



Guia do Desenvolvedor

Modelos do AWS Blockchain



Modelos do AWS Blockchain: Guia do Desenvolvedor

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

.....	iv
O que são os AWS Blockchain Templates?	1
Como começar	2
Sou proficiente em AWS e blockchain	2
Sou proficiente em AWS e novato em blockchain	3
Sou iniciante em AWS e proficiente com blockchain	3
Sou novo na AWS e no blockchain	3
Serviços relacionados	3
Configurar	5
Cadastre-se na AWS	5
Criar um usuário do IAM	6
Criar um par de chaves	8
Conceitos básicos	10
Configurar pré-requisitos	11
Crie uma VPC e sub-redes	11
Criar security groups	15
Crie um perfil do IAM para o Amazon ECS e um perfil de instância do EC2	18
Criar um bastion host	24
Criar a rede Ethereum	25
Conecte-se EthStats e EthExplorer usando o Bastion Host	28
Limpar recursos	31
AWS Blockchain Templates e recursos	33
AWS Blockchain Template para Ethereum	33
Links para execução	33
Opções do Ethereum	34
Pré-requisitos	37
Conectando-se aos recursos do Ethereum	46
AWS Blockchain Template para o Hyperledger Fabric	47
Links para execução	48
AWS Blockchain Template para componentes do Hyperledger Fabric	48
Pré-requisitos	49
Conectando-se aos recursos do Hyperledger Fabric	51
Histórico do documento	53
AWS Glossário	54

O AWS Blockchain Templates foi descontinuado em 30 de abril de 2019. Nenhuma atualização adicional será feita para este serviço ou nesta documentação de suporte. Para obter a melhor experiência com Managed Blockchain no AWS, recomendamos que você use o [Amazon Managed Blockchain \(AMB\)](#). Para saber mais sobre como começar a usar o Amazon Managed Blockchain, veja o nosso [workshop sobre o Hyperledger Fabric](#) ou o nosso [blog sobre a implantação de um nó Ethereum](#). Se você tiver dúvidas sobre o AMB ou precisar de mais suporte, [entre em contato com AWS Support](#) ou com a sua equipe de contas AWS.

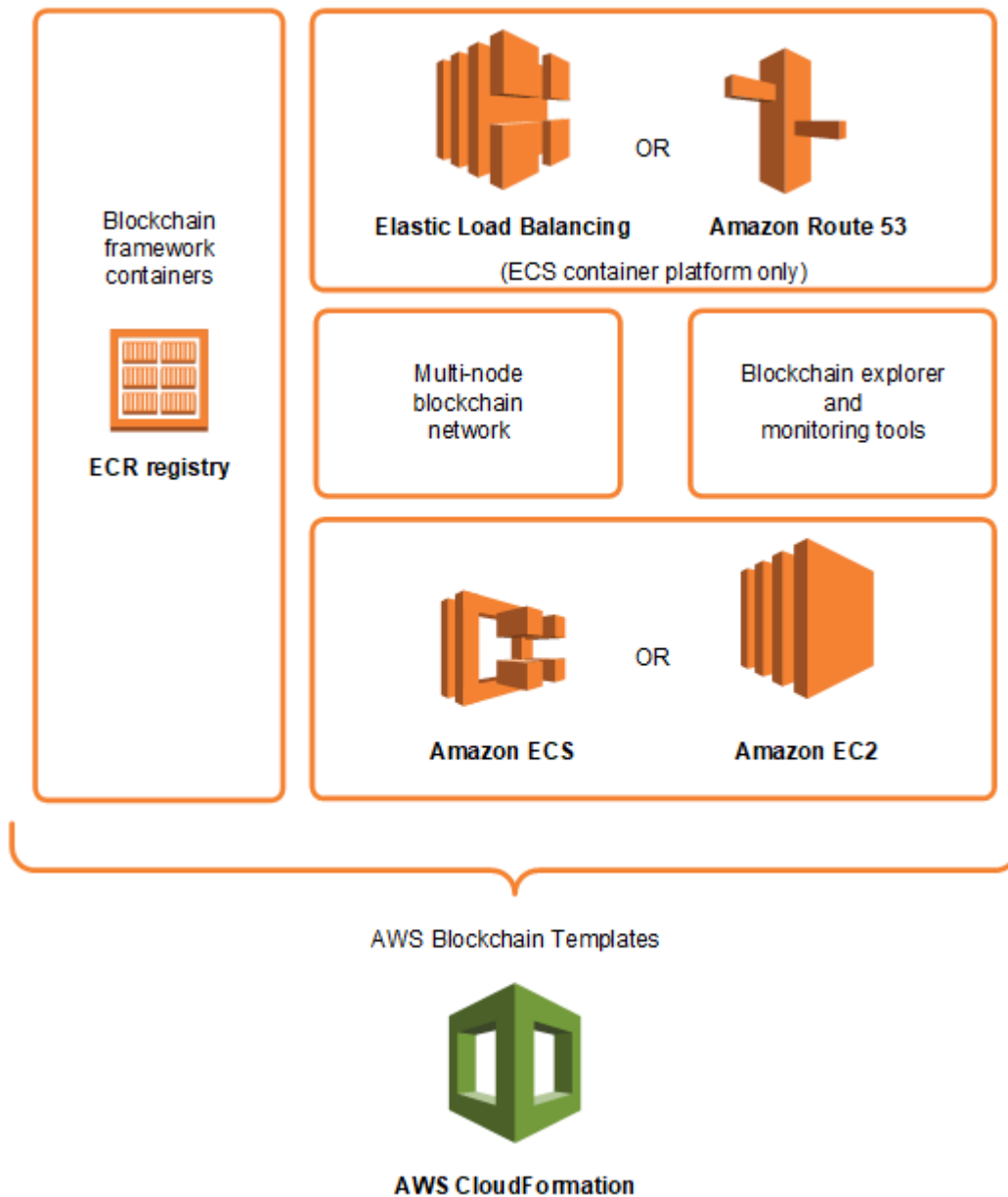
As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

O que são os AWS Blockchain Templates?

Os AWS Blockchain Templates ajuda você a criar e implementar rapidamente redes de blockchain na AWS usando diferentes estruturas de blockchain. Blockchain é uma tecnologia de banco de dados que mantém um conjunto cada vez maior de transações e contratos inteligentes protegidos contra violações e revisões por meio de criptografia.

Uma rede blockchain é uma peer-to-peer rede que melhora a eficiência e a imutabilidade das transações para processos de negócios, como pagamentos internacionais, gerenciamento da cadeia de suprimentos, registro de terras, financiamento coletivo, governança, transações financeiras e muito mais. Isso permite que pessoas e organizações que talvez nem se conheçam confiem e verifiquem o registro da transação de forma independente.

Use os AWS Blockchain Templates para configurar e executar pilhas AWS CloudFormation para criar redes de blockchain. Os recursos e serviços da AWS que você usa dependem do AWS Blockchain Template que você escolheu e das opções que especificou. Para obter informações sobre modelos disponíveis e seus atributos, consulte [AWS Blockchain Templates e recursos](#). Os componentes fundamentais de uma rede de blockchain em AWS, criados usando o AWS Blockchain Templates são mostrados no diagrama a seguir.



Como começar

O melhor lugar para começar depende do seu nível de conhecimento sobre blockchains e AWS—particularmente os serviços relacionados aos AWS Blockchain Templates.

Sou proficiente em AWS e blockchain

Comece com o tópico em [AWS Blockchain Templates e recursos](#) sobre o framework que deseja usar. Use os links para executar o AWS Blockchain Template e configure a rede de blockchain ou faça download dos templates para verificá-los por conta própria.

Sou proficiente em AWS e novato em blockchain

Comece com o tutorial [Conceitos básicos do AWS Blockchain Templates](#). Ele orientará você na criação de uma rede introdutória de blockchains Ethereum por meio de configurações padrão. Quando você terminar, consulte [AWS Blockchain Templates e recursos](#) para ver uma visão geral sobre estruturas e links para saber mais sobre opções e recursos de atributos.

Sou iniciante em AWS e proficiente com blockchain

Comece com [Configurando os AWS Blockchain Templates](#). Isso ajuda você a se preparar com os conceitos básicos de AWS, como uma conta e um perfil de usuário. Em seguida, veja o tutorial [Conceitos básicos do AWS Blockchain Templates](#). Esse tutorial orientará você na criação de uma rede introdutória de blockchain Ethereum. Mesmo que você não vá usar o Ethereum, terá experiência prática na configuração de serviços relacionados. Essa experiência é útil para todas as estruturas de blockchain. Por fim, consulte o tópico na seção [AWS Blockchain Templates e recursos](#) para sua estrutura.

Sou novo na AWS e no blockchain

Comece com [Configurando os AWS Blockchain Templates](#). Isso ajuda você a se preparar com os conceitos básicos de AWS, como uma conta e um perfil de usuário. Em seguida, consulte o tutorial [Conceitos básicos do AWS Blockchain Templates](#). Esse tutorial orientará você na criação de uma rede introdutória de blockchain Ethereum. Aproveite o tempo para explorar os links e saber mais sobre os serviços AWS e o Ethereum.

Serviços relacionados

Dependendo das opções selecionadas, o AWS Blockchain Templates poderá usar os seguintes recursos dos serviços da AWS para implementar o blockchain:

- Amazon EC2 — Fornece capacidade computacional para a sua rede de blockchain. Para obter mais informações, consulte o [Guia do usuário do Amazon EC2 para Instâncias do Linux](#).
- O Amazon ECS — Organiza a implantação de contêineres entre instâncias do EC2 em um cluster para sua rede de blockchain, se você optar por usá-lo. Para obter mais informações, consulte o [Guia do desenvolvedor do serviço Elastic Container da Amazon](#).
- Amazon VPC — Fornece acesso à rede para os recursos da Ethereum criados por você. Você pode personalizar a configuração para acessibilidade e segurança. Para obter mais informações, consulte o [Guia do desenvolvedor do Amazon VPC](#).

- Application Load Balancing — Serve como um único ponto de contato para acesso a interfaces de usuário disponíveis e descoberta de serviços internos ao usar o Amazon ECS como uma plataforma de contêiner. Para obter mais informações, consulte [O que é um Application Load Balancer](#) no Guia do Usuário de Application Load Balancers.

Configurando os AWS Blockchain Templates

Antes de começar com o AWS Blockchain Templates, conclua as tarefas a seguir:

- [Cadastre-se na AWS](#)
- [Criar um usuário do IAM](#)
- [Criar um par de chaves](#)

Esses são pré-requisitos fundamentais para todas as configurações de blockchain. Além disso, a rede de blockchains escolhida pode ter pré-requisitos que variam de acordo com ambiente e as opções de configuração desejadas. Para obter mais informações, consulte a seção relevante do seu modelo de blockchain em [AWS Blockchain Templates e recursos](#).

Para step-by-step obter instruções sobre como configurar os pré-requisitos para uma rede Ethereum privada usando um cluster Amazon ECS, consulte. [Conceitos básicos do AWS Blockchain Templates](#)

Cadastre-se na AWS

Ao se cadastrar na AWS, sua conta da AWS será automaticamente cadastrada em todos os serviços. Você será cobrado apenas pelos serviços que usar.

Se já tiver uma conta da AWS, passe para a próxima tarefa. Se você ainda não possuir uma conta da AWS, use o procedimento a seguir para criar uma.

Para criar uma conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de aplicação envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se cadastra em uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e atributos na conta. Como prática recomendada de segurança, [atribua acesso administrativo a um usuário administrativo](#) e use somente o usuário raiz para realizar as [tarefas que exigem acesso do usuário raiz](#).

Anote o número da sua conta da AWS. Ele será solicitado ao criar um usuário do IAM; na próxima tarefa.

Criar um usuário do IAM

Serviços na AWS exigem que você forneça credenciais quando acessá-las, para que o serviço possa determinar se você tem permissão para acessar seus recursos. O console requer sua senha. Você pode criar chaves de acesso para sua conta da AWS para acessar a interface de linha de comando ou a API. Contudo, não recomendamos que você acesse a AWS usando as credenciais de sua conta da AWS; em vez disso, recomendamos que você use o AWS Identity and Access Management (IAM). Crie um usuário do IAM e, em seguida, adicione o usuário a um grupo do IAM com permissões administrativas ou conceda permissões administrativas a esse usuário. Então, você pode acessar a AWS usando um URL especial e as credenciais do usuário do IAM.

Se tiver se cadastrado na AWS, mas não criou um usuário do IAM para você, crie um usando o console do IAM. Se você já tiver um usuário do IAM, ignore essa etapa.

Para criar um usuário administrador, selecione uma das opções a seguir.

Selecionar uma forma de gerenciar o administrador	Para	Por	Você também pode
Centro de Identidade do IAM (Recomendado)	Use credenciais de curto prazo para acessar a AWS. Isso está de acordo com as práticas recomendadas de segurança. Para obter informações sobre as práticas recomenda	Seguindo as instruções em Conceitos básicos no AWS IAM Identity Center Guia do usuário.	Para configurar o acesso programático, consulte Configurar a AWS CLI para usar o AWS IAM Identity Center no AWS Command Line Interface Guia do usuário.

Selecione uma forma de gerenciar o administrador	Para	Por	Você também pode
	<p>das, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.</p>		
No IAM (Não recomendado)	Use credenciais de curto prazo para acessar a AWS.	Seguindo as instruções em Criar o seu primeiro usuário administrador e um grupo de usuários do IAM no Guia do usuário do IAM.	Para configurar o acesso programático, consulte Gerenciamento de chaves de acesso de usuários do IAM no Guia do usuário do IAM.

Para fazer login como esse novo usuário do IAM, faça logout do AWS Management Console e use o seguinte URL, em que `your_aws_account_id` é o número de sua conta da AWS sem hifens (por exemplo, se o número da conta da AWS é 1234-5678-9012, o ID da conta da AWS é 123456789012):

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

Insira o nome e a senha de usuário do IAM que você acabou de criar. Quando você está conectado, a barra de navegação exibe "`your_user_name @ your_aws_account_id`".

Se você não quiser que a URL da página de cadastro contenha o ID da sua conta da AWS, crie um alias da conta. No painel do IAM, escolha Criar apelido da conta e insira um apelido, como o nome de sua empresa. Para fazer o login depois de criar o alias de uma conta, use o seguinte URL:

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

Para verificar o link de cadastro para usuários do IAM para a conta, abra o console do IAM e marque IAM users sign-in link no painel.

Para mais informações, consulte o [Guia do usuário do AWS Identity and Access Management](#).

Criar um par de chaves

A AWS usa a criptografia de chave pública para proteger as informações de login das instâncias em uma rede de blockchain. Você especifica o nome do par de chaves ao usar cada AWS Blockchain Template. Em seguida, você poderá usar o par de chaves para acessar instâncias diretamente, por exemplo, para efetuar login usando o SSH.

Se você já tiver um par de chaves na Região correta, você pode ignorar esta etapa. Se ainda não tiver criado um par de chaves, será possível criar um usando o console do Amazon EC2. Crie o par de chaves na mesma Região usada para executar a rede de Ethereum. Para obter mais informações, consulte [Regiões e zonas de disponibilidade](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Para criar um par de chaves

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. Na barra de navegação, selecione uma região para o par de chaves. Você pode selecionar qualquer Região que estiver disponível, independentemente da sua localização. No entanto, pares de chaves são específicos de uma Região. Por exemplo, se você planeja executar uma instância na região do Leste dos EUA (Ohio), você deverá criar um par de chaves para a instância na mesma região.
3. No painel de navegação, escolha Pares de chaves, Criar par de chaves.
4. Em Key pair name (Nome do par de chaves), insira um nome para o novo par de chaves. Escolha um nome que seja fácil de lembrar, como o nome do usuário do IAM seguido por -key-pair, mais o nome da região. Por exemplo, me-key-pair-useast2. Escolha Criar.
5. O arquivo de chave privada é baixado automaticamente pelo navegador. O nome do arquivo base é o especificado por você como sendo o nome do par de chaves, e a extensão do nome do arquivo é .pem. Salve o arquivo de chave privada em um lugar seguro.

 **Important**

Esta é a única chance de você salvar o arquivo de chave privada. Você precisa fornecer o nome do par de chaves ao iniciar a rede de Ethereum.

Para obter mais informações, consulte [Pares de chaves do Amazon EC2](#) no Guia do usuário do Amazon EC2 para instâncias do Linux. Para obter mais informações sobre como se conectar a instâncias do EC2 usando o par de chaves, consulte [Conectar-se à sua instância do Linux](#) no Guia do Usuário do Amazon EC2 para Instâncias do Linux.

Conceitos básicos do AWS Blockchain Templates

Este tutorial demonstra como usar o AWS Blockchain Template para que o Ethereum crie uma rede de blockchain privada na AWS por meio do AWS CloudFormation. A rede criada tem dois clientes Ethereum e um minerador operando em instâncias EC2 da Amazon em um cluster ECS da Amazon. O ECS da Amazon executa esses serviços em contêineres do Docker retirados do ECR da Amazon. Antes de começar este tutorial, será útil se você conhecer as redes de blockchain e os serviços da AWS envolvidos nela, embora isso não seja necessário.

Este tutorial pressupõe que você configurou os pré-requisitos gerais abordados em [Configurando os AWS Blockchain Templates](#). Além disso, são necessários alguns recursos de AWS, como uma rede VPC da Amazon e permissões específicas para perfis do IAM, antes de usar o modelo.

O tutorial demonstra como configurar esses pré-requisitos. Fizemos escolhas de configuração, mas elas não são prescritivas. Contanto que você atenda aos pré-requisitos, será possível fazer outras escolhas de configuração com base nas necessidades de seu aplicativo e ambiente. Para obter informações sobre os recursos e pré-requisitos gerais de cada modelo e fazer download de modelos ou executá-los diretamente no AWS CloudFormation, consulte [AWS Blockchain Templates e recursos](#).

Neste tutorial, os exemplos usam a região Oeste dos EUA (Oregon) (us-west-2), mas é possível usar qualquer região que ofereça suporte ao AWS Blockchain Templates:

- Região Oeste dos EUA (Oregon) (us-west-2)
- Região Leste dos EUA (Norte da Virgínia) (us-east-1)
- Região Leste dos EUA (Ohio) (us-east-2)

Note

Executar um modelo em uma Região não listada acima abrirá recursos na região Leste dos EUA (Norte da Virgínia) (us-east-1).

O AWS Blockchain Template para o Ethereum que você configura utilizando este tutorial cria os seguintes recursos:

- Instâncias do EC2 sob demanda de tipo e número especificados. O tutorial usa o tipo de instância t2.medium padrão.
- Um Application Load Balancer interno.

Seguindo o tutorial, são fornecidas etapas para limpar recursos criados por você.

Tópicos

- [Configurar pré-requisitos](#)
- [Criar a rede Ethereum](#)
- [Conecte-se EthStats e EthExplorer usando o Bastion Host](#)
- [Limpar recursos](#)

Configurar pré-requisitos

A configuração do AWS Blockchain Template especificada neste tutorial requer que você faça o seguinte:

- [Crie uma VPC e sub-redes](#)
- [Criar security groups](#)
- [Crie um perfil do IAM para o Amazon ECS e um perfil de instância do EC2](#)
- [Criar um bastion host](#)

Crie uma VPC e sub-redes

O AWS Blockchain Template para o Ethereum lança recursos da em uma rede virtual definida por você usando a Amazon Virtual Private Cloud (Amazon VPC). A configuração especificada neste tutorial cria um Application Load Balancer, que exige duas sub-redes públicas em zonas de disponibilidade diferentes. Além disso, uma sub-rede privada é necessária para as instâncias de contêiner, e a sub-rede deve estar na mesma zona de disponibilidade que o Application Load Balancer. Primeiro, use o assistente da VPC para criar uma sub-rede pública e uma sub-rede privada na mesma zona de disponibilidade. Depois disso, crie uma segunda sub-rede pública nesta VPC em uma zona de disponibilidade diferente.

Para obter mais informações, consulte [O que é a Amazon VPC?](#) no Guia do usuário da Amazon VPC.

Use o console Amazon VPC (<https://console.aws.amazon.com/vpc/>) para criar o endereço IP elástico, a VPC e a sub-rede, conforme descrito abaixo.

Para criar um endereço IP elástico

1. Abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.
2. Escolha Elastic IPs (IPs elásticos), Allocate new address (Alocar novo endereço), Allocate (Alocar).
3. Anote o endereço IP elástico criado e escolha Close (Fechar).
4. Na lista de endereços IP elásticos, localize Allocation ID (ID de alocação) para o endereço IP elástico criado anteriormente. Você o usará mais tarde, quando criar a VPC.

Para criar a VPC

1. Na barra de navegação, selecione uma região para a VPC. As VPCs são específicas para a região. Portanto, selecione a mesma região na qual você criou o par de chaves e executou a pilha do Ethereum. Para ter mais informações, consulte [Criar um par de chaves](#).
2. No painel da VPC, escolha Start VPC Wizard.
3. Na página Step 1: Select a VPC Configuration (Etapa 1: Selecionar uma configuração da VPC), escolha VPC with Public and Private Subnets (VPC com sub-redes públicas e privadas) e Select (Selecionar).
4. Na página Step 2: VPC with Public and Private Subnets (Etapa 2: VPC com sub-redes públicas e privadas), deixe IPv4 CIDR block (Bloco CIDR IPv4) e IPv6 CIDR block (Bloco CIDR IPv6) com seus valores padrão. Em VPC name (Nome da VPC), insira um nome fácil de lembrar.
5. Em Public subnet's IPv4 CIDR (CIDR IPv4 da sub-rede pública), deixe o valor padrão. Em Availability Zone (Zona de disponibilidade), escolha uma zona. Em Public subnet name (Nome da sub-rede pública), insira um nome fácil de lembrar.

Essa sub-rede é especificada como uma das duas primeiras sub-redes do Application Load Balancer ao usar o modelo.

Anote a zona de disponibilidade desta sub-rede porque você deverá selecionar a mesma zona de disponibilidade para a sub-rede privada e uma diferente para a outra sub-rede pública.

6. Em Private subnet's IPv4 CIDR (CIDR IPv4 da sub-rede privada), deixe o valor padrão. Em Availability Zone (Zona de disponibilidade), selecione a mesma zona de disponibilidade da etapa anterior. Em Private subnet name (Nome da sub-rede privada), insira um nome fácil de lembrar.

7. Em Elastic IP Allocation ID (ID de alocação do IP elástico), selecione o endereço IP elástico criado anteriormente.
8. Mantenha os valores padrão das demais configurações.
9. Escolha Create VPC (Criar VPC).

O exemplo abaixo mostra uma VPC EthereumNetworkVPC com uma sub-rede pública EthereumPubSub1 e uma sub-rede privada 1. EthereumPvtSub A sub-rede pública usa a zona de disponibilidade us-west-2a.

Step 2: VPC with Public and Private Subnets

IPv4 CIDR block:* (65531 IP addresses available)

IPv6 CIDR block: No IPv6 CIDR Block
 Amazon provided IPv6 CIDR block

VPC name:

Public subnet's IPv4 CIDR:* (251 IP addresses available)

Availability Zone:* ▼

Public subnet name:

Private subnet's IPv4 CIDR:* (251 IP addresses available)

Availability Zone:* ▼

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT gateway ([NAT gateway rates apply](#)). [Use a NAT instance instead](#)

Elastic IP Allocation ID:*

Service endpoints

Enable DNS hostnames:* Yes No

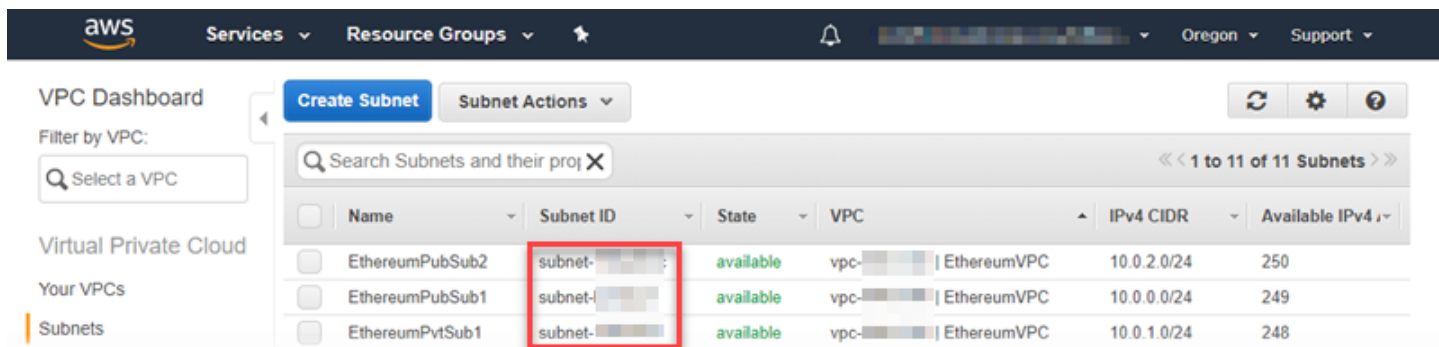
Hardware tenancy:* ▼

Para criar a segunda sub-rede pública em uma zona de disponibilidade diferente

1. Escolha Subnets (Sub-redes) e selecione a sub-rede pública criada anteriormente na lista. Selecione a guia Route Table (Tabela de rotas) e anote o ID da Route table (Tabela de rotas). Especifique essa mesma tabela de rotas para a segunda sub-rede pública abaixo.
2. Selecione Create Subnet.

3. Em Name tag (Tag de nome), insira um nome para a sub-rede. Use esse nome mais tarde quando criar o bastion host nessa rede.
4. Em VPC, selecione a VPC criada anteriormente.
5. Em Availability Zone (Zona de disponibilidade), selecione uma zona diferente daquela selecionada para a primeira sub-rede pública.
6. Em IPv4 CIDR block (Bloco CIDR IPv4), insira 10.0.2.0/24.
7. Escolha Yes, Create (Sim, criar). A sub-rede é adicionada à lista de sub-redes.
8. Com a sub-rede selecionada na lista, escolha Subnet Actions (Ações de sub-rede) e Modify auto-assign IP settings (Modificar configurações de IP de atribuição automática). Selecione Auto-assign IPs (Atribuir IPs automaticamente), Save (Salvar) e Close (Fechar). Isso permite que o bastion host obtenha um endereço IP público quando você o cria nessa sub-rede.
9. Na guia Route Table (Tabela de rotas), escolha Edit (Editar). Em Change to (Alterar para), selecione o ID da tabela de rotas anotado anteriormente e escolha Save (Salvar).

Agora você deverá ver três sub-redes para a VPC que criou anteriormente. Anote os nomes e IDs das sub-redes para que você possa especificá-los usando o modelo.



Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
EthereumPubSub2	subnet-...	available	vpc-... EthereumVPC	10.0.2.0/24	250
EthereumPubSub1	subnet-...	available	vpc-... EthereumVPC	10.0.0.0/24	249
EthereumPvtSub1	subnet-...	available	vpc-... EthereumVPC	10.0.1.0/24	248

Criar security groups

Os grupos de segurança agem como firewalls, controlando o tráfego de entrada e saída para recursos. Ao usar o modelo para criar uma rede de Ethereum em um cluster do Amazon ECS, você especifica dois grupos de segurança:

- Um security group para instâncias do EC2 que controla o tráfego de e para instâncias do EC2 no cluster
- Um grupo de segurança para o Application Load Balancer que controla o tráfego entre o Application Load Balancer, instâncias do EC2 e o bastion host. Associe esse grupo de segurança com o bastion host também.

Cada grupo de segurança tem regras que permitem a comunicação entre o Application Load Balancer e as instâncias do EC2, além de outras regras mínimas. Isso requer que os grupos de segurança referenciem-se uns aos outros. Por esse motivo, você primeiro cria os grupos de segurança e, então, você os atualiza com as regras apropriadas.

Para criar dois grupos de segurança

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação, escolha Security Groups, Create Security Group (Criar Security Group).
3. Em Security group name (nome do grupo de segurança), insira um nome para o grupo de segurança que seja fácil de identificar e diferente do outro, como EthereumEC2-SG ou EthereumALB-SG. Você usará esses nomes depois. Em Description (Descrição), insira um breve resumo.
4. Em VPC, selecione a VPC criada anteriormente.
5. Escolha Criar.
6. Repita as etapas acima para criar o outro grupo de segurança.

Adicione regras de entrada ao grupo de segurança para instâncias do EC2

1. Selecione o grupo de segurança para instâncias do EC2 que você criou anteriormente
2. Na guia Entrada, escolha Editar.
3. Para Tipo, escolha Todo o tráfego. Em Source (Origem), mantenha a opção Custom (Personalizada) selecionada e escolha o grupo de segurança que você está editando na lista, por exemplo, EthereumEC2-SG. Isso permite que as instâncias do EC2 no grupo de segurança comuniquem-se umas com as outras.
4. Escolha Add Rule.
5. Para Tipo, escolha Todo o tráfego. Em Source (Origem), mantenha a opção Custom (Personalizada) selecionada. Escolha o grupo de segurança do Application Load Balancer na lista, por exemplo, EthereumALB-SG. Isso permite que as instâncias do EC2 no grupo de segurança se comuniquem com o Application Load Balancer.
6. Escolha Salvar.

Adicionar entrada e editar regras de saída para o grupo de segurança do Application Load Balancer

1. Selecione o grupo de segurança para os Application Load Balancers que você criou anteriormente
2. Na guia Inbound (Entrada), escolha Edit (Editar) e adicione as seguintes regras de entrada:
 - a. Para Tipo, escolha Todo o tráfego. Em Source (Origem), mantenha a opção Custom (Personalizada) selecionada e escolha o grupo de segurança que você está editando na lista, por exemplo, EthereumALB-SG. Isso permite que o Application Load Balancer comunique-se consigo mesmo e com o bastion host.
 - b. Escolha Add Rule.
 - c. Para Tipo, escolha Todo o tráfego. Em Source (Origem), mantenha a opção Custom (Personalizada) selecionada. Escolha o grupo de segurança para instâncias do EC2 na lista, por exemplo, EthereumEC2-SG. Isso permite que as instâncias do EC2 no grupo de segurança se comuniquem com o Application Load Balancer e o bastion host.
 - d. Escolha Add Rule.
 - e. Para Tipo, escolha SSH. Em Source (Origem), selecione My IP (Meu IP), que detecta o CIDR IP do computador e o insere.

Important

Essa regra permite que o bastion host aceite o tráfego SSH do seu computador, permitindo que o computador use o bastion host para visualizar interfaces da Web e se conectar a instâncias do EC2 na rede Ethereum. Para permitir que outras pessoas se conectem à rede Ethereum, adicione-as como origens a essa regra. Permita apenas tráfego de entrada para origens confiáveis.

- f. Escolha Salvar.
3. Na guia Outbound (Saída), escolha Edit (Editar) e exclua a regra que foi criada automaticamente para permitir o tráfego de saída para todos os endereços IP.
4. Escolha Add Rule.
5. Para Tipo, escolha Todo o tráfego. Em Destination (Destino), mantenha a opção Custom (Personalizada) selecionada. Escolha o grupo de segurança para instâncias do EC2 na lista. Isso permite conexões de saída do Application Load Balancer e o bastion host para instâncias do EC2 na rede Ethereum.
6. Escolha Add Rule.

7. Para Tipo, escolha Todo o tráfego. Em Destination (Destino), mantenha a opção Custom (Personalizada) selecionada. Escolha o grupo de segurança que você está editando na lista, por exemplo, EthereumALB-SG. Isso permite que o Application Load Balancer comunique-se consigo mesmo e com o bastion host.
8. Escolha Salvar.

Crie um perfil do IAM para o Amazon ECS e um perfil de instância do EC2

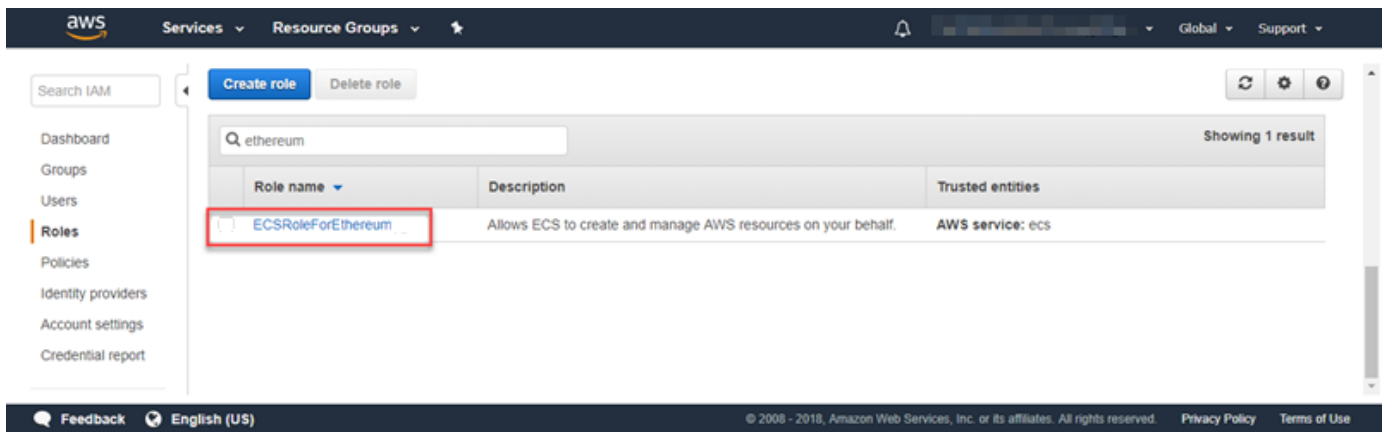
Ao usar este modelo, você especificará um perfil do IAM para o Amazon ECS e um perfil de instância do EC2. As políticas de permissões anexadas a essas funções permitem que os recursos e instâncias AWS no cluster interajam com os outros recursos AWS. Para obter mais informações, consulte [Perfis do IAM](#) no Guia do usuário do IAM. Você configura o perfil do IAM para o Amazon ECS e o perfil de instância EC2 usando o console do IAM (<https://console.aws.amazon.com/iam/>).

Para criar os perfis do IAM do Amazon ECS

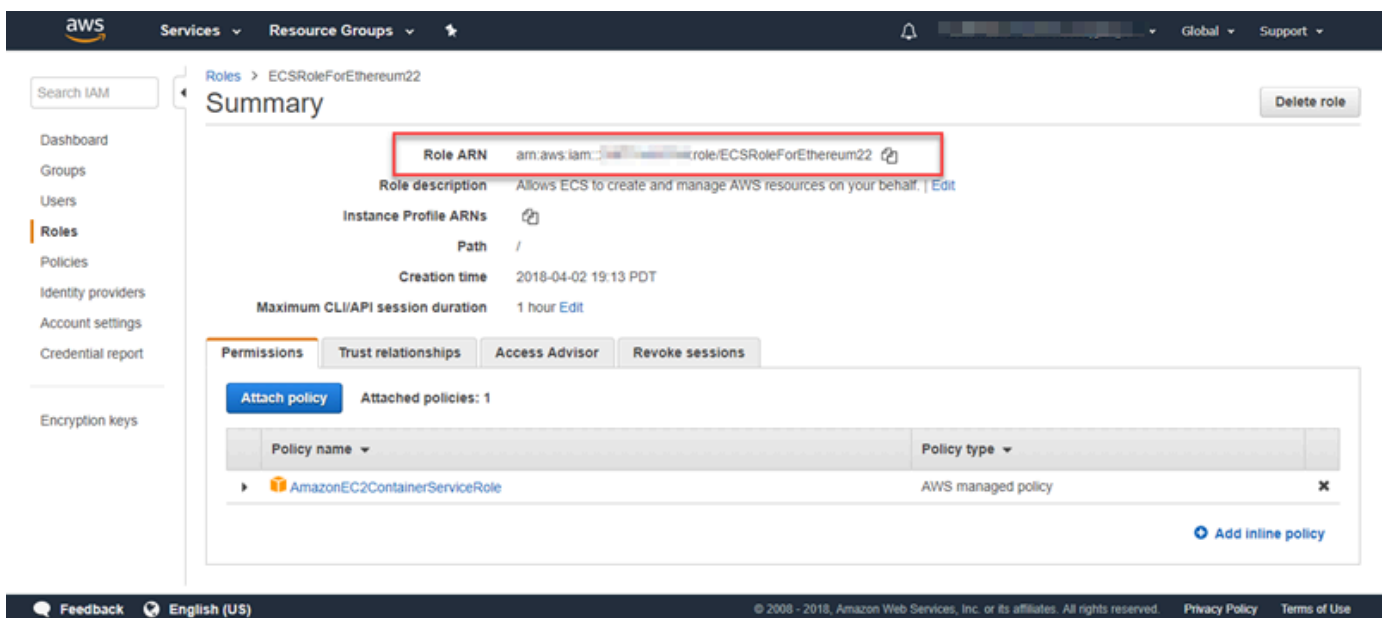
1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Roles (Funções) e Create Role (Criar função).
3. Em Select type of trusted entity (Selecionar tipo de entidade confiável), escolha AWS service (Produto da AWS).
4. Em Choose the service that will use this role (Escolher o serviço que usará esta função), escolha Elastic Container Service.
5. Em Select your use case (Selecionar seu caso de uso), escolha Elastic Container Service, Next:Permissions (Próximo: permissões).

The screenshot shows the AWS IAM console 'Create role' page. The 'Select type of trusted entity' step is active, with 'AWS service' selected. The 'Choose the service that will use this role' section lists various AWS services, with 'Elastic Container Service' highlighted in a red box. The 'Select your use case' section shows 'Elastic Container Service' selected, also highlighted in a red box. The 'Next: Permissions' button is visible at the bottom right.

6. Em Política de permissões, deixe a política padrão (AmazonEC2 ContainerServiceRole) selecionada e escolha Next:Review.
7. Em Nome da função, insira um valor que o ajude a identificar a função, como ECS. RoleForEthereum Em Role Description (Descrição da função), digite um resumo breve. Anote o nome da função para mais tarde.
8. Selecione Criar perfil.
9. Na lista, selecione a função que você acabou de criar. Caso sua conta tenha várias funções, você pode pesquisar o nome da função.



10. Copie o valor de Role ARN (ARN da função) e salve-o para copiá-lo novamente depois. Você precisará dele quando for criar a rede de Ethereum.



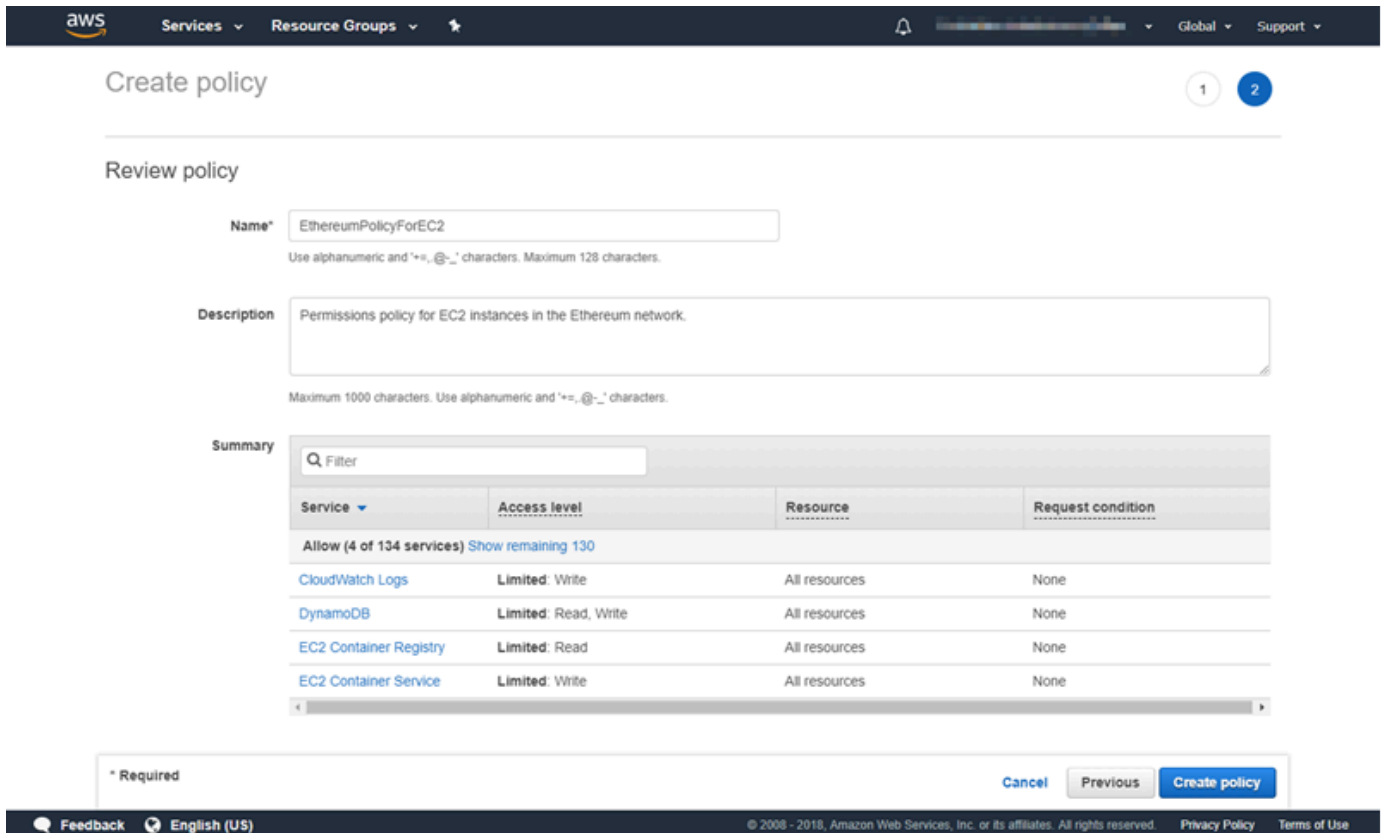
O perfil da instância do EC2 que você especifica no modelo é assumido por instâncias do EC2 na rede de Ethereum para interagir com outros serviços da AWS. Você cria uma política de permissões para a função, cria a função (que cria automaticamente um perfil de instância com o mesmo nome) e, em seguida, anexa a política de permissões à função.

Para criar um perfil da instância do EC2

1. No painel de navegação, escolha Políticas, Create policy.
2. Escolha JSON e substitua a instrução de política padrão pela seguinte política JSON:


```
{
  "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. Escolha Revisar política.
4. Em Nome, insira um valor que o ajude a identificar essa política de permissões, por exemplo, EthereumPolicyForEC2. Em Description (Descrição), insira um breve resumo. Escolha Criar política.



Create policy 1 2

Review policy

Name* EthereumPolicyForEC2
Use alphanumeric and '+, @, _' characters. Maximum 128 characters.

Description Permissions policy for EC2 instances in the Ethereum network.
Maximum 1000 characters. Use alphanumeric and '+, @, _' characters.

Summary

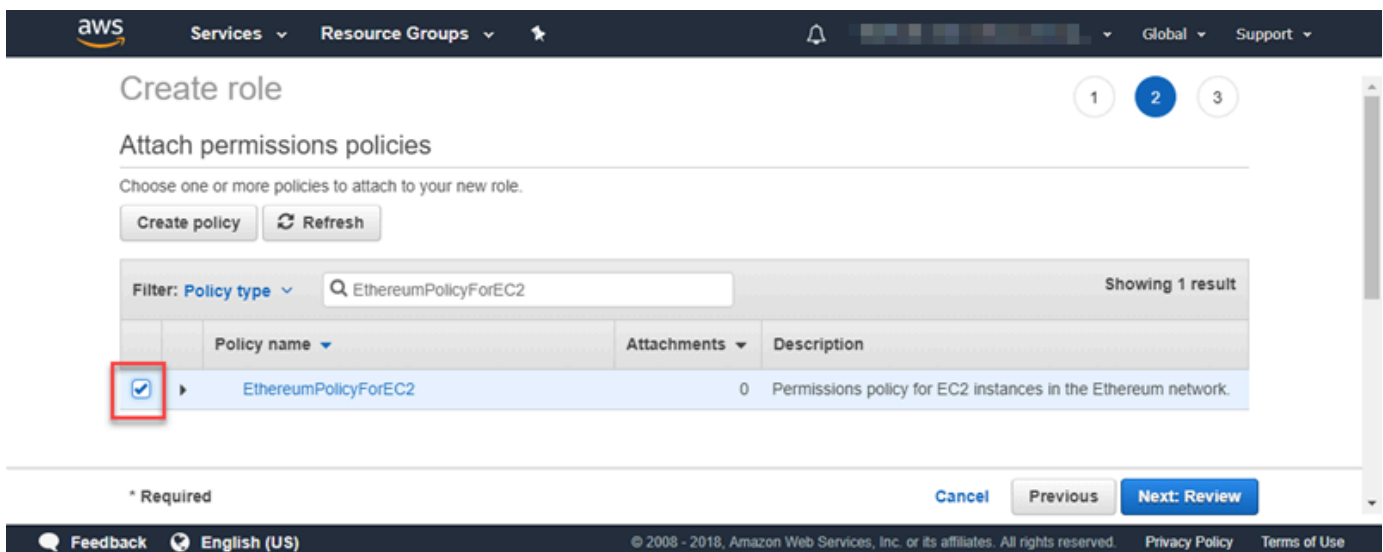
Filter

Service	Access level	Resource	Request condition
Allow (4 of 134 services) Show remaining 130			
CloudWatch Logs	Limited: Write	All resources	None
DynamoDB	Limited: Read, Write	All resources	None
EC2 Container Registry	Limited: Read	All resources	None
EC2 Container Service	Limited: Write	All resources	None

* Required Cancel Previous Create policy

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

5. Selecione Roles (Funções), Create role (Criar função).
6. Escolha EC2 e Next: Permissions (Próximo: permissões).
7. No campo Pesquisar, insira o nome da política de permissões que você criou anteriormente, por exemplo, EthereumPolicyForEC2.
8. Escolha a marca de seleção da política que criada anteriormente e escolha Next: Review (Próximo: revisão).



Create role 1 2 3

Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy Refresh

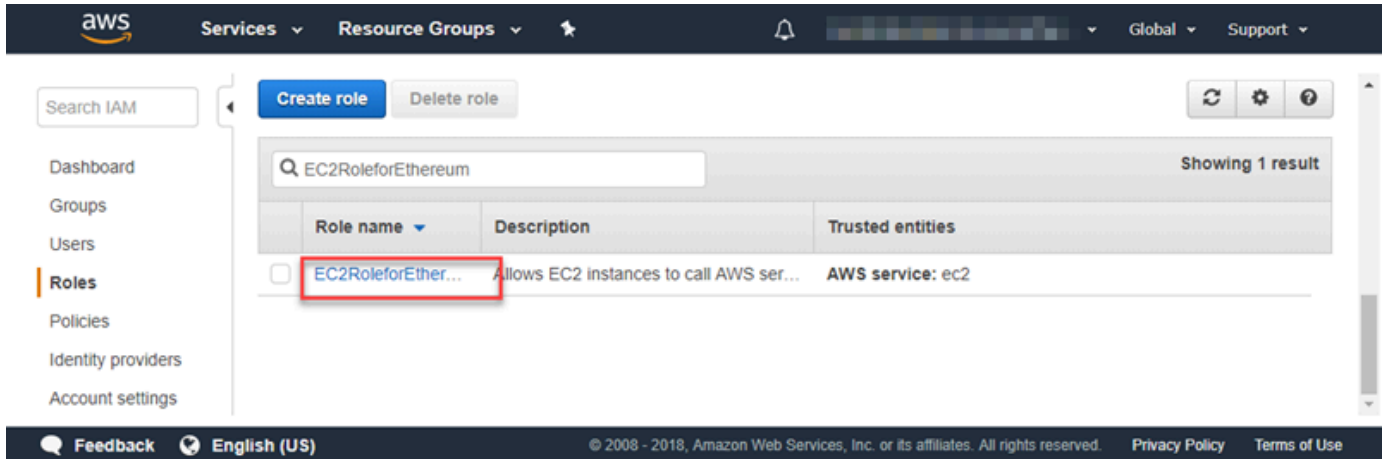
Filter: Policy type Showing 1 result

Policy name	Attachments	Description
<input checked="" type="checkbox"/> EthereumPolicyForEC2	0	Permissions policy for EC2 instances in the Ethereum network.

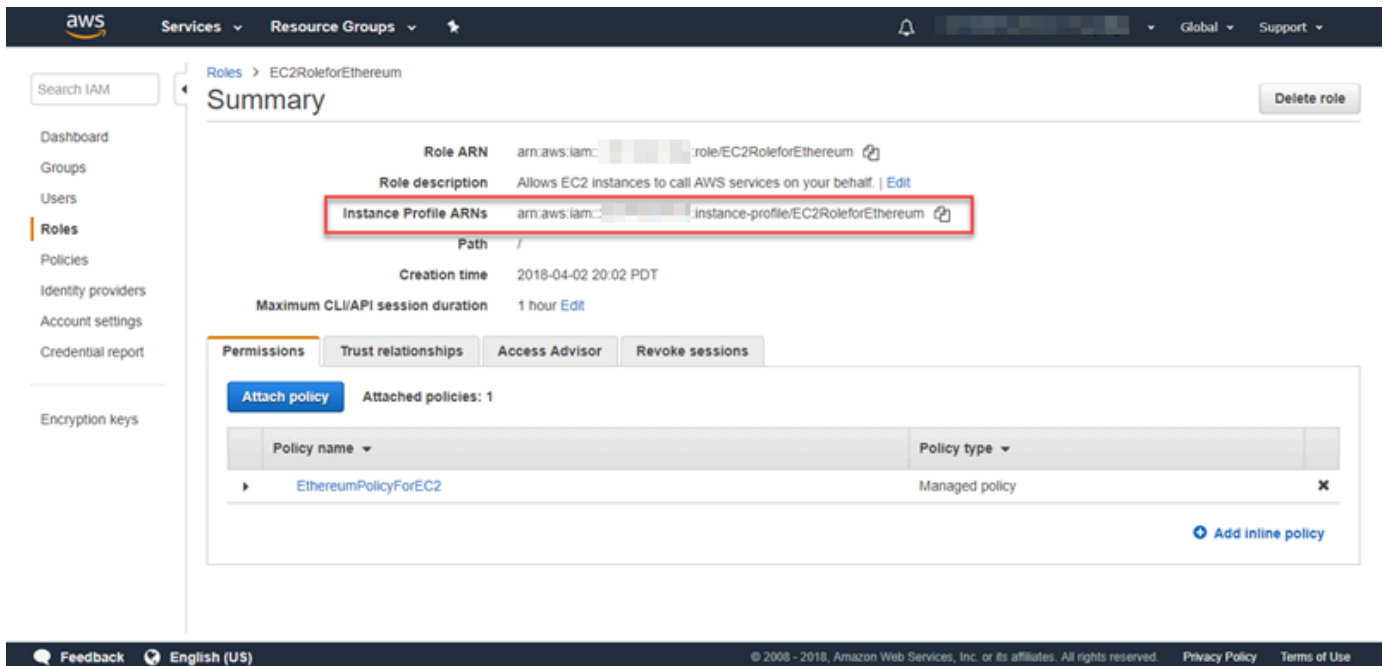
* Required Cancel Previous Next: Review

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Em Nome da função, insira um valor que o ajude a identificar a função, por exemplo, EC2 RoleForEthereum. Em Role description (Descrição da função), insira um resumo e escolha Create role (Criar função).
- Na lista, selecione a função que você acabou de criar. Caso sua conta tenha várias funções, é possível inserir o nome da função no campo Search (Pesquisa).



- Copie o valor de Instance Profile ARN (ARN do perfil de instância) e salve-o para copiá-lo novamente depois. Você precisará dele quando for criar a rede de Ethereum.



Criar um bastion host

Neste tutorial, crie um bastion host. Essa é uma instância do EC2 usada para se conectar às interfaces da Web e instâncias na sua rede Ethereum. Seu único objetivo é encaminhar o tráfego SSH de clientes confiáveis fora da VPC para que eles possam acessar recursos de rede Ethereum.

Configure o bastion host porque o Application Load Balancer criado pelo modelo é interno, o que significa que ele roteia somente endereços IP internos. O bastion host:

- Tem um endereço IP interno que o Application Load Balancer reconhece porque você o inicia na segunda sub-rede pública criada anteriormente.
- Tem um endereço IP público que a sub-rede atribui, que pode ser acessado por origens confiáveis fora da VPC.
- É associado ao grupo de segurança do Application Load Balancer criado anteriormente, que tem uma regra de entrada que permite tráfego SSH (porta 22) de clientes confiáveis.

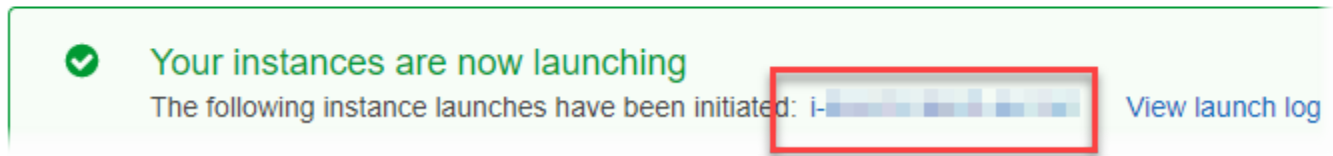
Para acessar a rede Ethereum, clientes confiáveis precisam ser configurados para se conectarem por meio do bastion host. Para ter mais informações, consulte [Conecte-se EthStats e EthExplorer usando o Bastion Host](#). Um bastion host é uma das abordagens. É possível usar qualquer abordagem que forneça acesso de clientes confiáveis a recursos privados em uma VPC.

Como criar um bastion host

1. Siga as cinco primeiras etapas para [Iniciar uma Instância](#) no Guia do usuário do Amazon EC2 para instâncias Linux.
2. Escolha Edit Instance Details (Editar detalhes da instância). Em Network (Rede), escolha a VPC criada anteriormente. Em Subnet (Sub-rede), selecione a segunda sub-rede pública criada anteriormente. Deixe todas as outras configurações com seus valores padrão.
3. Confirme a alteração quando solicitado e escolha Review and Launch (Revisar e executar).
4. Escolha Edit security groups (Editar grupos de segurança). Em Assign a security group (Atribuir um grupo de segurança), escolha Select an existing security group (Selecionar um security group existente).
5. Na lista de grupos de segurança, selecione o grupo de segurança para o Application Load Balancer criado anteriormente e escolha Review and Launch (Revisar e executar).
6. Escolha Executar.

7. Anote o ID da instância. Ele será necessário mais tarde, quando você [Conecte-se EthStats e EthExplorer usando o Bastion Host](#).

Launch Status



Criar a rede Ethereum

A rede Ethereum que você especifica usando o modelo neste tópico executará um pilha AWS CloudFormation que cria um cluster do Amazon ECS de instâncias do EC2 para a rede Ethereum. O modelo depende dos recursos criados anteriormente em [Configurar pré-requisitos](#).

Quando você executa a pilha do AWS CloudFormation usando o modelo, ele cria pilhas aninhadas para algumas tarefas. Depois que eles forem concluídos, será possível se conectar aos recursos fornecidos pelo Application Load Balancer da rede por meio do bastion host e verificar se sua rede Ethereum está em execução e acessível.

Criar a rede Ethereum usando o modelo AWS Blockchain Template para o Ethereum

1. Consulte [Conceitos básicos do AWS Blockchain Templates](#) e abra o AWS Blockchain Template mais recente para o Ethereum no console AWS CloudFormation usando os links rápidos para a sua região da AWS.
2. Insira os valores de acordo com as seguintes diretrizes:
 - Em Stack name (Nome da pilha), insira um nome de fácil identificação. Esse nome é usado dentro dos nomes de recursos que a pilha cria.
 - Em Ethereum Network Parameters (Parâmetros da rede Ethereum) e Private Ethereum Network Parameters (Parâmetros da rede privada Ethereum), mantenha as configurações padrão.

⚠ Warning

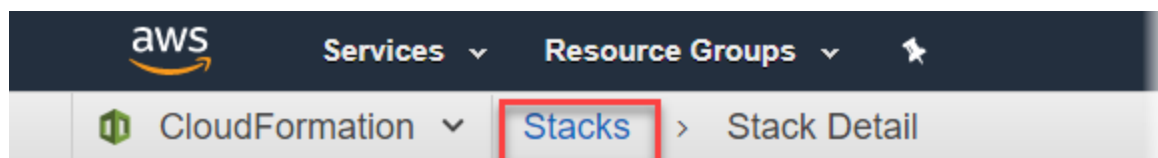
Use as contas padrão e a frase mnemônica associada somente para fins de teste. Não envie Ether real usando o conjunto padrão de contas, pois qualquer pessoa com acesso à frase mnemônica pode acessar ou roubar o Ether das contas. Em vez disso, especifique contas personalizadas para fins de produção. A frase mnemônica associada à conta padrão é `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

- Em Configuração da plataforma, mantenha as configurações padrão, pois isso cria um cluster Amazon ECS de instâncias EC2. A alternativa, `docker-local`, cria uma rede Ethereum usando uma única instância EC2.
- Em EC2 configuration (Configuração do EC2), selecione opções de acordo com as seguintes diretrizes:
 - Em EC2 Key Pair (Par de chaves do EC2), selecione um par de chaves. Para obter mais informações sobre como criar um par de chaves, consulte [Criar um par de chaves](#).
 - Em EC2 Security Group (Grupo de segurança do EC2), selecione o grupo de segurança criado anteriormente em [Criar security groups](#).
 - Em EC2 Instance Profile ARN (ARN do perfil de instância do EC2), insira o ARN do perfil de instância criado anteriormente em [Crie um perfil do IAM para o Amazon ECS e um perfil de instância do EC2](#).
- Em VPC network configuration (Configuração de rede da VPC), selecione opções de acordo com as seguintes diretrizes:
 - Em VPC ID (ID da VPC), selecione o ID da VPC criado anteriormente em [Crie uma VPC e sub-redes](#).
 - Em Ethereum Network Subnet IDs (IDs de sub-rede da rede Ethereum), selecione a única sub-rede privada criada anteriormente no procedimento [To create the VPC](#).
- Em ECS cluster configuration (Configuração de cluster do ECS), mantenha os valores padrão. Isso cria um cluster do ECS de três instâncias do EC2.
- Em Application Load Balancer configuration (ECS only) (Configuração do Application Load Balancer (somente o ECS)), selecione as opções de acordo com as seguintes diretrizes:
 - Em Application Load Balancer Subnet IDs (IDs de sub-rede do Application Load Balancer), selecione duas sub-redes públicas do [list of subnets](#) anotadas anteriormente.

- Em Application Load Balancer Security Group (Grupo de segurança do Application Load Balancer), selecione o grupo de segurança para o Application Load Balancer criado anteriormente em [Criar security groups](#).
 - Para o perfil do IAM, insira o ARN da função do ECS criada anteriormente em [Crie um perfil do IAM para o Amazon ECS e um perfil de instância do EC2](#).
- Em EthStats, selecione as opções de acordo com as seguintes diretrizes:
 - Para Implantar EthStats, deixe a configuração padrão, que é verdadeira.
 - Para Segredo de EthStats conexão, digite um valor arbitrário com pelo menos seis caracteres.
 - Em EthExplorer, deixe a configuração padrão para Implantar EthExplorer, que é verdadeira.
 - Em Other parameters (Outros parâmetros), mantenha o valor padrão para Nested Template S3 URL Prefix (Prefixo de URL do S3 do modelo aninhado) e anote-o. Aqui é onde é possível encontrar modelos aninhados.
3. Mantenha todas as demais configurações com seus valores padrão, marque a caixa de seleção de confirmação e escolha Create (Criar).

É exibida a página Stack Detail (Detalhes da pilha) para a pilha raiz que o AWS CloudFormation executa.

4. Para monitorar o progresso da pilha raiz e das pilhas aninhadas, escolha Stacks (Pilhas).



MyFirstEthereumStack

Stack name: MyFirstEthereumStack

5. Quando todas as pilhas exibirem CREATE_COMPLETE no Status, você poderá se conectar às interfaces de usuário do Ethereum para verificar se a rede está funcionando e acessível. Quando você usa a plataforma de contêiner ECS, URLs para EthStats conexão e EthJson RPC por meio do Application Load Balancer estão disponíveis na guia Saídas da pilha raiz. EthExplorer

⚠ Important

Não será possível se conectar diretamente a esses URLs ou SSH diretamente até que você configure uma conexão de proxy por meio do bastion host em seu computador cliente. Para ter mais informações, consulte [Conecte-se EthStats e EthExplorer usando o Bastion Host](#).

The screenshot shows the AWS CloudFormation console interface. At the top, there are navigation menus for 'Services', 'Resource Groups', and 'Stacks'. Below this, there are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter is set to 'Active' and 'By Stack Name'. A table lists four stacks, with the first one, 'MyFirstEthereumStack', selected and highlighted with a red box. Below the table, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. The 'Outputs' tab is active, showing a table with columns 'Key', 'Value', 'Description', and 'Export Name'. Three outputs are listed: 'EthStatsURL', 'EthExplorerURL', and 'EthJsonRPCURL'. The 'EthStatsURL' value is highlighted with a red box.

Stack Name	Created Time	Status	Description
MyFirstEthereumStack-Ether... NESTED	2018-04-12 13:26:46 UTC-0700	CREATE_COMPLETE	This template creates an AutoScalingGroup of EC2 I...
MyFirstEthereumStack-Ether... NESTED	2018-04-12 13:26:38 UTC-0700	CREATE_COMPLETE	This template creates the ECS cluster and Ethereu...
MyFirstEthereumStack-Ether... NESTED	2018-04-12 13:25:59 UTC-0700	CREATE_COMPLETE	This template deploys an Ethereum cluster on an ex...
MyFirstEthereumStack	2018-04-12 13:25:54 UTC-0700	CREATE_COMPLETE	This template creates an Ethereum network on an A...

Key	Value	Description	Export Name
EthStatsURL	http://MyFir-... us-west-2.elb.amazonaws.com	Visit this URL to see the status of your ...	
EthExplorerURL	http://MyFir-... us-west-2.elb.amazonaws.com:8080	Visit this URL to view transactions on yo...	
EthJsonRPCURL	http://MyFir-... us-west-2.elb.amazonaws.com:8545	Use this URL to access the Geth JSON ...	

Conecte-se EthStats e EthExplorer usando o Bastion Host

Para se conectar aos recursos da Ethereum neste tutorial, configure o encaminhamento de porta SSH (túnel SSH) por meio do bastion host. As instruções a seguir demonstram como fazer isso para que você possa se conectar a URLs EthStats e EthExplorer URLs usando um navegador. Nas instruções abaixo, configure primeiro um proxy SOCKS em uma porta local. Em seguida, você usa uma extensão de navegador, [FoxyProxy](#), para usar essa porta encaminhada para seus URLs de rede Ethereum.

Se você usar Mac OS ou Linux, use um cliente SSH para configurar a conexão de proxy SOCKS para o bastion host. Se você é um usuário do Windows, use PuTTY. Antes de se conectar, verifique se o computador cliente usado está especificado como uma origem permitida para o tráfego SSH de entrada no grupo de segurança para o Application Load Balancer configurado anteriormente.

Como se conectar ao bastion host com encaminhamento de porta SSH usando SSH

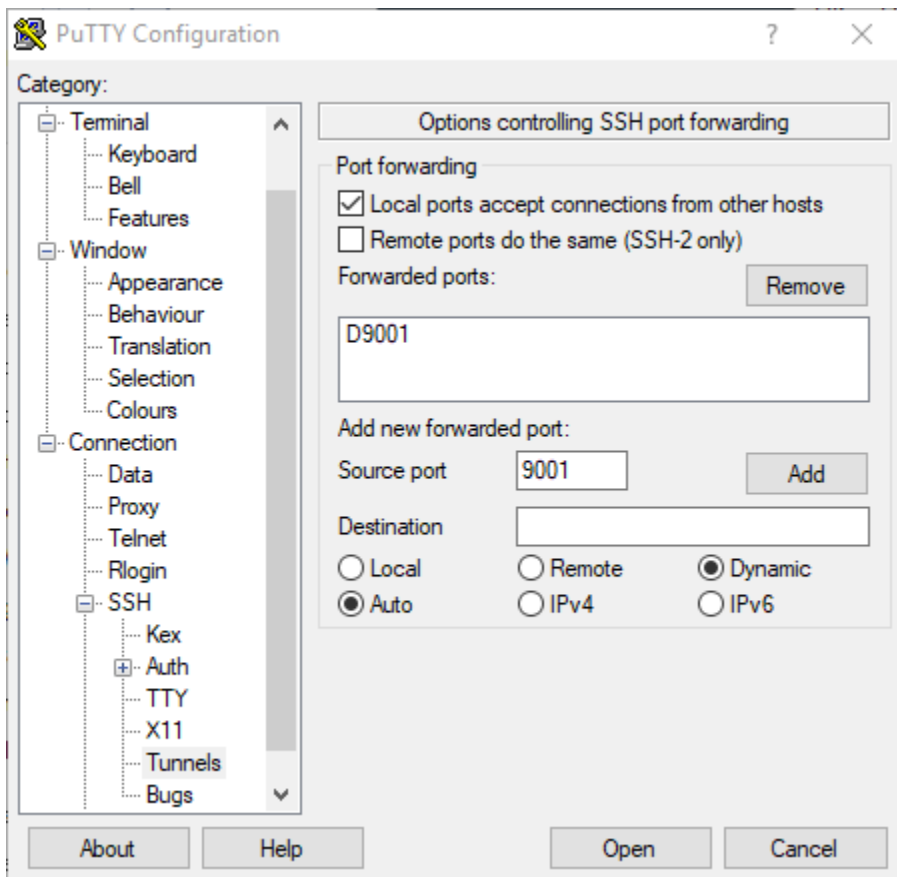
- Siga os procedimentos contidos em [Conectando a sua Instância Linux usando SSH](#) no Guia do usuário do Amazon EC2 para instâncias do Linux. Na etapa 4 do procedimento [Conectando a sua Instância Linux](#), adicione `-D 9001` ao comando SSH, especifique o mesmo par de chaves especificado no AWS Blockchain Template para a configuração do Ethereum e especifique o nome DNS do bastion host.

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

Como conectar-se ao bastion host com encaminhamento de porta SSH usando o PuTTY (Windows)

1. Siga os procedimentos na seção [Conectar-se à sua Instância Linux do Windows usando PuTTY](#) no Guia do Usuário do Amazon EC2 para Instâncias Linux até a etapa 7 do procedimento [Iniciando uma Sessão PuTTY](#), usando o mesmo par de chaves especificado no AWS Blockchain Template para a configuração do Ethereum.
2. Em PuTTY, em Category (Categoria), escolha Connection (Conexão), SSH e Tunnels (Túneis).
3. Em Port forwarding (Encaminhamento de porta), escolha Local ports accept connections from other hosts (Portas locais aceitam conexões de outros hosts).
4. Em Add new forwarded port (Adicionar nova porta encaminhada):
 - a. Em Source port (Porta de origem), digite 9001. Essa é uma porta não utilizada arbitrária que escolhemos, e é possível escolher uma diferente, se necessário.
 - b. Deixe Destination (Destino) em branco.
 - c. Selecione Dynamic (Dinâmico).
 - d. Escolha Adicionar.

Em Forwarded ports (Portas encaminhadas), o D9001 deve ser exibido conforme mostrado abaixo.



5. Escolha Open (Abrir) e autentique para o bastion host conforme exigido pela sua configuração de chave. Deixe a conexão aberta.

Com a conexão PuTTY aberta, agora configure seu sistema ou uma extensão do navegador para usar a porta encaminhada para seus URLs da rede Ethereum. As instruções a seguir se baseiam no uso do FoxyProxy Padrão para encaminhar conexões com base no padrão de URL de EthStats EthExplorer e e na porta 9001, que você estabeleceu anteriormente como a porta encaminhada, mas você pode usar qualquer método de sua preferência.

FoxyProxy Para configurar o uso do túnel SSH para URLs de rede Ethereum

Este procedimento foi escrito com base no Chrome. Se você usa outro navegador, traduza as configurações e a sequência FoxyProxy para a versão desse navegador.

1. Baixe e instale a extensão de navegador FoxyProxy padrão e, em seguida, abra Opções de acordo com as instruções do seu navegador.
2. Escolha Add New Proxy (Adicionar novo proxy).

3. Na guia General (Geral) verifique se o proxy está Enabled (Habilitado) e insira um Proxy Name (Nome do proxy) e Proxy Notes (Anotações do proxy) que o ajudem a identificar essa configuração de proxy.
4. Na guia Proxy Details (Detalhes do proxy), escolha Manual Proxy Configuration (Configuração manual do proxy). Em Host or IP Address (Host ou endereço IP) (ou Server or IP Address (Servidor ou endereço IP) em algumas versões), insira localhost. Em Port (Porta), insira 9001. Selecione SOCKS Proxy?.
5. Na guia URL Pattern (Padrão de URL), escolha Add New Pattern (Adicionar novo padrão).
6. Em Nome do padrão, insira um nome que seja fácil de identificar e, em Padrão de URL, insira um padrão que corresponda a todos os URLs de recursos do Ethereum que você criou com o modelo, por exemplo, http://internal - MyUser -loadB-*. Para obter informações sobre como visualizar URLs, consulte [Ethereum URLs](#).
7. Mantenha as seleções padrão para outras configurações e escolha Save (Salvar).

Agora você pode se conectar aos URLs do Ethereum, que estão disponíveis no CloudFormation console usando a guia Saídas da pilha raiz que você criou com o modelo.

Limpar recursos

O AWS CloudFormation facilita a limpeza de recursos criados pela pilha. Quando você exclui a pilha, todos os recursos que a pilha criou são excluídos.

Para excluir recursos criados pelo modelo

- Abra o console do AWS CloudFormation, selecione a pilha raiz criada anteriormente e escolha Actions (Ações) e Delete (Excluir).

O Status da pilha raiz criada anteriormente e as pilhas aninhadas associadas são atualizados para DELETE_IN_PROGRESS.

Você pode optar por excluir os pré-requisitos criados para a rede de Ethereum.

Exclua a VPC

- Abra o console Amazon VPC, selecione a VPC que você criou anteriormente e escolha Ações, Excluir VPC. Isso também exclui as sub-redes, os security groups e o gateway NAT associados à VPC.

Excluir a função do IAM e o perfil de instância do EC2

- Abra o console do IAM e selecione Funções. Selecione a função do ECS e a função do EC2 criadas anteriormente e escolha Delete (Excluir).

Encerrar a instância do EC2 para o bastion host

- Abra o painel do Amazon EC2, escolha Instâncias em execução, selecione a instância do EC2 criada para o bastion host, escolha Ações, Estado da instância e Encerrar.

AWS Blockchain Templates e recursos

Esta seção fornece links para você começar a criar uma rede de blockchain imediatamente, além de informações sobre opções de configuração e pré-requisitos para configurar a rede no AWS.

Os seguintes modelos estão disponíveis:

- [AWS Blockchain Template para Ethereum](#)
- [AWS Blockchain Template para Hyperledger Fabric](#)

O AWS Blockchain Templates está disponível nas seguintes regiões:

- Região Oeste dos EUA (Oregon) (us-west-2)
- Região Leste dos EUA (Norte da Virgínia) (us-east-1)
- Região Leste dos EUA (Ohio) (us-east-2)

Note

Executar um modelo em uma Região não listada acima abrirá recursos na região Leste dos EUA (Norte da Virgínia) (us-east-1).

Usando o AWS Blockchain Template para Ethereum

O Ethereum é uma estrutura de blockchain que executa contratos inteligentes usando o Solidity, uma linguagem específica do Ethereum. O Homestead é o lançamento mais recente do Ethereum. Para obter mais informações, consulte a [documentação do Ethereum Homestead](#) e a documentação do [Solidity](#).

Links para execução

Consulte [Conceitos básicos do AWS Blockchain Templates](#) para obter links para iniciar o AWS CloudFormation em regiões específicas usando os modelos do Ethereum.

Opções do Ethereum

Ao configurar a rede Ethereum usando o modelo, você faz escolhas que determinam os requisitos subsequentes:

- [Escolher a plataforma do contêiner](#)
- [Escolher uma rede Ethereum Privada ou Pública](#)
- [Alterar as contas padrão e a frase mnemônica](#)

Escolher a plataforma do contêiner

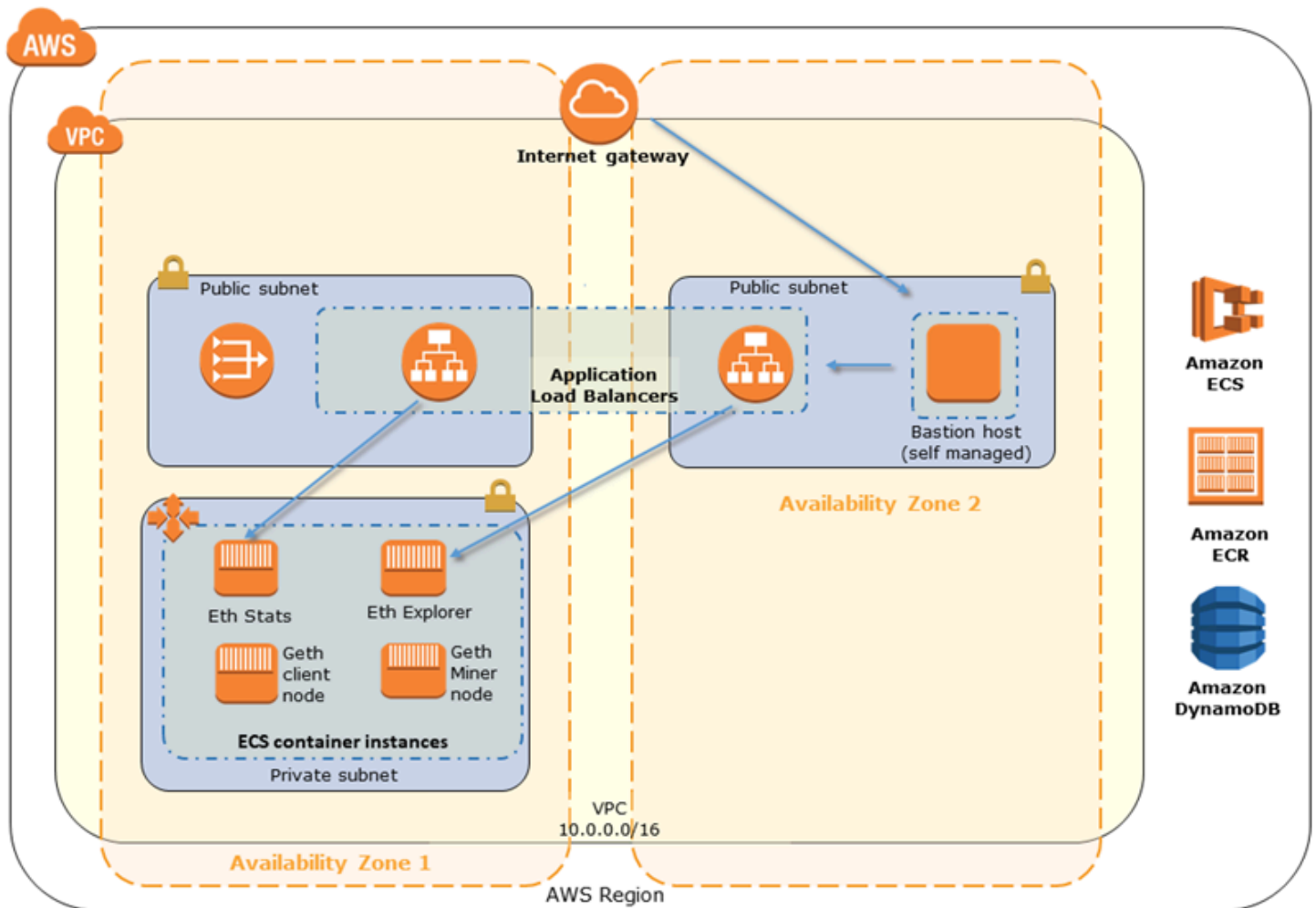
O AWS Blockchain Templates usa contêineres do Docker armazenados no Amazon ECR para implementar o software de blockchain. O AWS Blockchain Template para o Ethereum oferece duas opções para a Plataforma Container :

- `ecs` — Especifica que o Ethereum é executado em um cluster do Amazon ECS de instâncias do Amazon EC2.
- `docker-local` — Especifica que o Ethereum é executado em uma única instância do EC2.

Usando a Plataforma de Contêineres do Amazon ECS

Com o Amazon ECS, você cria a sua rede Ethereum em um cluster do ECS composto de várias instâncias do EC2, com um Application Load Balancer e recursos relacionados. Para mais informações sobre como usar a configuração do Amazon ECS, consulte o tutorial [Conceitos básicos do AWS Blockchain Templates](#).

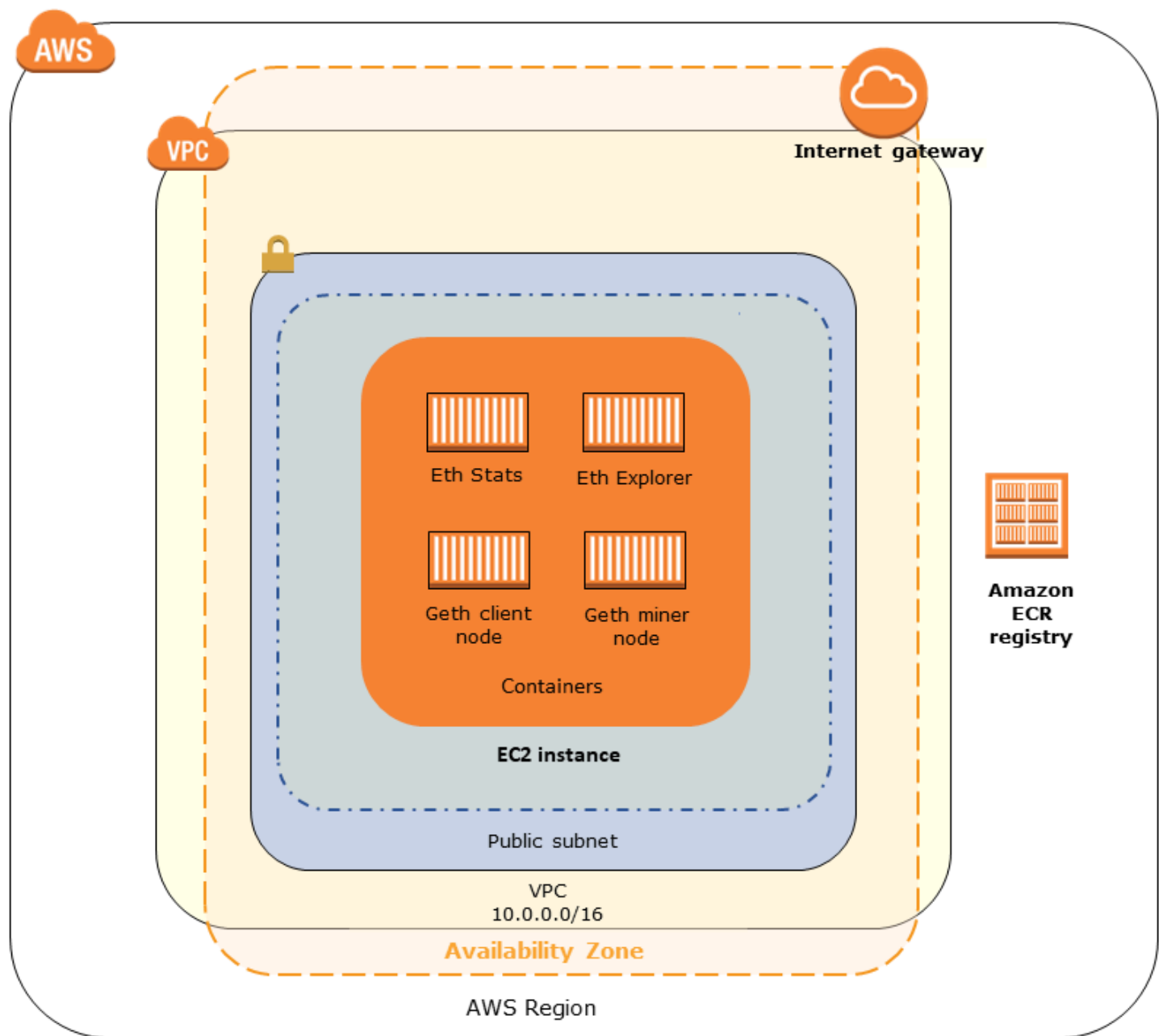
O diagrama a seguir descreve uma rede Ethereum criada em usando um modelo com a opção de plataforma de contêiner ECS:



Usando a plataforma Docker-Local

Se preferir, você pode lançar contêineres Ethereum em uma única instância do Amazon EC2. Todos os contêineres são executados em uma única instância do EC2. Esta é uma configuração simplificada.

O diagrama a seguir descreve uma rede Ethereum criada usando o template com a opção de plataforma de contêiner docker-local:



Escolher uma rede Ethereum Privada ou Pública

Escolher um valor para a Ethereum Network ID diferente de 1 a 4 cria nós privados no Ethereum que são executados dentro de uma rede que você define, usando os parâmetros de rede privada que você especifica.

Quando você escolhe um valor para a Ethereum Network ID de 1 a 4, os nós do Ethereum que você cria são unidos à rede pública do Ethereum. Você pode ignorar as configurações da rede privada e

seus respectivos padrões. Se você optar por unir os nós de Ethereum à rede pública de Ethereum, verifique se os serviços apropriados na sua rede podem ser acessados pela Internet.

Alterar as contas padrão e a frase mnemônica

Uma frase mnemônica é um conjunto aleatório de palavras que podem ser usadas para gerar carteiras de Ethereum (ou seja, pares de chaves privadas/públicas) para contas associadas em qualquer rede. A frase mnemônica pode ser usada para acessar Ether para contas associadas. Criamos um mnemônico padrão associado às contas padrão que o modelo de Ethereum usa.

Warning

Use as contas padrão e a frase mnemônica associada somente para fins de teste. Não envie Ether real usando o conjunto padrão de contas, pois qualquer pessoa com acesso à frase mnemônica pode acessar ou roubar o Ether das contas. Em vez disso, especifique contas personalizadas para fins de produção. A frase mnemônica associada à conta padrão é `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

Pré-requisitos

Quando você configura a sua rede Ethereum usando o AWS Blockchain Template para o Ethereum, os requisitos mínimos listados abaixo precisam ser atendidos. O modelo exige os componentes da AWS listados para cada uma das seguintes categorias:

Tópicos

- [Pré-requisitos para acessar recursos do Ethereum](#)
- [Pré-requisitos IAM](#)
- [Pré-requisitos do grupo de segurança](#)
- [Pré-requisitos do VPC](#)
- [Exemplo de permissões do IAM para o perfil de instância do EC2 e a regra do ECS](#)

Pré-requisitos para acessar recursos do Ethereum

Pré-requisito	Para plataforma do ECS	Para Docker-Local
Um par de chaves Amazon EC2 que você pode usar para acessar instâncias do EC2. A chave precisa existir na mesma Região que o cluster do ECS e outros recursos.	✓	✓
Um componente voltado para a Internet, como um bastion host ou um balanceador de carga, com um endereço interno pelo qual o tráfego é permitido no Application Load Balancer. Isso é necessário com a plataforma do ECS porque o modelo cria um balanceador de carga interno por motivos de segurança. Isso é necessário com a plataforma docker-local quando a instância do EC2 está em uma sub-rede privada, o que recomendamos. Para obter informações sobre como configurar um bastion host, consulte Criar um bastion host .	✓	✓ (com sub-rede privada)

Pré-requisitos IAM

Pré-requisito	Para plataforma do ECS	Para Docker-Local
Uma entidade principal do IAM; (usuário ou grupo) com permissões para trabalhar com todos os serviços relacionados.	✓	✓
Um perfil de instância do Amazon EC2 com permissões apropriadas para instâncias do EC2 para interagir com outros serviços. Para obter mais informações, consulte To create an EC2 instance profile.	✓	✓
Uma função do perfil do IAM, com permissões para que o Amazon ECS interaja com outros serviços. Para ter mais informações, consulte Criar a função e as permissões do ECS.	✓	

Pré-requisitos do grupo de segurança

Pré-requisito	Para plataforma do ECS	Para Docker-Local
Um security group para instâncias do EC2, com os seguintes requisitos:	✓	✓

Pré-requisito	Para plataforma do ECS	Para Docker-Local
<ul style="list-style-type: none">Regras de saída que permitem tráfego para 0.0.0.0/0 (padrão).	✓	✓
<ul style="list-style-type: none">Uma regra de entrada que permite todo o tráfego de si (o mesmo grupo de segurança).	✓	✓
<ul style="list-style-type: none">Uma regra de entrada que permite todo o tráfego do grupo de segurança para o Application Load Balancer.	✓	
<ul style="list-style-type: none">Regras de entrada que permitem HTTP (porta 80) EthStats (servido na porta 8080), JSON RPC sobre HTTP (porta 8545) e SSH (porta 22) de fontes externas confiáveis, como o CIDR IP do seu computador cliente.		✓

Pré-requisito	Para plataforma do ECS	Para Docker-Local
<p>Um grupo de segurança para o Application Load Balancer, com os seguintes requisitos:</p> <ul style="list-style-type: none">• Uma regra de entrada que permite todo o tráfego de si (o mesmo grupo de segurança).• Uma regra de entrada que permite o tráfego do grupo de segurança para instâncias do EC2.• As regras de saída que permitem todo o tráfego somente ao grupo de segurança para instâncias do EC2. Para ter mais informações, consulte Criar security groups.• Se associar esse mesmo grupo de segurança a um bastion host, uma regra de entrada que permite o tráfego SSH (porta 22) de origens confiáveis.• Se o bastion host ou outro componente voltado para a Internet estiver em um grupo de segurança diferente, uma regra de entrada que permite o tráfego desse componente.	✓	

Pré-requisitos do VPC

Pré-requisito	Para plataforma do ECS	Para Docker-Local
Um endereço IP Elástico que é usado para acessar os serviços do Ethereum.	✓	✓
Uma sub-rede para executar instâncias do EC2. Recomendamos muito uma sub-rede privada.	✓	✓
Duas sub-redes acessíveis ao público geral. Cada sub-rede deve estar em zonas de disponibilidade diferentes uma da outra, com uma na mesma zona de disponibilidade que a sub-rede para instâncias do EC2.	✓	

Exemplo de permissões do IAM para o perfil de instância do EC2 e a regra do ECS

Você especifica um ARN do perfil de instância do EC2 como um dos parâmetros ao usar o modelo. Se você usar a plataforma de contêiner do ECS, também especificará um ARN de função do ECS. As políticas de permissões anexadas a essas funções permitem que os recursos e as instâncias da AWS no cluster interajam com os outros recursos da AWS. Para obter mais informações, consulte [Perfis do IAM](#) no Guia do usuário do IAM. Use as instruções e os procedimentos de política abaixo como ponto de partida para criar permissões.

Exemplo de política de permissões para o perfil de instância do EC2

A política de permissões a seguir demonstra as ações permitidas para o perfil de instância do EC2 quando você escolhe a plataforma do contêiner ECS. As mesmas instruções de política podem ser usadas em uma plataforma de contêiner local do Docker, com chaves de contexto ecs removidas para limitar o acesso.

```
{
  "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": "*"
    }
  ]
}
```

Criar a função e as permissões do ECS

Para as permissões associadas à função do ECS, recomendamos que você comece com a política de permissões do AmazonEC2 ContainerServiceRole. Use o procedimento a seguir para criar uma função e anexar esta política de permissões. Use o console do IAM para ver a maioria das up-to-date permissões nessa política.

Para criar os perfis do IAM do Amazon ECS

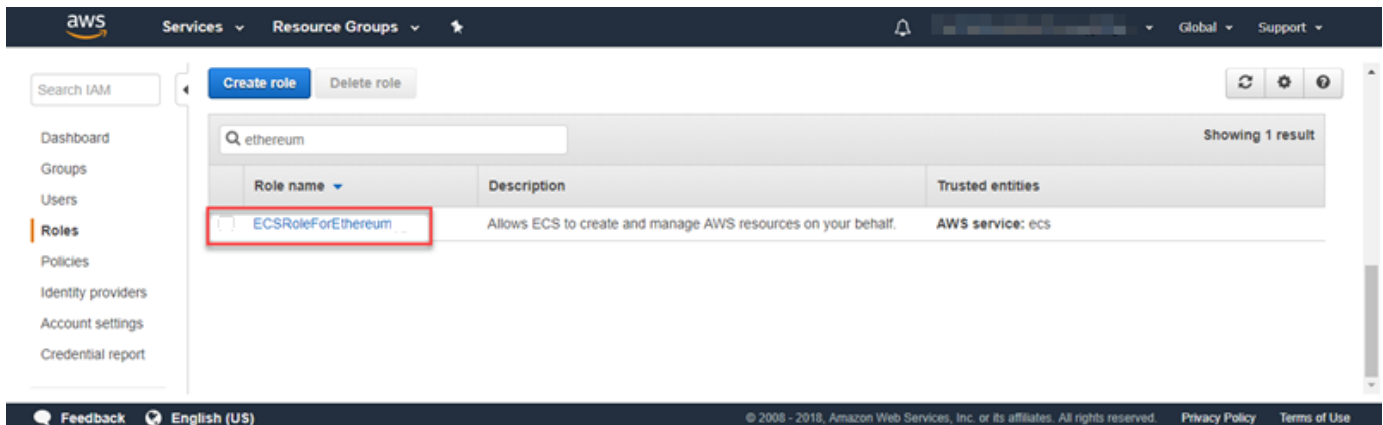
1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Roles (Funções) e Create Role (Criar função).
3. Em Select type of trusted entity (Selecionar tipo de entidade confiável), escolha AWS service (Produto da AWS).
4. Em Choose the service that will use this role (Escolher o serviço que usará esta função), escolha Elastic Container Service.
5. Em Select your use case (Selecionar seu caso de uso), escolha Elastic Container Service, Next:Permissions (Próximo: permissões).

The screenshot shows the AWS IAM console 'Create role' wizard. The 'Select type of trusted entity' step has 'AWS service' selected. The 'Choose the service that will use this role' step shows a grid of services, with 'Elastic Container Service' highlighted. The 'Select your use case' step shows 'Elastic Container Service' selected. The 'Next: Permissions' button is visible at the bottom right.

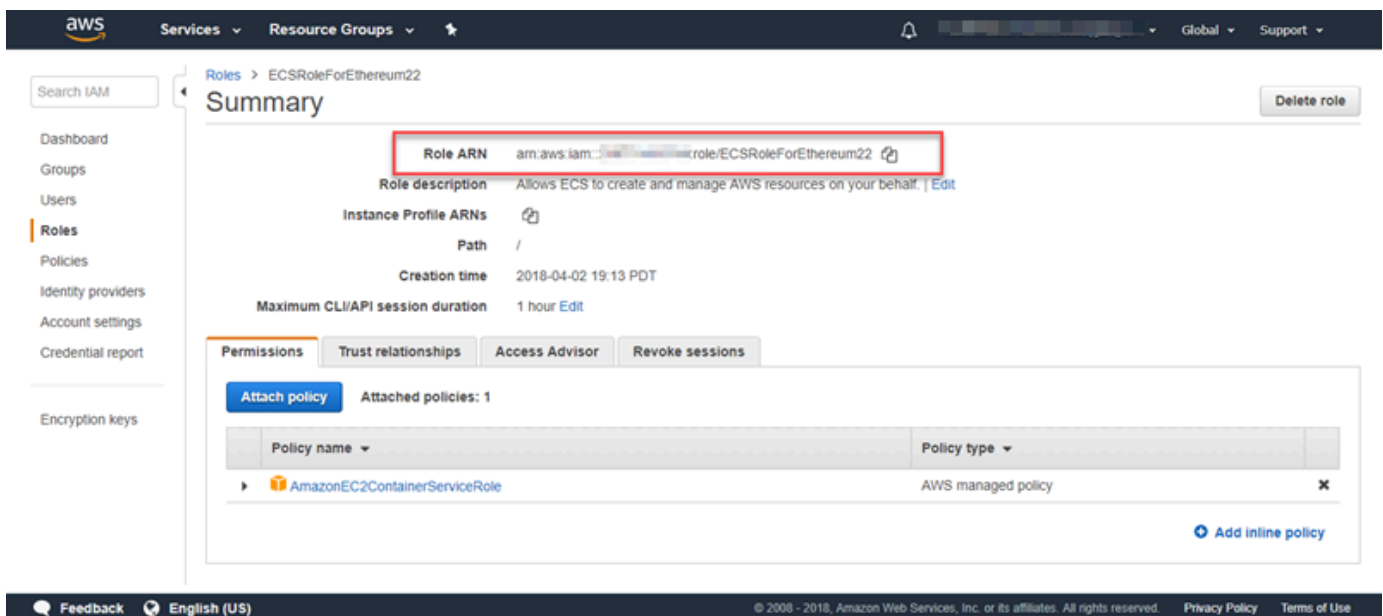
Choose the service that will use this role				
EC2 Allows EC2 instances to call AWS services on your behalf.				
Lambda Allows Lambda functions to call AWS services on your behalf.				
API Gateway	DMS	Elastic Transcoder	Machine Learning	SageMaker
Application Auto Scaling	Data Pipeline	ElasticLoadBalancing	MediaConvert	Service Catalog
Auto Scaling	DeepLens	Glue	OpsWorks	Step Functions
Batch	Directory Service	Greengrass	RDS	Storage Gateway
CloudFormation	DynamoDB	GuardDuty	Redshift	
CloudHSM	EC2	Inspector	Rekognition	
CloudWatch Events	EMR	IoT	S3	
CodeBuild	ElastiCache	Kinesis	SMS	
CodeDeploy	Elastic Beanstalk	Lambda	SNS	
Config	Elastic Container Service	Lex	SWF	

6. Em Política de permissões, deixe a política padrão (AmazonEC2 ContainerServiceRole) selecionada e escolha Next:Review.

7. Em Nome da função, insira um valor que o ajude a identificar a função, como ECS. RoleForEthereum Em Role Description (Descrição da função), digite um resumo breve. Anote o nome da função para mais tarde.
8. Selecione Criar perfil.
9. Na lista, selecione a função que você acabou de criar. Caso sua conta tenha várias funções, você pode pesquisar o nome da função.



10. Copie o valor de Role ARN (ARN da função) e salve-o para copiá-lo novamente depois. Você precisará dele quando for criar a rede de Ethereum.



Conectando-se aos recursos do Ethereum

Depois que a pilha raiz que você criou usando o modelo exibir `CREATE_COMPLETE`, você poderá conectar-se aos recursos do Ethereum usando o console do AWS CloudFormation. O modo como você se conecta depende da plataforma de contêiner escolhida, ECS ou docker-local:

- ECS — A aba Saída da pilha raiz fornece links para serviços em execução no Application Load Balancer. Essas URLs não estão diretamente acessíveis por motivos de segurança. Para se conectar, é possível configurar e usar um bastion host para conexões de proxy com eles. Para obter mais informações, consulte [Conexões de proxy usando um Bastion Host](#) abaixo.
- docker-local — Conecte-se usando o endereço IP da instância do EC2 que hospeda os serviços do Ethereum, conforme listados abaixo. Use o console do EC2 para encontrar o endereço *ec2-IP-address* da instância que o modelo criou.
 - EthStats—*Use o endereço IP http://EC2-IP*
 - EthExplorer— *Use http://endereço IP do EC2:8080*
 - EthJsonRpc—*Use o endereço IP http://EC2-IP:8545*

Se você especificou uma sub-rede pública para a ID da Sub-rede da Ethereum Network (Lista de sub-redes de VPC a serem usadas dentro do modelo), será possível conectar-se diretamente. Seu cliente deve ser uma fonte confiável de tráfego de entrada para SSH (porta 22), bem como as portas listadas. Isso é determinado pelo Grupo de Segurança EC2 que você especificar usando o Blockchain Template para o Ethereum.

Se você especificou uma sub-rede privada, é possível configurar e usar um bastion host para conexões de proxy com esses endereços. Para obter mais informações, consulte [Conexões de proxy usando um Bastion Host](#) abaixo.

Conexões de proxy usando um Bastion Host

Com algumas configurações, os serviços do Ethereum podem não estar disponíveis publicamente. Nesses casos, você pode se conectar aos recursos do Ethereum por meio de um bastion host. Para obter mais informações sobre bastion hosts, consulte [Arquitetura de Bastion Host do Linux](#) no Guia de início rápido de Bastion Hosts do Linux.

O bastion host é uma instância do EC2. Certifique-se de cumprir os seguintes requisitos:

- A instância EC2 do bastion host está em uma sub-rede pública com atribuição automática de IP público habilitada e inclui gateway da internet.

- O bastion host tem o par de chaves que permite conexões ssh.
- O bastion host está associado a um grupo de segurança que permite o tráfego SSH de entrada dos clientes que se conectam.
- O grupo de segurança atribuído aos hosts Ethereum (por exemplo, o Application Load Balancer, se o ECS for a plataforma de contêiner, ou a instância EC2 do host, se docker-local for a plataforma de contêiner) permite tráfego de entrada em todas as portas a partir de fontes dentro da VPC.

Com um bastion host configurado, certifique-se de que os clientes que se conectam usem o bastion host como proxy. O exemplo a seguir demonstra a configuração de uma conexão proxy usando o Mac OS. Substitua *bastionIP* pelo endereço IP da instância EC2 do bastion host e *MySshKey.pem* pelo arquivo de key pair que você copiou para o bastion host.

Na linha de comando, digite o seguinte:

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

Isto configura o encaminhamento de portas para a porta 9001 na máquina local para o bastion host.

Em seguida, configure seu navegador ou sistema para usar o proxy SOCKS para `localhost:9001`. Por exemplo, ao usar o Mac OS, selecione System Preferences (Preferências do sistema), Network (Rede), Advanced (Avançado), selecione SOCKS proxy (Proxy SOCKS) e digite `localhost:9001`.

Usando o FoxyProxy Padrão com o Chrome, selecione Mais ferramentas, extensões. Em FoxyProxy Padrão, selecione Detalhes, Opções de extensão e Adicionar novo proxy. Selecione Manual Proxy Configuration (Configuração manual de proxy). Em Host or IP Address (Host ou endereço IP), digite `localhost` e para Port (Porta), digite `9001`. Selecione SOCKS Proxy?, Save (Salvar).

Agora você deve poder conectar-se aos endereços de host dos endereços Ethereum listados na saída do modelo.

Usando o AWS Blockchain Template para o Hyperledger Fabric

O Hyperledger Fabric é uma estrutura de blockchain que executa contratos inteligentes chamados de chaincode, escritos em Go. Você pode criar uma rede privada com o Hyperledger Fabric, limitando os pares que podem se conectar e participar da rede. Para obter mais informações sobre o Hyperledger Fabric, consulte a documentação do [Hyperledger Fabric](#). Para obter mais

informações sobre chaincode, consulte o tópico [Chaincode para desenvolvedores](#) na documentação do [Hyperledger Fabric](#).

O AWS Blockchain Template para o Hyperledger Fabric só oferece suporte a uma plataforma de contêiner docker-local, ou seja, os contêineres do Hyperledger Fabric são implantados em uma única instância do EC2.

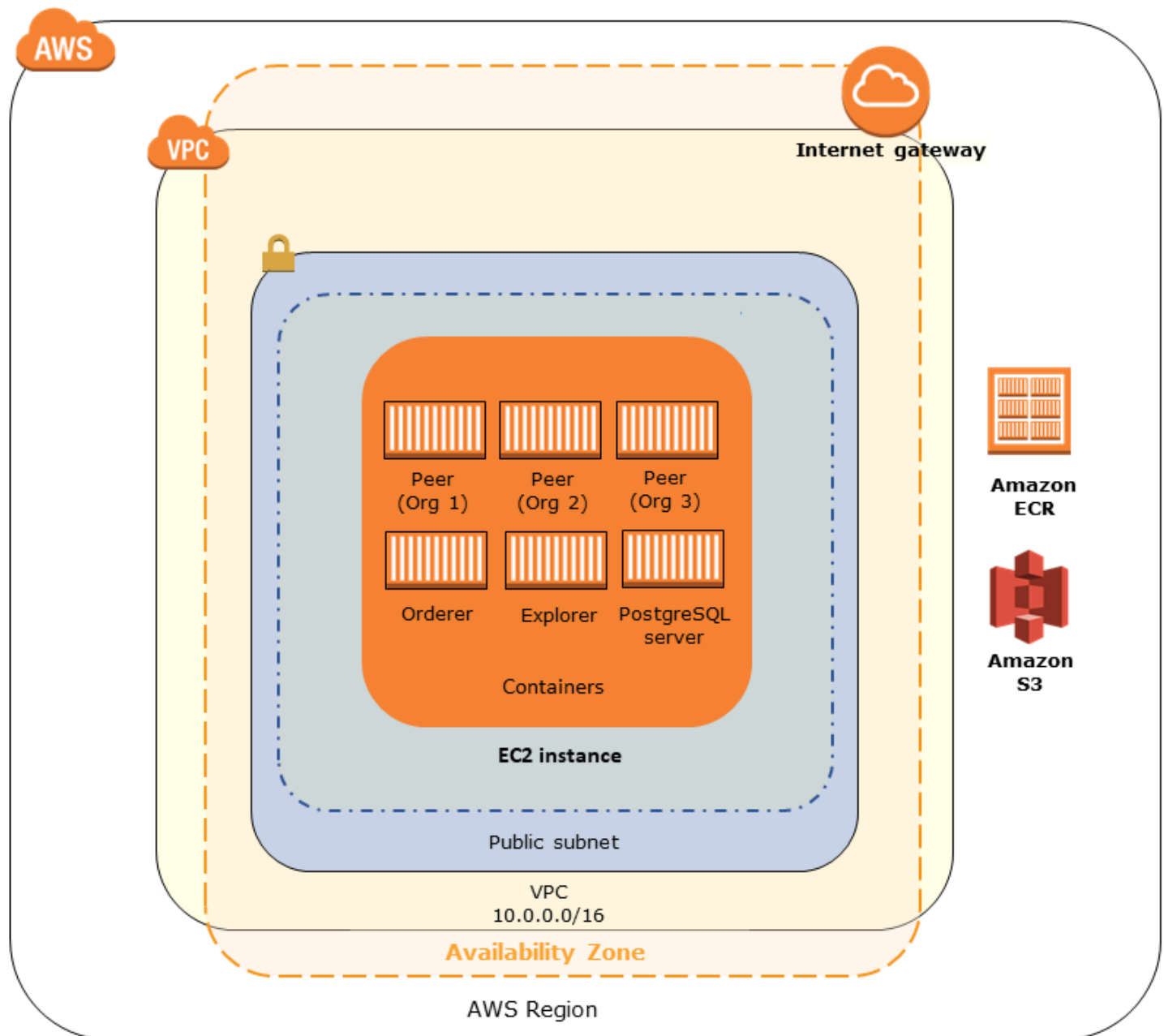
Links para execução

Consulte os [Conceitos básicos do AWS Blockchain Templates](#) para obter links para iniciar o AWS CloudFormation em regiões específicas usando os modelos do Hyperledger Fabric.

AWS Blockchain Template para componentes do Hyperledger Fabric

O AWS Blockchain Template para o Hyperledger Fabric cria uma instância do EC2 com o Docker e executa uma rede da Hyperledger Fabric usando contêineres nessa instância. A rede inclui um serviço de pedidos e três organizações, cada uma com um serviço de um colega. O modelo também inicia um contêiner do Hyperledger Explorer que permite que você procure dados de blockchain. Um contêiner de servidor PostgreSQL é executado para oferecer suporte ao Hyperledger Explorer.

O diagrama a seguir descreve uma rede do Hyperledger Fabric criada usando o modelo:



Pré-requisitos

Antes de lançar uma rede do Hyperledger Fabric usando o modelo, verifique se os seguintes requisitos foram atendidos:

- O princípio de IAM; (usuário ou grupo) que você utiliza deve ter permissão para trabalhar com todos os serviços relacionados.
- Você precisa ter acesso a um par de chaves que pode ser usado para acessar instâncias do EC2 (por exemplo, usando SSH). A chave precisa existir na mesma região que a instância.

- É necessário ter um perfil de instância do EC2 com uma política de permissões anexada que permita acesso ao Amazon S3 e ao Amazon Elastic Container Registry (Amazon ECR) para a extração de contêineres. Para um exemplo de política de permissões, consulte [Exemplo de permissões IAM para o perfil de instância do EC2](#).
- É necessário ter uma rede Amazon VPC com uma sub-rede pública, ou uma sub-rede privada com um Gateway NAT e um endereço IP elástico para que o Amazon S3, AWS CloudFormation, and Amazon ECR possam ser acessados.
- É necessário ter um grupo de segurança EC2 com regras de entrada que permitam o tráfego SSH (porta 22) dos endereços IP que precisam se conectar à instância usando SSH, e o mesmo para os clientes que precisam se conectar ao Hyperledger Explorer (porta 8080).

Exemplo de permissões IAM para o perfil de instância do EC2

Você especifica um ARN do perfil de instância do EC2 como um dos parâmetros ao usar o AWS Blockchain Template para o Hyperledger Fabric.. Use a seguinte declaração de política como ponto de partida para a política de permissões anexada ao perfil de instância e à função do EC2.

```
{
  "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": "*"
    }
  ]
}
```

Conectando-se aos recursos do Hyperledger Fabric

Quando a pilha raiz que você criou com o modelo exibir `CREATE_COMPLETE`, será possível se conectar aos recursos do Hyperledger Fabric na instância do EC2. Se você especificou uma sub-rede pública, você poderá se conectar à instância do EC2 da mesma maneira como faria com qualquer outra instância do EC2. Para obter mais informações, consulte [Conectando-se à sua Instância do Linux usando SSH](#) no Guia do usuário do Amazon EC2 para instâncias do Linux.

Se você tiver especificado uma sub-rede privada, será possível configurar e usar um bastion host em conexões de proxy para recursos do Hyperledger Fabric. Para obter mais informações, consulte [Conexões de proxy usando um Bastion Host](#) abaixo.

Note

Você pode notar que o modelo aloca um endereço IP público para a instância do EC2 que hospeda os serviços do Hyperledger Fabric. No entanto, esse endereço IP não está acessível publicamente porque as políticas de roteamento na sub-rede privada que você especifica não permitem o tráfego entre esse endereço IP e as fontes públicas.

Conexões de proxy usando um Bastion Host

Com algumas configurações, os serviços do Hyperledger Fabric podem não estar disponíveis publicamente. Nesses casos, você pode se conectar aos recursos do Hyperledger Fabric por meio de um bastion host. Para obter mais informações sobre bastion hosts, consulte [Arquitetura de Bastion Host do Linux](#) no Guia de início rápido de Bastion Hosts do Linux.

O bastion host é uma instância do EC2. Certifique-se de cumprir os seguintes requisitos:

- A instância EC2 do bastion host está em uma sub-rede pública com atribuição automática de IP público habilitada e inclui gateway da internet.
- O bastion host tem o par de chaves que permite conexões ssh.
- O bastion host está associado a um grupo de segurança que permite o tráfego SSH de entrada dos clientes que se conectam.
- O grupo de segurança atribuído aos hosts do Hyperledger Fabric (por exemplo, o Application Load Balancer, se o ECS for a plataforma de contêiner, ou a instância EC2 do host, se o docker-local for a plataforma de contêiner) permite tráfego de entrada em todas as portas a partir de fontes dentro da VPC.

Com um bastion host configurado, certifique-se de que os clientes que se conectam usem o bastion host como proxy. O exemplo a seguir demonstra a configuração de uma conexão proxy usando o Mac OS. Substitua *bastionIP* pelo endereço IP da instância EC2 do bastion host e *MySshKey.pem* pelo arquivo de key pair que você copiou para o bastion host.

Na linha de comando, digite o seguinte:

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

Isto configura o encaminhamento de portas para a porta 9001 na máquina local para o bastion host.

Em seguida, configure seu navegador ou sistema para usar o proxy SOCKS para `localhost:9001`. Por exemplo, ao usar o Mac OS, selecione System Preferences (Preferências do sistema), Network (Rede), Advanced (Avançado), selecione SOCKS proxy (Proxy SOCKS) e digite `localhost:9001`.

Usando o FoxyProxy Padrão com o Chrome, selecione Mais ferramentas, extensões. Em FoxyProxy Padrão, selecione Detalhes, Opções de extensão e Adicionar novo proxy. Selecione Manual Proxy Configuration (Configuração manual de proxy). Em Host or IP Address (Host ou endereço IP), digite `localhost` e para Port (Porta), digite `9001`. Selecione SOCKS Proxy?, Save (Salvar).

Agora você deve poder conectar-se aos endereços de host do Hyperledger Fabric listados na saída do modelo.

Histórico do documento

A tabela a seguir descreve as alterações na documentação deste guia.

Última atualização da documentação: 1º de maio de 2019

Alteração	Descrição	Data
Descontinuação do AWS Blockchain Templates.	O AWS Blockchain Templates foi descontinuado em 30 de abril de 2019. Nenhuma atualização adicional será feita para este serviço ou nesta documentação de suporte. Para obter a melhor experiência com Managed Blockchain no AWS, recomendamos que você use o Amazon Managed Blockchain (AMB) .	1º de maio de 2019
Atualizações do bastion host.	Modificação do tutorial de conceitos básicos e requisitos de pré-requisitos para a adição de um bastion host, que permite o acesso a recursos da Web atendidos por meio do load balancer interno ao usar a plataforma do ECS e a instância do EC2 ao usar docker-local.	3 de maio de 2018
Guia criado.	Novo guia do desenvolvedor para dar suporte ao lançamento inicial do AWS Blockchain Templates.	19 de abril de 2018

AWS Glossário

Para obter a terminologia mais recente da AWS, consulte o [glossário da AWS](#) na Referência do Glossário da AWS.