connexions SSH.
- L'hôte Bastion est associé à un groupe de sécurité qui autorise le trafic SSH entrant en provenance des clients qui se connectent.

- Le groupe de sécurité attribué aux hôtes Ethereum (par exemple, l'Application Load Balancer si ECS est la plate-forme de conteneur, ou l'instance EC2 hôte si docker-local est la plate-forme de conteneur) autorise le trafic entrant sur tous les ports à partir de sources au sein du VPC.

Une fois qu'un hôte bastion est configuré, assurez-vous que les clients qui se connectent utilisent l'hôte bastion comme proxy. L'exemple suivant montre comment configurer une connexion proxy à l'aide de Mac OS. Remplacez *BastionIP* par l'adresse IP de l'instance EC2 de l'hôte bastion et *MySshKey.pem* par le fichier de paire de clés que vous avez copié sur l'hôte bastion.

Sur la ligne de commande, tapez ce qui suit :

```
ssh -i mySshKey.pem  ec2-user@BastionIP -D 9001
```

Cela configure la redirection de port pour le port 9001 sur la machine locale vers l'hôte Bastion.

Configurez ensuite votre navigateur ou votre système pour utiliser le proxy SOCKS pour`localhost:9001`. Par exemple, à l'aide de Mac OS, sélectionnez System Preferences (Préférences système), Network (Réseau), Advanced (Paramètres avancés), puis sélectionnez SOCKS proxy (Proxy SOCKS) et saisissez localhost:9001

Dans FoxyProxy Standard avec Chrome, sélectionnez Plus d'outils, Extensions. Sous FoxyProxy Standard, sélectionnez Détails, Options d'extension, Ajouter un nouveau proxy. Sélectionnez Manual Proxy Configuration (Configuration manuelle du proxy). Pour Host or IP Address (Hôte ou adresse IP), saisissez localhost et pour Port, saisissez 9001. Sélectionnez SOCKS proxy? (Proxy SOCKS ?), Save (Enregistrer).

Vous devriez maintenant être en mesure de vous connecter aux adresses d'hôte Ethereum répertoriées dans la sortie du modèle.

Utilisation du modèle AWS Blockchain pour Hyperledger Fabric

Hyperledger Fabric est un framework de blockchain qui exécute des contrats intelligents appelés chaincode, écrits en Go. Vous pouvez créer un réseau privé avec Hyperledger Fabric, en limitant le nombre de pairs autorisés à se connecter au réseau et à y participer. Pour plus d'informations sur Hyperledger Fabric, consultez la documentation d'Hyperledger Fabric. Pour plus d'informations sur le chaincode, consultez la rubrique Chaincode pour les développeurs dans la documentation d'Hyperledger Fabric.

Le modèle AWS Blockchain pour Hyperledger Fabric ne prend en charge qu'une plate-forme de conteneurs locale à Docker, ce qui signifie que les conteneurs Hyperledger Fabric sont déployés sur une seule instance EC2.

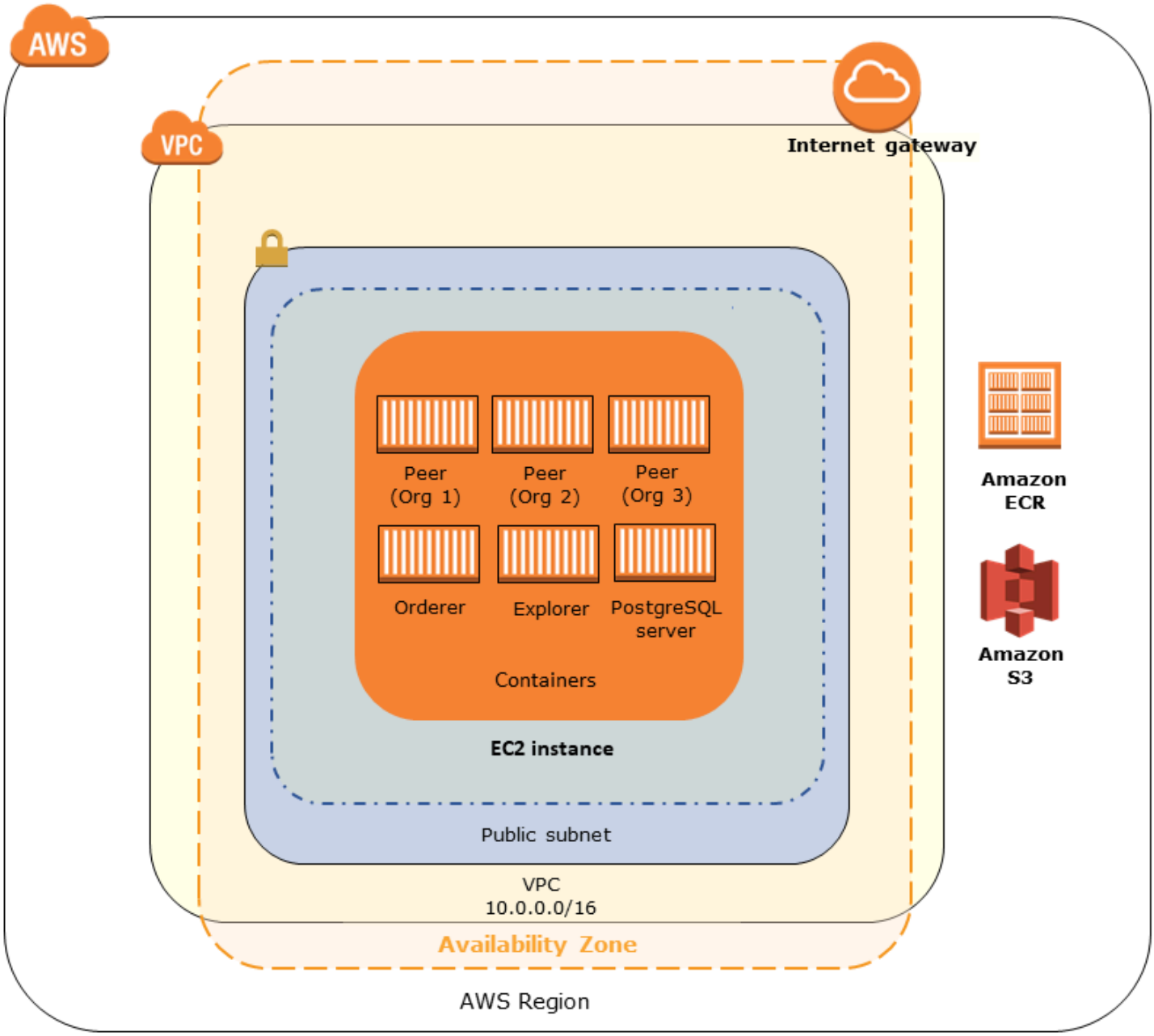
Liens pour le lancement

Consultez Getting Started with AWS Blockchain Templates pour obtenir des liens permettant de lancer AWS CloudFormation dans des régions spécifiques à l'aide des modèles Hyperledger Fabric.

Modèle de blockchain AWS pour les composants Hyperledger Fabric

Le modèle AWS Blockchain pour Hyperledger Fabric crée une instance EC2 avec Docker et lance un réseau Hyperledger Fabric en utilisant des conteneurs sur cette instance. Le réseau comprend un service de commande et trois organisations, chacune dotée d'un service pair. Le modèle lance également un conteneur Hyperledger Explorer, qui vous permet de parcourir les données de la blockchain. Un conteneur de serveur PostgreSQL est lancé pour prendre en charge Hyperledger Explorer.

Le schéma suivant illustre un réseau Hyperledger Fabric créé à l'aide du modèle :



Prérequis

Avant de lancer un réseau Hyperledger Fabric à l'aide d'un modèle, assurez-vous que les exigences suivantes sont satisfaites :

- Le principe IAM (utilisateur ou groupe) que vous utilisez doit être autorisé à utiliser tous les services associés.
- Vous devez avoir accès à une paire de clés que vous pouvez utiliser pour accéder aux instances EC2 (par exemple, via SSH). La clé doit exister dans la même région que l'instance.

- Vous devez disposer d'un profil d'instance EC2 associé à une politique d'autorisation autorisant l'accès à Amazon S3 et à Amazon Elastic Container Registry (Amazon ECR) pour extraire des conteneurs. Pour obtenir un exemple de politique d'autorisations, consultez Exemple d'autorisations IAM pour le profil d'instance EC2.
- Vous devez disposer d'un réseau Amazon VPC avec un sous-réseau public ou d'un sous-réseau privé avec une passerelle NAT et une adresse IP élastique pour pouvoir accéder à Amazon S3 AWS CloudFormation et Amazon ECR.
- Vous devez disposer d'un groupe de sécurité EC2 avec des règles entrantes autorisant le trafic SSH (port 22) à partir des adresses IP ayant besoin de se connecter à l'instance à l'aide de SSH, idem pour les clients ayant besoin de se connecter à Hyperledger Explorer (port 8080).

Exemple d'autorisations IAM pour le profil d'instance EC2

Vous spécifiez un ARN de profil d'instance EC2 comme l'un des paramètres lorsque vous utilisez le modèle AWS Blockchain pour Hyperledger Fabric. Utilisez la déclaration de stratégie suivante comme point de départ pour la stratégie d'autorisations attachée à ce rôle EC2 et au profil d'instance.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:ListImages",
                "ecr:DescribeImages",
                "ecr:BatchGetImage",
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Connexion aux ressources d'Hyperledger Fabric

Une fois que la pile racine que vous créez avec le modèle indique CREATE_COMPLETE, vous pouvez vous connecter aux ressources Hyperledger Fabric sur l'instance EC2. Si vous avez spécifié un sous-réseau public, vous pouvez vous connecter à l'instance EC2 comme n'importe quelle autre instance EC2. Pour plus d'informations, consultez Connexion à votre instance Linux à l'aide du protocole SSH dans le Guide de l'utilisateur Amazon EC2 pour instances Linux.

Si vous avez spécifié un sous-réseau privé, vous pouvez configurer et utiliser un hôte bastion pour établir des connexions par proxy aux ressources Hyperledger Fabric. Pour plus d'informations, consultez Connexions proxy à l'aide d'un hôte Bastion ci-dessous.

> **Note**
>
> Vous remarquerez peut-être que le modèle attribue une adresse IP publique à l'instance EC2 hébergeant les services Hyperledger Fabric ; toutefois, cette adresse IP n'est pas accessible au public car les politiques de routage du sous-réseau privé que vous spécifiez n'autorisent pas le trafic entre cette adresse IP et les sources publiques.

Connexions proxy à l'aide d'un hôte Bastion

Dans certaines configurations, les services Hyperledger Fabric peuvent ne pas être accessibles au public. Dans ces cas, vous pouvez vous connecter aux ressources d'Hyperledger Fabric via un hôte bastion. Pour plus d'informations sur les hôtes bastion, consultez Architecture d'hôtes bastion Linux dans le guide Quick Start de l'hôte bastion Linux.

L'hôte du bastion est une instance EC2. Assurez-vous que les conditions suivantes sont remplies :

- L'instance EC2 de l'hôte Bastion se trouve dans un sous-réseau public sur lequel l'attribution automatique d'une adresse IP publique est activée et qui possède une passerelle Internet.
- L'hôte Bastion possède la paire de clés qui autorise les connexions SSH.
- L'hôte Bastion est associé à un groupe de sécurité qui autorise le trafic SSH entrant en provenance des clients qui se connectent.
- Le groupe de sécurité attribué aux hôtes Hyperledger Fabric (par exemple, l'Application Load Balancer si ECS est la plate-forme de conteneur, ou l'instance EC2 hôte si docker-local est la plate-forme de conteneur) autorise le trafic entrant sur tous les ports en provenance de sources au sein du VPC.

Une fois qu'un hôte bastion est configuré, assurez-vous que les clients qui se connectent utilisent l'hôte bastion comme proxy. L'exemple suivant montre comment configurer une connexion proxy à l'aide de Mac OS. Remplacez *BastionIP* par l'adresse IP de l'instance EC2 de l'hôte bastion et *MySshKey.pem* par le fichier de paire de clés que vous avez copié sur l'hôte bastion.

Sur la ligne de commande, tapez ce qui suit :

```
ssh -i mySshKey.pem  ec2-user@BastionIP -D 9001
```

Cela configure la redirection de port pour le port 9001 sur la machine locale vers l'hôte Bastion.

Configurez ensuite votre navigateur ou votre système pour utiliser le proxy SOCKS pour`localhost:9001`. Par exemple, à l'aide de Mac OS, sélectionnez System Preferences (Préférences système), Network (Réseau), Advanced (Paramètres avancés), puis sélectionnez SOCKS proxy (Proxy SOCKS) et saisissez localhost:9001

Dans FoxyProxy Standard avec Chrome, sélectionnez Plus d'outils, Extensions. Sous FoxyProxy Standard, sélectionnez Détails, Options d'extension, Ajouter un nouveau proxy. Sélectionnez Manual Proxy Configuration (Configuration manuelle du proxy). Pour Host or IP Address (Hôte ou adresse IP), saisissez localhost et pour Port, saisissez 9001. Sélectionnez SOCKS proxy? (Proxy SOCKS ?), Save (Enregistrer).

Vous devriez maintenant être en mesure de vous connecter aux adresses hôtes Hyperledger Fabric répertoriées dans la sortie du modèle.

Historique du document

Le tableau suivant décrit les modifications apportées à la documentation de ce guide.

Dernière mise à jour de la documentation : 1 mai 2019

Modification	Description	Date
Arrêt des modèles AWS Blockchain Templates.	Les modèles AWS Blockchain Templates ont été abandonnés le 30 avril 2019. Aucune autre mise à jour de ce service ou de cette documentation justificative ne sera apportée. Pour une expérience optimale de Managed BlockchainAWS, nous vous recommandons d'utiliser Amazon Managed Blockchain (AMB).	1er mai 2019
Mises à jour de l'hôte Bastion.	Modification du didacticiel de mise en route et des conditions préalables d'Ethereum pour l'ajout d'un hôte bastion, ce qui permet d'accéder aux ressources web fournies via l'équilibreur de charge interne lors de l'utilisation de la plateforme ECS et de l'instance EC2 lors de l'utilisation de docker-local.	3 mai 2018
Guide créé.	Nouveau guide du développeur destiné à soutenir la publication initiale des modèles AWS Blockchain Templates.	19 avril 2018

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le Glossaire AWS dans la Référence Glossaire AWS.