



Guía para desarrolladores

AWS Blockchain Templates



AWS Blockchain Templates: Guía para desarrolladores

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

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

.....	iv
¿Qué es AWS Blockchain Templates?	1
Cómo empezar	2
Conozco bien la cadena de bloques AWS	2
Soy experto en la cadena de bloques y soy nuevo en ella AWS	3
Soy un principiante AWS y domino la cadena de bloques	3
Soy nuevo en el mundo de la cadena AWS de bloques	3
Servicios relacionados	3
Configuración	5
Inscripción en AWS	5
Creación de un usuario de IAM	6
Creación de un par de claves	8
Introducción	10
Configuración de requisitos previos	11
Creación de una VPC y de subredes	11
Creación de los grupos de seguridad	15
Creación de un rol de IAM; para Amazon ECS; y un perfil de instancia EC2	18
Creación de un host bastión	24
Creación de la red Ethereum	25
Conectarse EthStats a Bastion Host y EthExplorer usarlo	28
Eliminación de recursos de	31
AWS Blockchain Templates y sus características	33
AWS Blockchain Template para Ethereum	33
Enlaces para el lanzamiento	33
Opciones de Ethereum	34
Requisitos previos	37
Conexión a Ethereum Resources	46
AWS Blockchain Template para Hyperledger Fabric	47
Enlaces para el lanzamiento	48
AWS Blockchain Template para componentes de Hyperledger Fabric	48
Requisitos previos	49
Conexión a los recursos de Hyperledger Fabric	51
Historial del documento	53
Glosario de AWS	54

AWSBlockchain Templates se suspendió el 30 de abril de 2019. No habrá más actualizaciones de este servicio ni de esta documentación complementaria. Para disfrutar de la mejor experiencia de cadena de bloques gestionada en AWS, le recomendamos que utilice [Amazon Managed Blockchain \(AMB\)](#). Para obtener más información sobre cómo empezar a utilizar Amazon Managed Blockchain, consulte nuestro [taller sobre Hyperledger Fabric](#) o nuestro [blog sobre la implementación de un nodo de Ethereum](#). Si tiene preguntas AMB o necesita más ayuda, [póngase en contacto con AWS Support](#)o tu AWS equipo de cuentas.

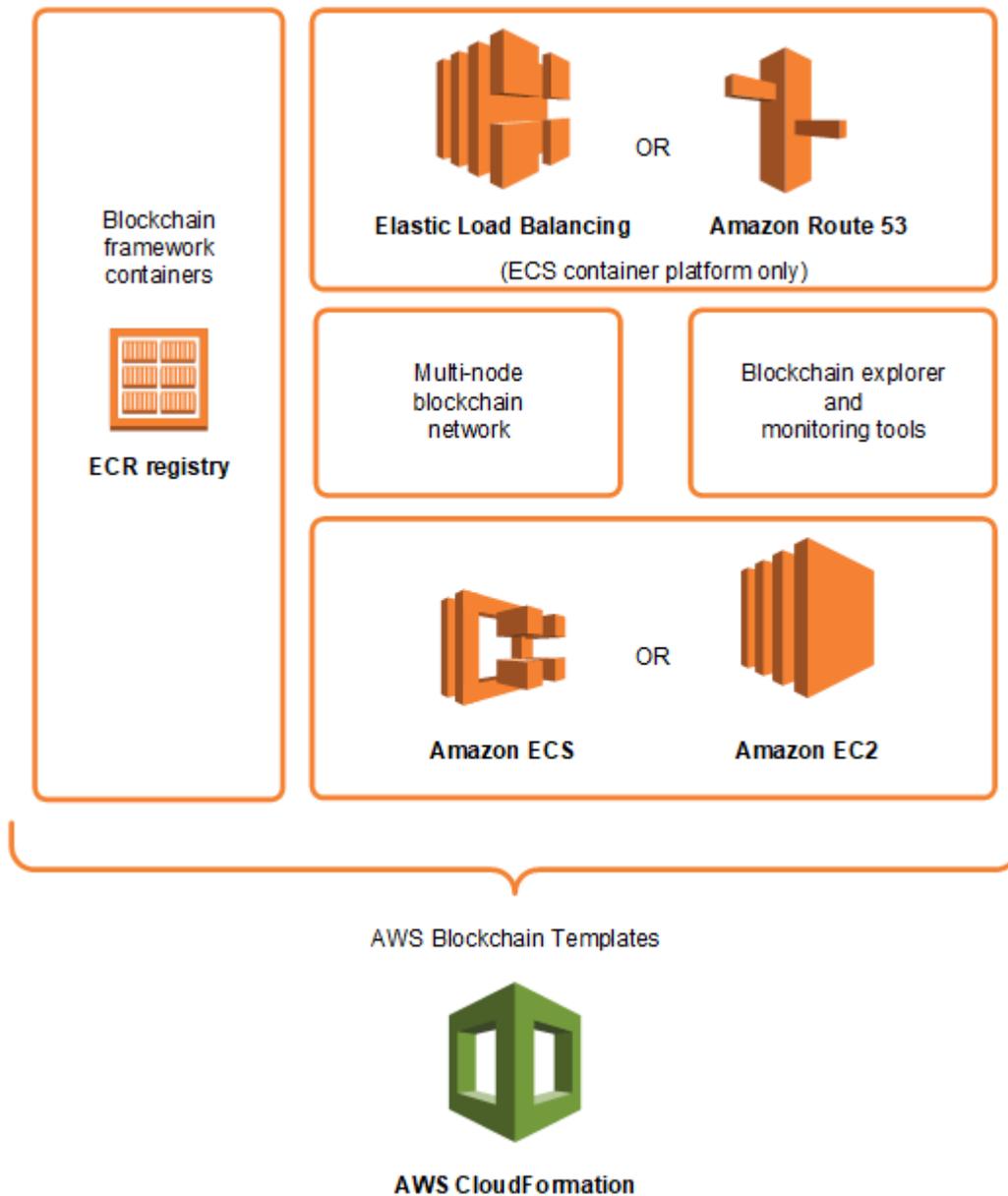
Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.

¿Qué es AWS Blockchain Templates?

AWS Blockchain Templates le ayuda a crear e implementar redes de cadenas de bloques con rapidez AWS utilizando diferentes marcos de cadenas de bloques. La cadena de bloques (o blockchain) es una tecnología de base de datos descentralizada que mantiene un conjunto de transacciones y contratos inteligentes que crece continuamente y que están protegidos contra su revisión y manipulación mediante criptografía.

Una red de cadenas de bloques es una peer-to-peer red que mejora la eficiencia y la inmutabilidad de las transacciones para procesos empresariales como los pagos internacionales, la gestión de la cadena de suministro, el registro de tierras, la financiación colectiva, la gobernanza, las transacciones financieras, etc. Esto permite que las personas y organizaciones que quizás no se conozcan entre sí confíen y verifiquen de forma independiente el registro de transacciones.

Utiliza AWS Blockchain Templates para configurar y lanzar AWS CloudFormation pilas para crear redes de cadenas de bloques. Los recursos y servicios de AWS que utilice dependen de la AWS Blockchain Templates que elija y de las opciones que especifique. Para obtener información sobre las plantillas disponibles y sus características, consulte [AWS Blockchain Templates y sus características](#). Los componentes fundamentales de una red de cadenas de bloques AWS creada con AWS Blockchain Templates se muestran en el siguiente diagrama.



Cómo empezar

El mejor punto de partida depende de su nivel de experiencia con la cadena de bloques y AWS, en particular, con los servicios relacionados con AWS Blockchain Templates.

Conozco bien la cadena de bloques AWS

Comience por el tema [AWS Blockchain Templates y sus características](#) relacionado con el marco de trabajo que desea utilizar. Utilice los enlaces para lanzar la AWS Blockchain Templates y configurar la red de cadena de bloques o para descargar las plantillas y examinarlas por su cuenta.

Soy experto en la cadena de bloques y soy nuevo en ella AWS

Comience por el tutorial [Introducción a AWS Blockchain Templates](#). Este le guiará en el proceso de creación de una red de cadena de bloques Ethereum introductoria con la configuración predeterminada. Cuando haya terminado, consulte [AWS Blockchain Templates y sus características](#) para obtener información general sobre los marcos de trabajo de cadena de bloques, así como enlaces para obtener más información sobre las características y opciones de configuración.

Soy un principiante AWS y domino la cadena de bloques

Comience por [Configuración de Plantillas AWS de Blockchain](#). Esto te ayuda a familiarizarte con los aspectos básicos AWS, como una cuenta y un perfil de usuario. A continuación, realice el tutorial [Introducción a AWS Blockchain Templates](#). Este tutorial le guiará en el proceso de creación de una red de cadena de bloques Ethereum introductoria con la configuración predeterminada. Incluso si al final no utiliza Ethereum, obtendrá experiencia práctica en la configuración de los servicios relacionados. Esta experiencia es útil para todos los marcos de trabajo de cadena de bloques. Por último, consulte el tema relacionado con su marco de trabajo en la sección [AWS Blockchain Templates y sus características](#).

Soy nuevo en el mundo de la cadena AWS de bloques

Comience por [Configuración de Plantillas AWS de Blockchain](#). Esto te ayuda a familiarizarte con los aspectos básicos AWS, como una cuenta y un perfil de usuario. A continuación, realice el tutorial [Introducción a AWS Blockchain Templates](#). Este tutorial le guiará en el proceso de creación de una red de cadena de bloques Ethereum introductoria con la configuración predeterminada. Tómate tu tiempo para explorar los enlaces y obtener más información sobre AWS los servicios y Ethereum.

Servicios relacionados

Según las opciones que seleccione, AWS Blockchain Templates puede utilizar los siguientes AWS servicios para implementar la cadena de bloques:

- Amazon EC2: proporciona la capacidad de cómputo para la red de cadena de bloques. Para obtener más información, consulte la Guía del [usuario de Amazon EC2](#).
- Amazon ECS: organiza la implementación de contenedores entre las instancias EC2 de un clúster para la red de cadena de bloques, si decide utilizarlo. Para obtener más información, consulte [Amazon Elastic Container Service Developer Guide](#) (Guía para desarrolladores de Amazon Elastic Container Service).

- Amazon VPC: proporciona acceso de red para los recursos de Ethereum que se creen. Es posible personalizar la configuración para mejorar la accesibilidad y la seguridad. Para obtener más información, consulte la [Guía del desarrollador de Amazon VPC](#).
- Equilibrador de carga de aplicaciones: sirve como un único punto de contacto para el acceso a las interfaces de usuario disponibles y a la detección de servicios internos cuando se utiliza Amazon ECS como plataforma de contenedores. Para obtener más información, consulte [What is an Application Load Balancer?](#) (¿Qué es un equilibrador de carga de aplicaciones?) en la Guía del usuario de los equilibradores de carga de aplicaciones.

Configuración de Plantillas AWS de Blockchain

Antes de empezar con AWS Blockchain Templates, complete las siguientes tareas:

- [Inscripción en AWS](#)
- [Creación de un usuario de IAM](#)
- [Creación de un par de claves](#)

Estos son los requisitos previos fundamentales para todas las configuraciones de cadena de bloques. Además, la red de cadena de bloques que elija puede tener sus propios requisitos previos, que pueden variar de acuerdo con las opciones de configuración y el entorno elegido. Para obtener más información, consulte la sección correspondiente a su plantilla de cadena de bloques en [AWS Blockchain Templates y sus características](#).

Para step-by-step obtener instrucciones sobre cómo configurar los requisitos previos para una red Ethereum privada mediante un ECS clúster de Amazon, consulte [Introducción a AWS Blockchain Templates](#).

Inscripción en AWS

Cuando te registras en AWS, tu AWS la cuenta se registra automáticamente en todos los servicios. Solo se le cobrará por los servicios que utilice.

Si tienes un AWS ya tienes una cuenta, pasa a la siguiente tarea. Si no tienes una AWS utilice el siguiente procedimiento para crear una.

Para crear una AWS cuenta

1. Abre el <https://portal.aws.amazon.com/billing/registro>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, un Usuario raíz de la cuenta de AWS se crea. El usuario root tiene acceso a todos Servicios de AWS y los recursos de la cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

Tenga en cuenta sus AWS número de cuenta. Lo necesitará cuando cree un IAM usuario en la siguiente tarea.

Creación de un usuario de IAM

Servicios en AWS requieren que proporcione credenciales al acceder a ellos, de modo que el servicio pueda determinar si tiene permisos para acceder a sus recursos. La consola requiere que especifique la contraseña. Puede crear claves de acceso para su AWS cuenta para acceder a la interfaz de línea de comandos oAPI. Sin embargo, no le recomendamos que acceda AWS utilizando las credenciales de su AWS cuenta; le recomendamos que utilice AWS Identity and Access Management (IAM) en su lugar. Cree un IAM usuario y, a continuación, agréguelo a un IAM grupo con permisos administrativos u otorgue permisos administrativos a este usuario. A continuación, podrá acceder AWS utilizando un especial URL y las credenciales del IAM usuario.

Si te registraste en AWS pero no has creado un IAM usuario para ti, puedes crear uno desde la IAM consola. Si ya tiene un IAM usuario, puede omitir este paso.

Para crear un usuario administrador, elija una de las siguientes opciones.

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
En IAM Identity Center (recomendado)	Utilice credenciales de corta duración para acceder AWS. Esto se ajusta a las prácticas recomendadas de seguridad . Para obtener información sobre las	Siga las instrucciones de Primeros pasos en la AWS IAM Identity Center Guía del usuario.	Configure el acceso programático mediante la configuración del AWS CLI para usar AWS IAM Identity Center en la AWS Command Line Interface Guía del usuario.

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
	prácticas recomendadas, consulte las prácticas recomendadas de seguridad IAM en la Guía del IAM usuario.		
En IAM (No recomendado)	Utilice credenciales de larga duración para acceder AWS.	Siga las instrucciones de Cómo crear su primer usuario IAM administrador y grupo de usuarios en la Guía del IAM usuario.	Configure el acceso mediante programación mediante la administración de las claves de acceso de IAM los usuarios en la Guía del IAM usuario.

Para iniciar sesión como este nuevo IAM usuario, cierre sesión en AWS Management Console, luego usa lo siguiente URL, donde `your_aws_account_id` es tu AWS número de cuenta sin guiones (por ejemplo, si su AWS el número de cuenta es 1234-5678-9012, tu AWS el ID de cuenta es 123456789012):

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

Introduzca el nombre IAM de usuario y la contraseña que acaba de crear. Cuando haya iniciado sesión, en la barra de navegación se mostrará “`su_nombre_de_usuario @ su_id_de_cuenta_de_aws`”.

Si no quieres que la página URL de inicio de sesión contenga tus AWS ID de cuenta, puedes crear un alias de cuenta. En el IAM panel de control, selecciona Crear alias de cuenta e introduce un alias, como el nombre de tu empresa. Para iniciar sesión después de crear un alias de cuenta, usa lo siguiente URL:

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

Para verificar el enlace de inicio de sesión de IAM los usuarios de tu cuenta, abre la IAM consola y consulta el enlace de inicio de sesión de IAM los usuarios en el panel de control.

Para obtener más información, consulte la [AWS Guía del usuario de Identity and Access Management](#).

Creación de un par de claves

AWS utiliza criptografía de clave pública para proteger la información de inicio de sesión de las instancias de una red de cadenas de bloques. Usted especifica el nombre del key pair cuando usa cada plantilla de cadena de AWS bloques. A continuación, puede usar el key pair para acceder a las instancias directamente, por ejemplo, para iniciar sesión utilizando SSH.

Si ya tiene un par de claves en la región adecuada, puede omitir este paso. Si aún no has creado un key pair, puedes crear uno con la EC2 consola de Amazon. Debe crear el par de claves en la misma región que utiliza para lanzar la red Ethereum. Para obtener más información, consulta [Regiones y zonas de disponibilidad](#) en la Guía del EC2 usuario de Amazon.

Crear un par de claves

1. Abre la EC2 consola de Amazon en <https://console.aws.amazon.com/ec2/>.
2. En la barra de navegación, seleccione una región para el par de claves. Puede seleccionar cualquier región disponible, independientemente de su ubicación, pero los pares de claves son específicos de cada región. Por ejemplo, si tiene previsto lanzar una instancia en la región Este de EE. UU. (Ohio), debe crear un par de claves para la instancia en la misma región.
3. En el panel de navegación, seleccione Key Pairs (Pares de claves), Create Key Pair (Crear par de claves).
4. En Key pair name (Nombre del par de claves), introduzca un nombre para el par de claves nuevo. Elige un nombre que te resulte fácil de recordar, como tu nombre de IAM usuario-key-pair, seguido del nombre de la región. Por ejemplo, me-key-pairuseast2. Seleccione Crear.
5. Su navegador descargará el archivo de clave privada automáticamente. El nombre de archivo base es el nombre que especificó como nombre del par de claves y la extensión del archivo es .pem. Guarde el archivo de clave privada en un lugar seguro.

 **Important**

Esta es la única oportunidad para guardar el archivo de clave privada. Proporcione el nombre del par de claves al lanzar la red Ethereum.

Para obtener más información, consulte [Amazon EC2 Key Pairs](#) en la Guía del EC2 usuario de Amazon. Para obtener más información sobre cómo conectarse a EC2 instancias mediante el par de claves, consulte [Connect to Your Linux Instance](#) en la Guía del EC2 usuario de Amazon.

Introducción a AWS Blockchain Templates

En este tutorial se muestra cómo utilizar la plantilla de cadena de bloques de AWS para Ethereum a fin de crear una red de cadena de bloques privada AWS a través de ella AWS CloudFormation. La red que se crea tiene dos clientes de Ethereum y un minero que se ejecutan en instancias Amazon EC2 de un clúster de Amazon ECS. Amazon ECS ejecuta de estos servicios en contenedores de Docker se extraen de Amazon ECR. Antes de comenzar este tutorial, es útil conocer las redes de cadenas de bloques y los AWS servicios involucrados, pero no es obligatorio.

En este tutorial se supone que ha configurado los requisitos previos generales indicados en [Configuración de Plantillas AWS de Blockchain](#). Además, debe configurar algunos AWS recursos, como una red de Amazon VPC y permisos específicos para las funciones de IAM, antes de usar la plantilla.

En el tutorial se explica cómo configurar dichos requisitos previos. Se han elegido determinadas opciones de configuración, pero no son preceptivas. Siempre que se cumplan los requisitos previos, puede elegir otras opciones de configuración que se adapten a las necesidades de su aplicación y entorno. Para obtener información sobre las características y los requisitos previos generales para cada plantilla, y para descargar plantillas o lanzarlas directamente en AWS CloudFormation, consulte [AWS Blockchain Templates y sus características](#).

En los ejemplos de este tutorial, se utiliza la región Oeste de EE. UU. (Oregón) (us-west-2), pero puede utilizar cualquier región que admita las AWS Blockchain Templates:

- Región Oeste de EE. UU. (Oregón) (us-west-2)
- Región Este de EE. UU. (Norte de Virginia) (us-east-1)
- Región Este de EE. UU. (Ohio) (us-east-2)

Note

Si se ejecuta una plantilla en una región distinta de las mencionadas arriba, se lanzarán recursos en la región Este de EE. UU. (Norte de Virginia) (us-east-1).

La AWS Blockchain Template para Ethereum que se configura mediante este tutorial crea los siguientes recursos:

- Instancias EC2 bajo demanda del tipo y el número que especifique. En este tutorial, se utiliza el tipo de instancia predeterminado t2.medium.
- Un equilibrador de carga de aplicaciones interno.

Tras el tutorial, se proporcionan pasos para limpiar los recursos creados.

Temas

- [Configuración de requisitos previos](#)
- [Creación de la red Ethereum](#)
- [Conectarse EthStats a Bastion Host y EthExplorer usarlo](#)
- [Eliminación de recursos de](#)

Configuración de requisitos previos

Para la configuración de la AWS Blockchain Template para Ethereum especificada en este tutorial, es necesario hacer lo siguiente:

- [Creación de una VPC y de subredes](#)
- [Creación de los grupos de seguridad](#)
- [Creación de un rol de IAM; para Amazon ECS; y un perfil de instancia EC2](#)
- [Creación de un host bastión](#)

Creación de una VPC y de subredes

AWS Blockchain Template para Ethereum lanza recursos en una red virtual que haya definido mediante Amazon Virtual Private Cloud (Amazon VPC). La configuración especificada en este tutorial crea un equilibrador de carga de aplicaciones que requiere dos subredes públicas en zonas de disponibilidad distintas. Además, se requiere una subred privada para las instancias de contenedor y la subred debe estar en la misma zona de disponibilidad que el equilibrador de carga de aplicaciones. En primer lugar, utilice el Asistente para VPC para crear una subred pública y una subred privada en la misma zona de disponibilidad. A continuación, cree una segunda subred pública dentro de esta VPC en una zona de disponibilidad distinta.

Para obtener más información, consulte [¿Qué es Amazon VPC?](#) en la Guía del usuario de Amazon VPC.

Utilice la consola de Amazon VPC (<https://console.aws.amazon.com/vpc/>) para crear la dirección IP elástica, la VPC y la subred, tal y como se describe a continuación.

Para crear una dirección IP elástica

1. Abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
2. Elija Elastic IPs (Direcciones IP elásticas), Allocate new address (Asignar nueva dirección), Allocate (Asignar).
3. Anote la dirección IP elástica que ha creado y elija Close (Cerrar).
4. En la lista de direcciones IP elásticas, busque el Allocation ID (ID de asignación) de la dirección IP elástica creada anteriormente. Lo utilizará al crear la VPC.

Para crear la VPC

1. En la barra de navegación, seleccione una región para la VPC. Cada VPC corresponde a una región específica, así que deberá seleccionar la misma región en la que creó el par de claves y en la que va a lanzar la pila de Ethereum. Para obtener más información, consulte [Creación de un par de claves](#).
2. En el panel de VPC, seleccione Start VPC Wizard.
3. En la página Step 1: Select a VPC Configuration (Paso 1: Seleccionar una configuración de la VPC), elija VPC with Public and Private Subnets (VPC con subredes públicas y privadas), Select (Seleccionar).
4. En la página Step 2: VPC with Public and Private Subnets (Paso 2: VPC con subredes públicas y privadas), deje los valores predeterminados para IPv4 CIDR block (Bloque de CIDR IPv4) e IPv6 CIDR block (Bloque de CIDR IPv6). En VPC name (Nombre de la VPC), escriba un nombre fácil de recordar.
5. En Public subnet's IPv4 CIDR (CIDR de IPv4 de la subred pública), deje el valor predeterminado. En Availability Zone (Zona de disponibilidad), elija una zona. En Public subnet name (Nombre de la subred pública), escriba un nombre fácil de recordar.

Debe especificar esta subred como una de las dos primeras subredes del equilibrador de carga de aplicaciones cuando utilice la plantilla.

Tenga en cuenta la zona de disponibilidad de esta subred porque selecciona la misma zona de disponibilidad para la subred privada y otra distinta para la otra subred pública.

6. En Private subnet's IPv4 CIDR (CIDR de IPv4 de la subred privada), deje el valor predeterminado. En Availability Zone (Zona de disponibilidad), seleccione la misma zona de disponibilidad que en el paso anterior. En Private subnet name (Nombre de la subred privada), escriba un nombre fácil de recordar.
7. En Elastic IP Allocation ID (ID de asignación de IP elástica), seleccione la dirección IP elástica que creó anteriormente.
8. Deje los valores predeterminados de las demás opciones.
9. Seleccione Crear VPC.

El siguiente ejemplo muestra una VPC EthereumNetworkVPC con una subred pública EthereumPubSub1 y una subred privada 1. EthereumPvtSub La subred pública utiliza la zona de disponibilidad 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

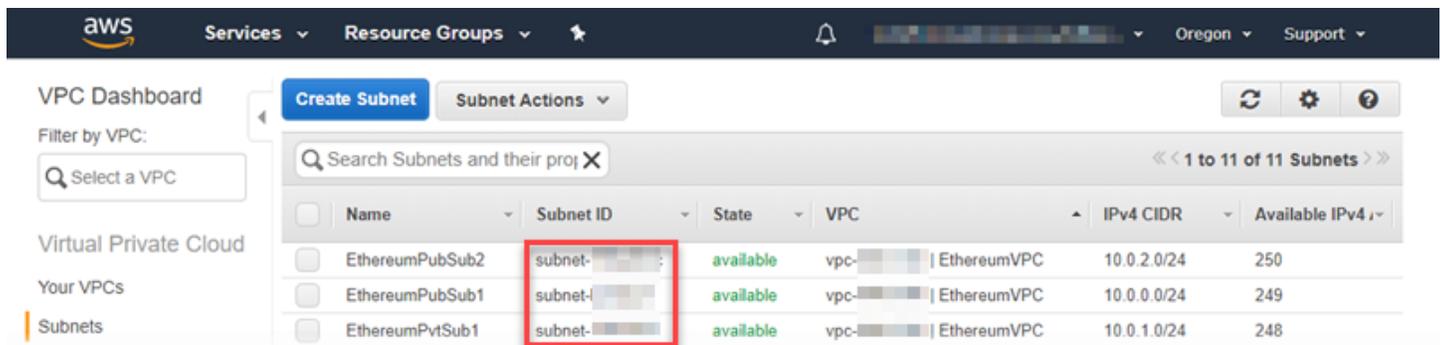
Hardware tenancy:* ▼

Para crear la segunda subred pública en otra zona de disponibilidad

1. Elija Subnets (Subredes) y, a continuación, seleccione de la lista la subred pública que creó anteriormente en la lista. Seleccione la pestaña Route Table (Tabla de enrutamiento) y anote el ID de la Route table (Tabla de enrutamiento). Especifique esta misma tabla de enrutamiento para la segunda subred pública a continuación.

2. Elija Create Subnet (Crear subred).
3. En Name tag (Etiqueta de nombre), introduzca un nombre para la subred. Este nombre se utilizará más adelante al crear el host bastión en esta red.
4. En VPC, seleccione la VPC que creó anteriormente.
5. En Availability Zone (Zona de disponibilidad), seleccione una zona distinta de la que seleccionó para la primera subred pública.
6. En IPv4 CIDR block (Bloque de CIDR IPv4), escriba 10.0.2.0/24.
7. Elija Sí, crear. La subred se agrega a la lista de subredes.
8. Con la subred seleccionada en la lista, elija Subnet Actions (Acciones de subred), Modify auto-assign IP settings (Modificar configuración de asignación automática de IP). Seleccione Auto-assign IPs (Asignar automáticamente direcciones IP), Save (Guardar), Close (Cerrar). Esto permite que el host bastión obtenga una dirección IP pública al crearla en esta subred.
9. En la pestaña Route Table (Tabla de enrutamiento), elija Edit (Editar). En Change to (Cambiar a), seleccione el ID de la tabla de enrutamiento que anotó anteriormente y elija Save (Guardar).

Ahora debería ver tres subredes para la VPC que ha creado anteriormente. Anote los nombres y los ID de las subredes para que pueda especificarlos utilizando la plantilla.



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

Creación de los grupos de seguridad

Los grupos de seguridad funcionan como firewalls que controlan el tráfico entrante y saliente de los recursos. Cuando se utiliza la plantilla para crear una red Ethereum en un clúster de Amazon ECS, se deben especificar dos grupos de seguridad:

- Un grupo de seguridad para las instancias EC2 que controla el tráfico entrante y saliente de las instancias EC2 del clúster.

- Un grupo de seguridad para el equilibrador de carga de aplicaciones que controla el tráfico entre el equilibrador de carga de aplicaciones, las instancias EC2 y el host bastión. Asocie este grupo de seguridad también con el host bastión.

Cada grupo de seguridad tiene reglas que permiten la comunicación entre el equilibrador de carga de aplicaciones y las instancias EC2, así como otras reglas mínimas. Para ello, es necesario que los grupos de seguridad se hagan referencia entre sí. Por este motivo, primero debe crear los grupos de seguridad y, a continuación, actualizarlos con las reglas apropiadas.

Para crear dos grupos de seguridad

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En el panel de navegación, seleccione Security Groups, Create Security Group.
3. En Security group name (Nombre del grupo de seguridad), escriba un nombre para el grupo de seguridad que sea fácil de identificar y que lo distinga de los demás, como, por ejemplo, EthereumEC2-SG o EthereumALB-SG. Utilizará estos nombres más adelante. En Description (Descripción), escriba un breve resumen.
4. En VPC, seleccione la VPC que creó anteriormente.
5. Seleccione Crear.
6. Repita los pasos anteriores para crear el otro grupo de seguridad.

Adición de reglas de entrada al grupo de seguridad para las instancias EC2

1. Seleccione el grupo de seguridad para las instancias EC2 que creó anteriormente.
2. En la pestaña Inbound (Entrada), seleccione Edit (Editar).
3. En Tipo, seleccione Todo el tráfico. En Source (Origen), deje seleccionado Custom (Personalizado) y, a continuación, elija el grupo de seguridad que está editando en la lista, por ejemplo, EthereumEC2-SG. Esto permite que las instancias EC2 del grupo de seguridad se comuniquen entre sí.
4. Seleccione Add Rule (Agregar regla).
5. En Tipo, seleccione Todo el tráfico. En Source (Origen), deje seleccionado Custom (Personalizado) y, a continuación, elija el grupo de seguridad para el equilibrador de carga de aplicaciones en la lista, por ejemplo, EthereumALB-SG. Esto permite que las instancias EC2 del grupo de seguridad se comuniquen con el equilibrador de carga de aplicaciones.
6. Seleccione Guardar.

Adición de las reglas de entrada y edición de las reglas de salida del grupo de seguridad del equilibrador de carga de aplicaciones

1. Seleccione el grupo de seguridad para los equilibradores de carga de aplicaciones que creó anteriormente.
2. En la pestaña Inbound (Entrada), elija Edit (Editar) y, a continuación, agregue las reglas de entrada siguientes:
 - a. En Tipo, seleccione Todo el tráfico. En Source (Origen), deje seleccionado Custom (Personalizado) y, a continuación, elija el grupo de seguridad que está editando actualmente en la lista, por ejemplo, EthereumALB-SG. Esto permite que el equilibrador de carga de aplicaciones se comunique consigo mismo y con el host bastión.
 - b. Seleccione Add Rule (Agregar regla).
 - c. En Tipo, seleccione Todo el tráfico. En Source (Origen), deje seleccionado Custom (Personalizado) y, a continuación, elija el grupo de seguridad para las instancias EC2 de la lista, por ejemplo EthereumEC2-SG. Esto permite que las instancias EC2 del grupo de seguridad se comuniquen con el equilibrador de carga de aplicaciones y el host bastión.
 - d. Seleccione Add Rule (Agregar regla).
 - e. En Tipo, seleccione SSH. En Source (Origen), seleccione My IP (Mi IP), que detecta el CIDR de la IP de su equipo y lo escribe.

Important

Esta regla permite que el host bastión acepte tráfico SSH desde su equipo, lo que le permite usar el host bastión para ver interfaces web y conectarse a instancias EC2 en la red Ethereum. Para permitir que otros se conecten a la red Ethereum, agréguelos como orígenes a esta regla. Permitir solo el tráfico entrante a orígenes de confianza.

- f. Seleccione Guardar.
3. En la pestaña Outbound (Salida), elija Edit (Editar) y elimine la regla que se creó automáticamente para permitir el tráfico saliente con destino a todas las direcciones IP.
4. Seleccione Add Rule (Agregar regla).
5. En Tipo, seleccione Todo el tráfico. En Destination (Destino), deje Custom (Personalizado) seleccionado y, a continuación, elija el grupo de seguridad para las instancias EC2 de la lista.

Esto permite las conexiones salientes desde el equilibrador de carga de aplicaciones y el host bastión a instancias EC2 en la red Ethereum.

6. Seleccione Add Rule (Agregar regla).
7. En Tipo, seleccione Todo el tráfico. En Destination (Destino), deje Custom (Personalizado) seleccionado y, a continuación, elija el grupo de seguridad que está editando actualmente de la lista, por ejemplo, EthereumALB-SG. Esto permite que el equilibrador de carga de aplicaciones se comunique consigo mismo y con el host bastión.
8. Seleccione Guardar.

Creación de un rol de IAM; para Amazon ECS; y un perfil de instancia EC2

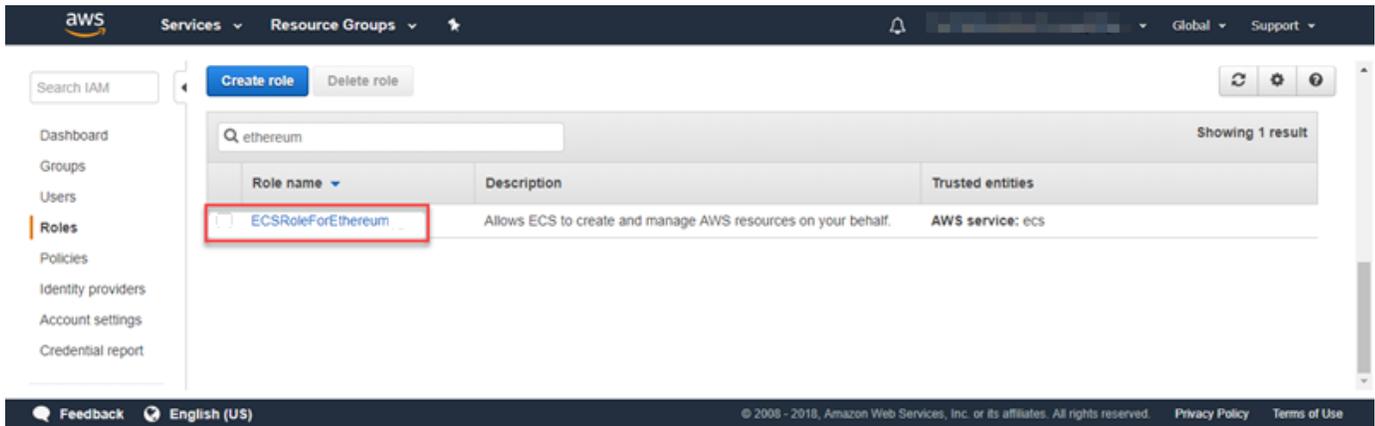
Cuando se utiliza esta plantilla, se especifica un rol de IAM; para Amazon ECS; y un perfil de instancia EC2. Las políticas de permisos asociadas a estos roles permiten a los recursos de AWS y a las instancias del clúster interactuar con otros recursos de AWS. Para obtener más información, consulte [Roles de IAM](#) en la Guía del usuario de IAM. El rol de IAM para Amazon ECS y el perfil de instancia de EC2 se configuran mediante la consola de IAM (<https://console.aws.amazon.com/iam/>).

Crear el rol de IAM para Amazon ECS

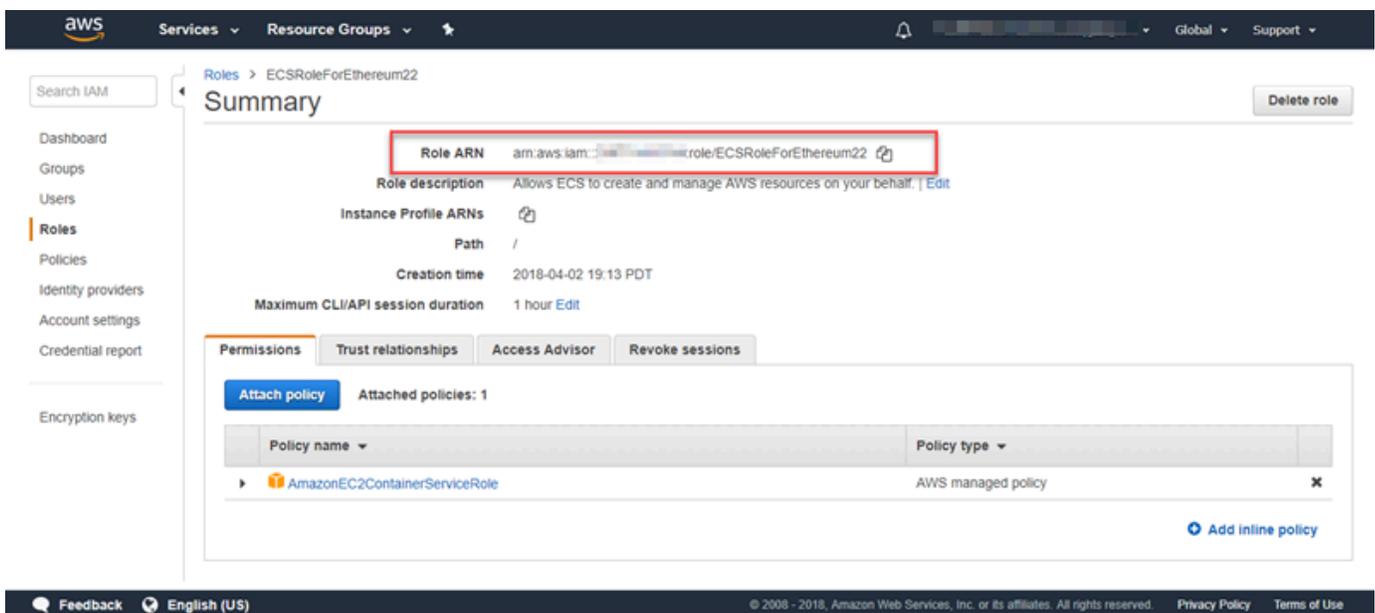
1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, elija Roles, Crear rol.
3. En Select type of trusted entity (Seleccionar tipo de entidad de confianza), elija AWS service.
4. En Choose the service that will use this role (Elegir el servicio que usará este rol), elija Elastic Container Service.
5. En Select your use case (Seleccione su caso de uso), elija Elastic Container Service (Servicio de contenedor elástico), Next:Permissions (Siguiendo: Permisos).

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. Para la política de permisos, deje seleccionada la política predeterminada (AmazonEC2ContainerServiceRole) y elija Next:Review.
7. En Nombre del rol, introduzca un valor que le ayude a identificar el rol, como ECS. RoleForEthereum En Role Description (Descripción del rol), escriba un breve resumen. Anote el nombre del rol para consultarlo más adelante.
8. Elija Crear rol.
9. Seleccione en la lista el rol que acaba de crear. Si su cuenta tiene varios roles, puede buscar el nombre de rol.



10. Copie el valor de Role ARN (ARN del rol) y guárdelo donde pueda copiarlo de nuevo. Necesitará este ARN al crear la red Ethereum.



Las instancias EC2 de la red Ethereum asumen el perfil de instancia EC2 que especifique en la plantilla para interactuar con otros AWS servicios. Primero debe crear una política de permisos para el rol, a continuación, crear el rol (que crea automáticamente un perfil de instancia con el mismo nombre) y, por último, asociar la política de permisos al rol.

Para crear un perfil de instancia EC2

1. En el panel de navegación, seleccione Políticas (Políticas), Create policy (Crear política).
2. Elija JSON y sustituya la instrucción de política predeterminada por la siguiente 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. Elija Revisar política.
4. En Nombre, introduce un valor que te ayude a identificar esta política de permisos, por ejemploEthereumPolicyFor, EC2. En Description (Descripción), escriba un breve resumen. Elija Crear política.

Create policy 1 2

Review policy

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

Description
Maximum 1000 characters. Use alphanumeric and '+, @, _' characters.

Summary

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. Elija Roles, Crear rol.
6. Elija EC2, Next: Permissions (Siguiente: Permisos).
7. En el campo de búsqueda, introduzca el nombre de la política de permisos que creó anteriormente, por ejemplo, EthereumPolicyForEC2.
8. Seleccione la marca de verificación situada junto a la política que ha creado antes y elija Next: Review (Siguiente: Revisar).

Create role 1 2 3

Attach permissions policies

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

Create policy Refresh

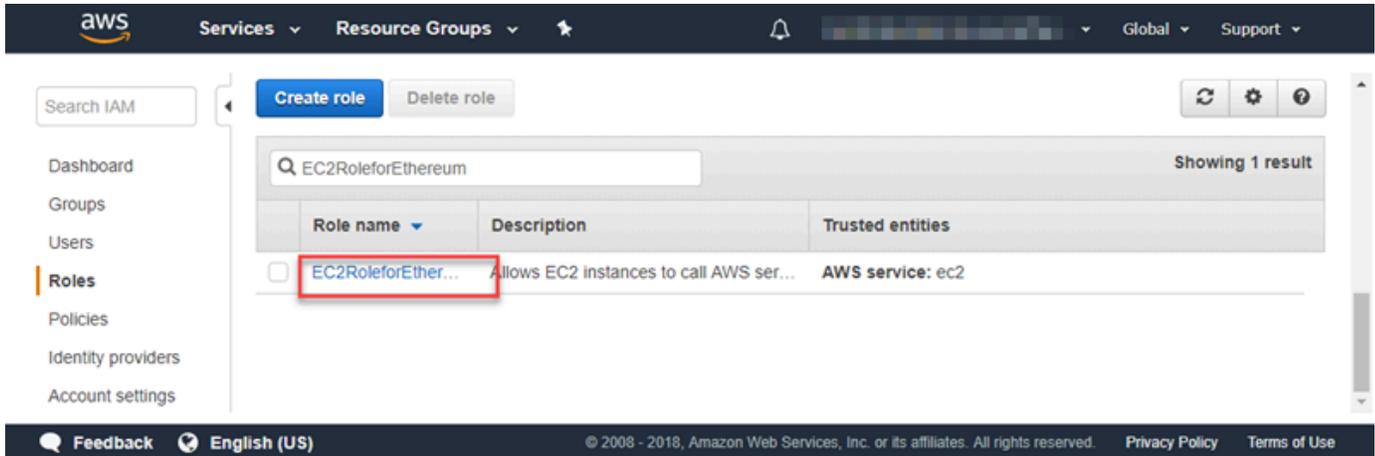
Filter: Policy type Showing 1 result

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

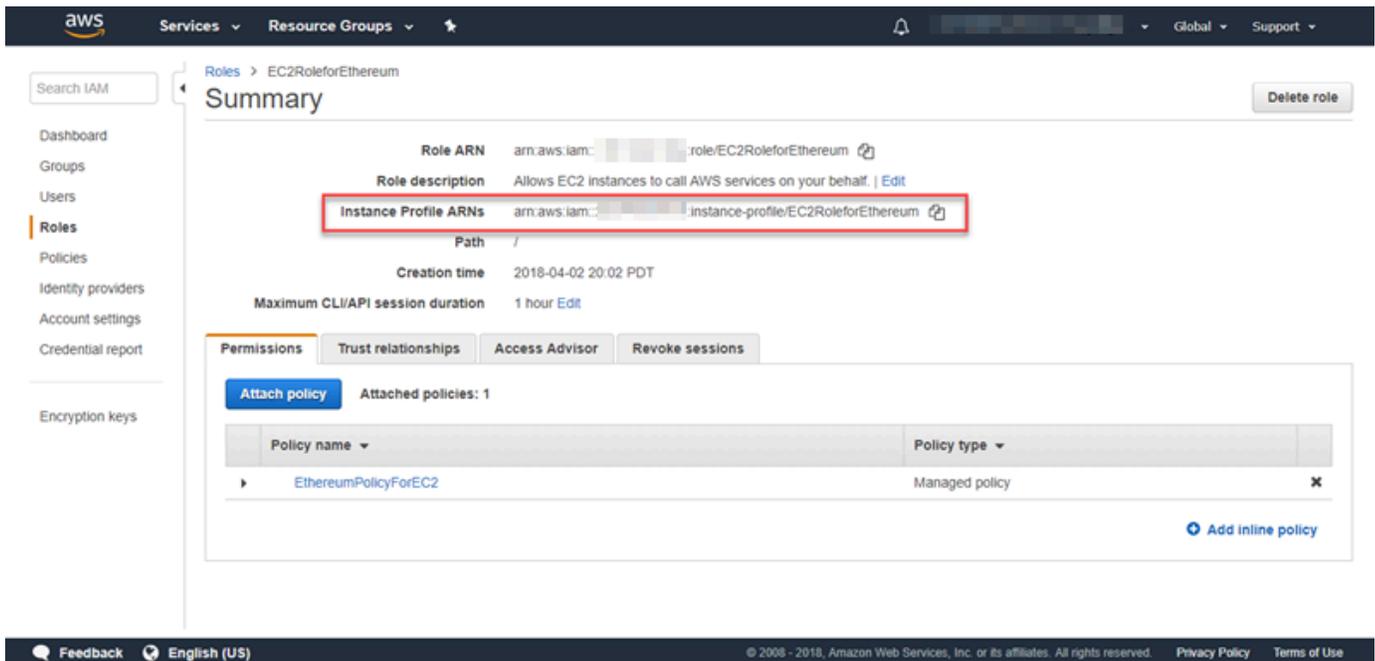
* Required Cancel Previous Next: Review

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

9. En Nombre del rol, introduzca un valor que le ayude a identificar el rol, por ejemplo RoleForEthereumEC2. En Role description (Descripción del rol), escriba un breve resumen. Elija Create role (Crear rol).
10. Seleccione en la lista el rol que acaba de crear. Si su cuenta tiene muchos roles, puede escribir el nombre del rol en el campo Search (Buscar).



11. Copie el valor de Instance Profile ARN (ARN del perfil de instancia) y guárdelo donde pueda copiarlo de nuevo. Necesitará este ARN al crear la red Ethereum.



Creación de un host bastión

En este tutorial, creará un host bastión. Se trata de una instancia EC2 que se utiliza para conectarse a las interfaces web e instancias de la red de Ethereum. Su único propósito es reenviar tráfico SSH desde clientes de confianza fuera de la VPC para que puedan acceder a los recursos de red de Ethereum.

El host bastión se configura porque el equilibrador de carga de aplicaciones que crea la plantilla es interno, lo que significa que solo enruta las direcciones IP internas. El host bastión:

- Tiene una dirección IP interna que el equilibrador de carga de aplicaciones reconoce porque la lanza en la segunda subred pública creada anteriormente.
- Tiene una dirección IP pública que asigna la subred, a la que pueden acceder orígenes de confianza fuera de la VPC.
- Está asociado con el grupo de seguridad para el equilibrador de carga de aplicaciones creado anteriormente, que tiene una regla de entrada que permite el tráfico SSH (puerto 22) de clientes de confianza.

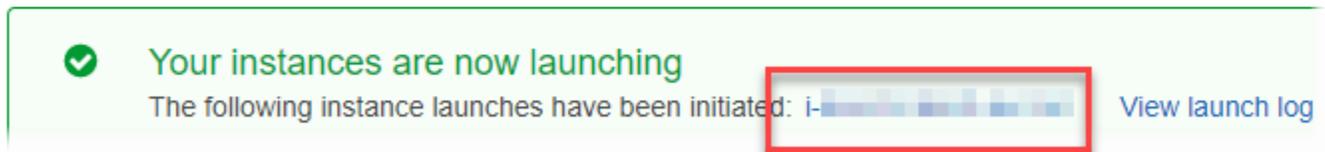
Para poder acceder a la red Ethereum, los clientes de confianza tienen que configurarse para conectarse a través del host bastión. Para obtener más información, consulte [Conectarse EthStats a Bastion Host y EthExplorer usarlo](#). Un host bastión es un enfoque. Puede utilizar cualquier enfoque que proporcione acceso desde clientes de confianza a recursos privados dentro de una VPC.

Para crear un host bastión

1. Siga los cinco primeros pasos para [lanzar una instancia](#) de la Guía del usuario de Amazon EC2.
2. Elija Edit Instance Details. En Network (Red), elija la VPC que creó anteriormente, en Subnet (Subred) seleccione la segunda subred pública que creó anteriormente. Deje el resto de opciones con sus valores predeterminados.
3. Confirme el cambio cuando se le solicite y, a continuación, elija Review and Launch (Revisar y lanzar).
4. Elija Edit Security Groups (Editar grupos de seguridad). En Assign a security group (Asignar un grupo de seguridad), seleccione Select an existing security group (Seleccionar un grupo de seguridad existente).
5. Desde la lista de grupos de seguridad, seleccione el grupo de seguridad para el equilibrador de carga de aplicaciones que creó anteriormente y, a continuación, elija Revisar y lanzar.

6. Elija Iniciar.
7. Anote el ID de la instancia. Lo necesitará más tarde cuando [Conectarse EthStats a Bastion Host y EthExplorer usarlo](#).

Launch Status



Creación de la red Ethereum

La red Ethereum que especifique mediante la plantilla de este tema lanza una AWS CloudFormation pila que crea un clúster Amazon ECS de instancias EC2 para la red Ethereum. La plantilla se basa en los recursos que ha creado anteriormente en [Configuración de requisitos previos](#).

Al lanzar la AWS CloudFormation pila con la plantilla, se crean pilas anidadas para algunas tareas. Cuando finalizan, puede conectarse a los recursos disponibles a través del equilibrador de carga de aplicaciones de la red a través del host bastión para verificar que la red Ethereum funciona y está accesible.

Pasos para crear la red Ethereum mediante AWS Blockchain Template para Ethereum

1. Consulte [Introducción a las plantillas de cadenas de bloques de AWS](#) y abra la plantilla de cadena de bloques de AWS más reciente para Ethereum en la AWS CloudFormation consola mediante los enlaces rápidos de su región de AWS.
2. Introduzca los valores de acuerdo con las directrices siguientes:
 - En Stack name (Nombre de la pila), escriba un nombre que sea fácil de identificar. Este nombre se utiliza dentro de los nombres de los recursos que crea la pila.
 - En Ethereum Network Parameters (Parámetros de la red Ethereum) y Private Ethereum Network Parameters (Parámetros de la red privada Ethereum), deje la configuración predeterminada.

⚠ Warning

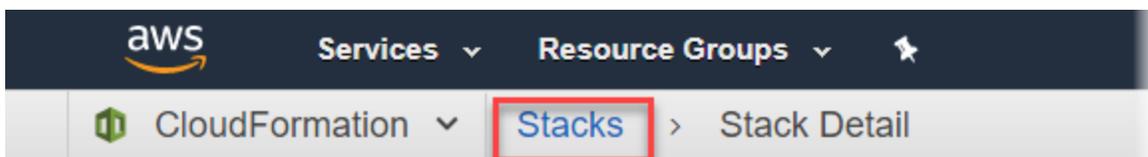
Utilice las cuentas predeterminadas y la frase mnemotécnica asociada solo con fines de prueba. No envíe Ether real usando el conjunto predeterminado de cuentas porque cualquier persona con acceso a la frase mnemotécnica puede acceder o robar Ether de las cuentas. En su lugar, especifique cuentas personalizadas para fines de producción. La frase mnemotécnica asociada a la cuenta predeterminada es `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

- En Configuración de la plataforma, deje la configuración predeterminada, que crea un clúster de Amazon ECS de instancias EC2. La alternativa, `docker-local`, crea una red Ethereum utilizando una sola instancia EC2.
- En EC2 configuration (Configuración de EC2), seleccione las opciones de acuerdo con las siguientes directrices:
 - En EC2 Key Pair (Par de claves de EC2), seleccione un par de claves. Para obtener información sobre cómo crear un par de claves, consulte [Creación de un par de claves](#).
 - En EC2 Security Group (Grupo de seguridad de EC2), seleccione el grupo de seguridad que creó anteriormente en [Creación de los grupos de seguridad](#).
 - En EC2 Instance Profile ARN (ARN del perfil de instancia EC2), escriba el ARN del perfil de instancia que creó anteriormente en [Creación de un rol de IAM; para Amazon ECS; y un perfil de instancia EC2](#).
- En VPC network configuration (Configuración de red de VPC), seleccione las opciones de acuerdo con las siguientes directrices:
 - En VPC ID (ID de VPC), seleccione la VPC que creó anteriormente en [Creación de una VPC y de subredes](#).
 - En Ethereum Network Subnet IDs (ID de subred de red Ethereum), seleccione la subred privada única que creó anteriormente en el procedimiento [To create the VPC](#).
- En ECS cluster configuration (Configuración del clúster ECS), deje los valores predeterminados. Esto crea un clúster ECS de tres instancias EC2.
- En Application Load Balancer configuration (ECS only) (Configuración del equilibrador de carga de aplicaciones (solo ECS)), seleccione las opciones de acuerdo con las siguientes directrices:

- En Application Load Balancer Subnet IDs (ID de subred de equilibrador de carga de aplicaciones), seleccione dos subredes públicas desde la [list of subnets](#) que anotó anteriormente.
 - En Application Load Balancer Security Group (Grupo de seguridad del equilibrador de carga de aplicaciones), seleccione el grupo de seguridad para el equilibrador de carga de aplicaciones que creó anteriormente en [Creación de los grupos de seguridad](#).
 - En IAM Role (Rol de IAM), escriba el ARN del rol de ECS que creó anteriormente en [Creación de un rol de IAM; para Amazon ECS; y un perfil de instancia EC2](#).
- En la sección EthStats, seleccione las opciones de acuerdo con las siguientes pautas:
 - EthStatsEn Deploy, deje la configuración predeterminada, que es verdadera.
 - En EthStats Connection Secret, escriba un valor arbitrario de seis caracteres como mínimo.
 - En EthExplorer, deje la configuración predeterminada de Deploy EthExplorer, que es verdadera.
 - En Other parameters (Otros parámetros), deje el valor predeterminado Nested Template S3 URL Prefix (Prefijo de URL S3 de plantilla anidada) y anótelo. Aquí es donde puede encontrar plantillas anidadas.
3. Deje todas las demás opciones con sus valores predeterminados, seleccione la casilla de confirmación y elija Create (Crear).

Aparece la página de detalles de la pila raíz que AWS CloudFormation se lanza.

4. Para monitorear el progreso de la pila raíz y las pilas anidadas, elija Stacks (Pilas).



MyFirstEthereumStack

Stack name: MyFirstEthereumStack

5. Cuando todas las pilas muestren CREATE_COMPLETE en Estado, puede conectarse a las interfaces de usuario de Ethereum para comprobar que la red funciona y está accesible. Cuando utiliza la plataforma de contenedores ECS, las URL para conectarse y el EthJson RPC a EthStats través del Application Load Balancer están disponibles en la pestaña Salidas de la pila raíz. EthExplorer

⚠ Important

No podrá conectarse directamente a estas URL o SSH directamente hasta que configure una conexión proxy a través del host bastión en su equipo cliente. Para obtener más información, consulte [Conectarse EthStats a Bastion Host y EthExplorer usarlo](#).

The screenshot shows the AWS CloudFormation console. The 'Stacks' page is active, displaying a list of stacks. The stack 'MyFirstEthereumStack' is selected. Below the stack list, the 'Outputs' tab is selected, showing a table of outputs. The 'EthStatsURL' output is highlighted with a red box. The URL is 'http://MyFir-...us-west-2.elb.amazonaws.com'. Other outputs include 'EthExplorerURL' and 'EthJsonRPCURL'.

Stack Name	Created Time	Status	Description
MyFirstEthereumStack-Ether...	2018-04-12 13:26:46 UTC-0700	CREATE_COMPLETE	This template creates an AutoScalingGroup of EC2 I...
MyFirstEthereumStack-Ether...	2018-04-12 13:26:38 UTC-0700	CREATE_COMPLETE	This template creates the ECS cluster and Ethereu...
MyFirstEthereumStack-Ether...	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:80	Visit this URL to view transactions on yo...	
EthJsonRPCURL	http://MyFir-...us-west-2.elb.amazonaws.com:85	Use this URL to access the Geth JSON ...	

Conectarse EthStats a Bastion Host y EthExplorer usarlo

Para conectarse a los recursos de Ethereum en este tutorial, configure el reenvío de puertos SSH (tunelización SSH) a través del host bastión. Las siguientes instrucciones muestran cómo hacerlo para poder conectarse a las URL EthStats y a las EthExplorer URL mediante un navegador. En las instrucciones siguientes, primero debe configurar un proxy SOCKS en un puerto local. A continuación, utilice una extensión de navegador, [FoxyProxy](#), para utilizar este puerto de reenvío para las URL de su red Ethereum.

Si utiliza Mac OS o Linux, utilice un cliente SSH para configurar la conexión proxy SOCKS con el host bastión. Si es usuario de Windows, use PuTTY. Antes de conectarse, confirme que el equipo cliente que está utilizando está especificado como origen permitido para el tráfico SSH entrante en el grupo de seguridad para el equilibrador de carga de aplicaciones que configuró anteriormente.

Para conectarse al host bastión con reenvío de puertos SSH mediante SSH

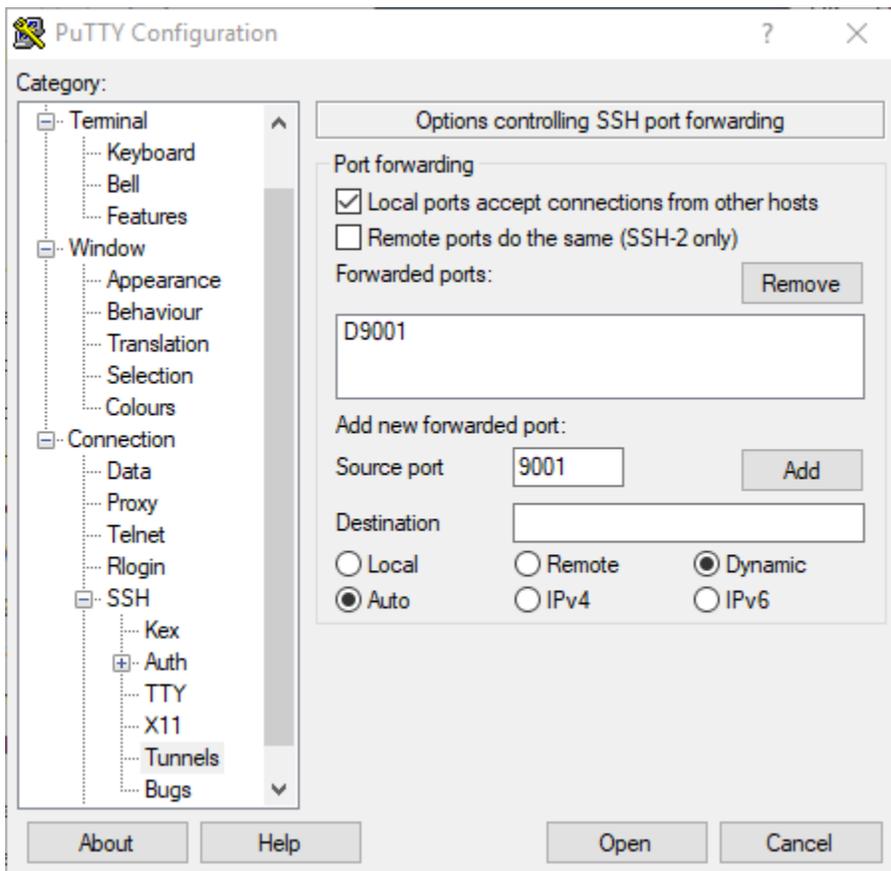
- Siga los procedimientos de [Conexión a su instancia de Linux mediante SSH](#) en la Guía del usuario de Amazon EC2. En el paso 4 del procedimiento [Conectarse a la instancia de Linux](#) agregue `-D 9001` al comando SSH, especifique el mismo par de claves que especificó en la configuración AWS Blockchain Template para Ethereum y especifique el nombre DNS del host bastión.

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

Para conectarse al host bastión con reenvío de puertos SSH mediante PuTTY (Windows)

1. Siga los procedimientos descritos en [Conexión a su instancia de Linux desde Windows mediante PuTTY](#) de la Guía del usuario de Amazon EC2 hasta el paso 7 del procedimiento [Inicio de una sesión de PuTTY](#), utilizando el mismo par de claves que especificó en la configuración de la plantilla de cadena de bloques de AWS para Ethereum.
2. En PuTTY, en Category (Categoría), elija Connection (Conexión), SSH, Tunnels (Túneles).
3. En Port forwarding (Reenvío de puertos), elija Local ports accept connections from other hosts (Los puertos locales aceptan conexiones de otros hosts).
4. En Add new forwarded port (Agregar nuevo puerto reenviado):
 - a. En Source port (Puerto de origen), escriba 9001. Este es un puerto arbitrario no utilizado que elegimos y puede elegir uno distinto si es necesario.
 - b. Deje Destination (Destino) en blanco.
 - c. Seleccione Dynamic (Dinámico).
 - d. Elija Añadir.

En Forwarded ports (Puertos reenviados), debería aparecer D9001 como se muestra debajo.



5. Elija Open (Abrir) y, a continuación, autentique en el host bastión según lo requiera la configuración de claves. Deje la conexión abierta.

Con la conexión PuTTY abierta, ahora configura su sistema o una extensión de navegador para usar el puerto reenviado para sus URL de su red Ethereum. Las siguientes instrucciones se basan en el uso del FoxyProxy estándar para reenviar las conexiones según el patrón de URL de EthStats y EthExplorer y el puerto 9001, que estableció anteriormente como puerto de reenvío, pero puede utilizar el método que prefiera.

FoxyProxy Para configurar el uso del túnel SSH para las URL de la red Ethereum

Este procedimiento se ha escrito basado en Chrome. Si utilizas otro navegador, traduce la configuración y la secuencia a la versión FoxyProxy de ese navegador.

1. Descargue e instale la extensión de navegador FoxyProxy estándar y, a continuación, abra las opciones según las instrucciones de su navegador.
2. Elija Add New Proxy (Agregar nuevo proxy).

3. En la pestaña General, asegúrese de que el proxy está Enabled (Habilitado) y escriba un Proxy Name (Nombre de proxy) y Proxy Notes (Notas de proxy) que le ayuden a identificar esta configuración de proxy.
4. En la pestaña Proxy Details (Detalles de proxy), elija Manual Proxy Configuration (Configuración manual de proxy). En Host or IP Address (Host o dirección IP) (o en Server or IP Address (Servidor o dirección IP) en algunas versiones), escriba localhost. En Port (Puerto), escriba 9001. Seleccione SOCKS Proxy?.
5. En la pestaña URL Pattern (Patrón de URL), elija Add New Pattern (Agregar nuevo patrón).
6. En Pattern name, introduce un nombre que sea fácil de identificar y, en URL Pattern, introduce un patrón que coincida con todas las URL de recursos de Ethereum que creaste con la plantilla, por ejemplo, http://internal - MyUser -loadB-*. Para obtener información sobre cómo ver direcciones URL, consulte [Ethereum URLs](#).
7. Deje las opciones predeterminadas en otros ajustes y elija Save (Guardar).

Ahora puedes conectarte a las URL de Ethereum, que están disponibles en la CloudFormation consola, mediante la pestaña Salidas de la pila raíz que creaste con la plantilla.

Eliminación de recursos de

AWS CloudFormation facilita la limpieza de los recursos que creó la pila. Cuando se elimina la pila, se eliminan todos los recursos creados por ella.

Para eliminar los recursos creados por la plantilla

- Abre la AWS CloudFormation consola, selecciona la pila raíz que creaste anteriormente, selecciona Acciones y Eliminar.

El Status (Estado) de la pila raíz que ha creado anteriormente y las pilas anidadas asociadas cambia a DELETE_IN_PROGRESS.

Si lo desea, puede eliminar los requisitos previos que ha creado para la red Ethereum.

Eliminación de la VPC

- Abra la consola de Amazon VPC, seleccione la VPC que creó anteriormente y, a continuación, elija Acciones, Eliminar VPC. Esto también elimina las subredes, los grupos de seguridad y la gateway NAT que están asociados a la VPC.

Eliminación del rol de IAM y el perfil de instancia EC2

- Abra la consola de IAM; y elija Roles. Seleccione el rol de ECS y el rol de EC2 que creó anteriormente, y elija Delete (Eliminar).

Terminar la instancia EC2 para el host bastión

- Abra el panel de Amazon EC2, elija Ejecutando instancias, seleccione la instancia EC2 que creó para el host bastión, elija Acciones, Estado de instancia, Terminar.

AWS Blockchain Templates y sus características

En esta sección, se proporcionan enlaces para que comience a crear una red de cadena de bloques inmediatamente, así como información sobre las opciones de configuración y los requisitos previos para la configuración de la red en AWS.

Están disponibles las siguientes plantillas:

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

AWS Blockchain Templates está disponible en las siguientes regiones:

- Región Oeste de EE. UU. (Oregón) (us-west-2)
- Región Este de EE. UU. (Norte de Virginia) (us-east-1)
- Región Este de EE. UU. (Ohio) (us-east-2)

Note

Si se ejecuta una plantilla en una región distinta de las mencionadas arriba, se lanzarán recursos en la región Este de EE. UU. (Norte de Virginia) (us-east-1).

Uso de AWS Blockchain Template para Ethereum

Ethereum es un marco de trabajo de cadena de bloques que administra contratos inteligentes mediante Solidity, un lenguaje específico de Ethereum. Homestead es la versión más reciente de Ethereum. Para obtener más información, consulte la [documentación de Ethereum Homestead](#) y la documentación de [Solidity](#).

Enlaces para el lanzamiento

Consulte [Introducción a AWS Blockchain Templates](#) para ver los vínculos para lanzar AWS CloudFormation en regiones específicas utilizando las plantillas de Ethereum.

Opciones de Ethereum

Al configurar la red Ethereum con la plantilla, debe elegir opciones que determinan los requisitos siguientes:

- [Elección de la plataforma de contenedores](#)
- [Elección de una red de Ethereum pública o privada](#)
- [Cambio de las cuentas predeterminadas y la frase mnemotécnica](#)

Elección de la plataforma de contenedores

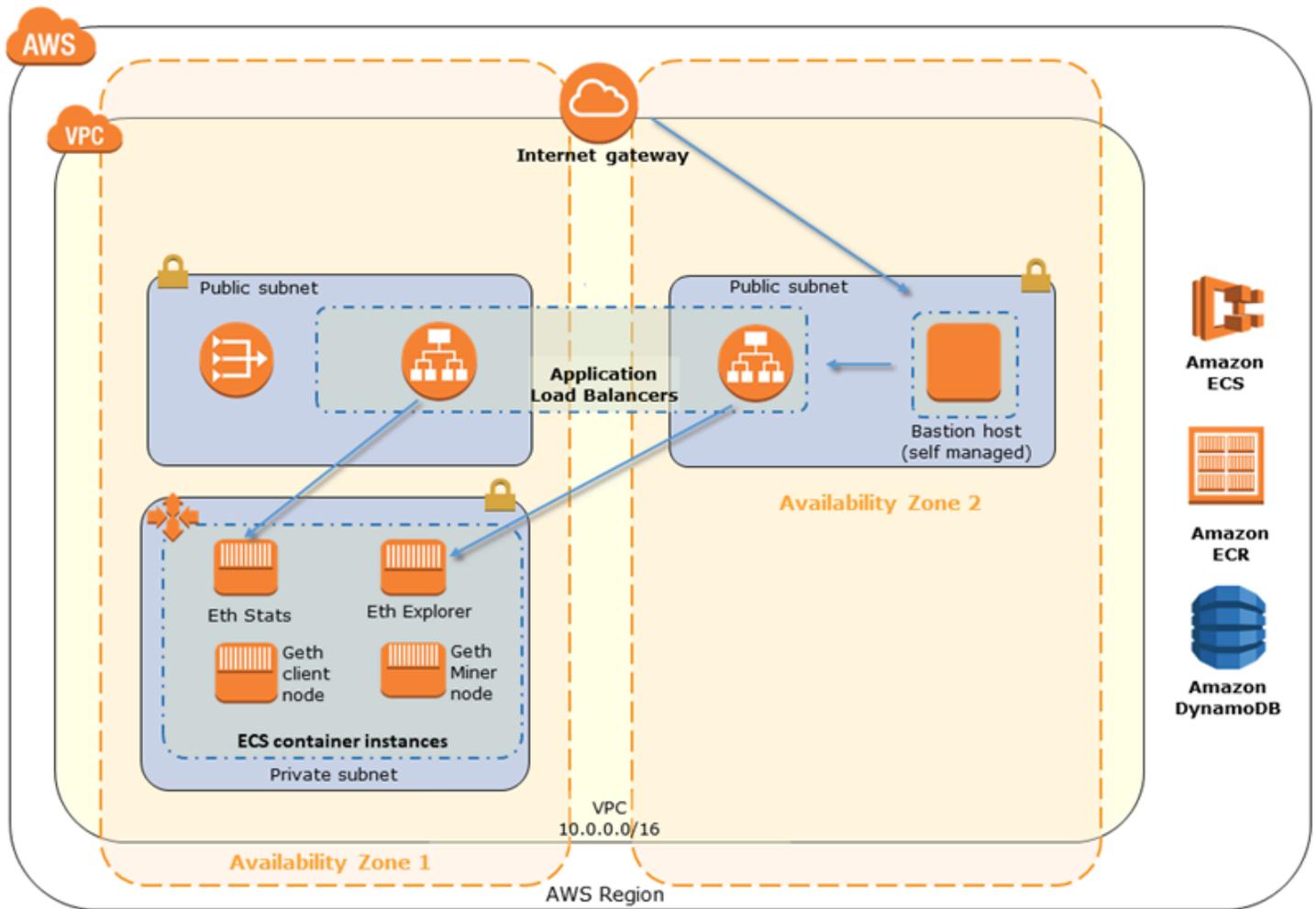
Las AWS Blockchain Templates utilizan contenedores de Docker almacenados en Amazon ECR para implementar un software de cadena de bloques. La AWS Blockchain Template para Ethereum ofrece dos opciones de plataforma de contenedores:

- `ecs`: especifica que Ethereum se ejecuta en un clúster de Amazon ECS de instancias de Amazon EC2.
- `docker-local`: especifica que Ethereum se ejecuta en una sola instancia EC2.

Uso de la plataforma de contenedores Amazon ECS

Con Amazon ECS, debe crear la red de Ethereum en un clúster de ECS formado por varias instancias EC2, con un equilibrador de carga de aplicaciones y con los recursos relacionados. Para obtener más información sobre cómo utilizar la configuración de Amazon ECS, consulte el tutorial [Introducción a AWS Blockchain Templates](#).

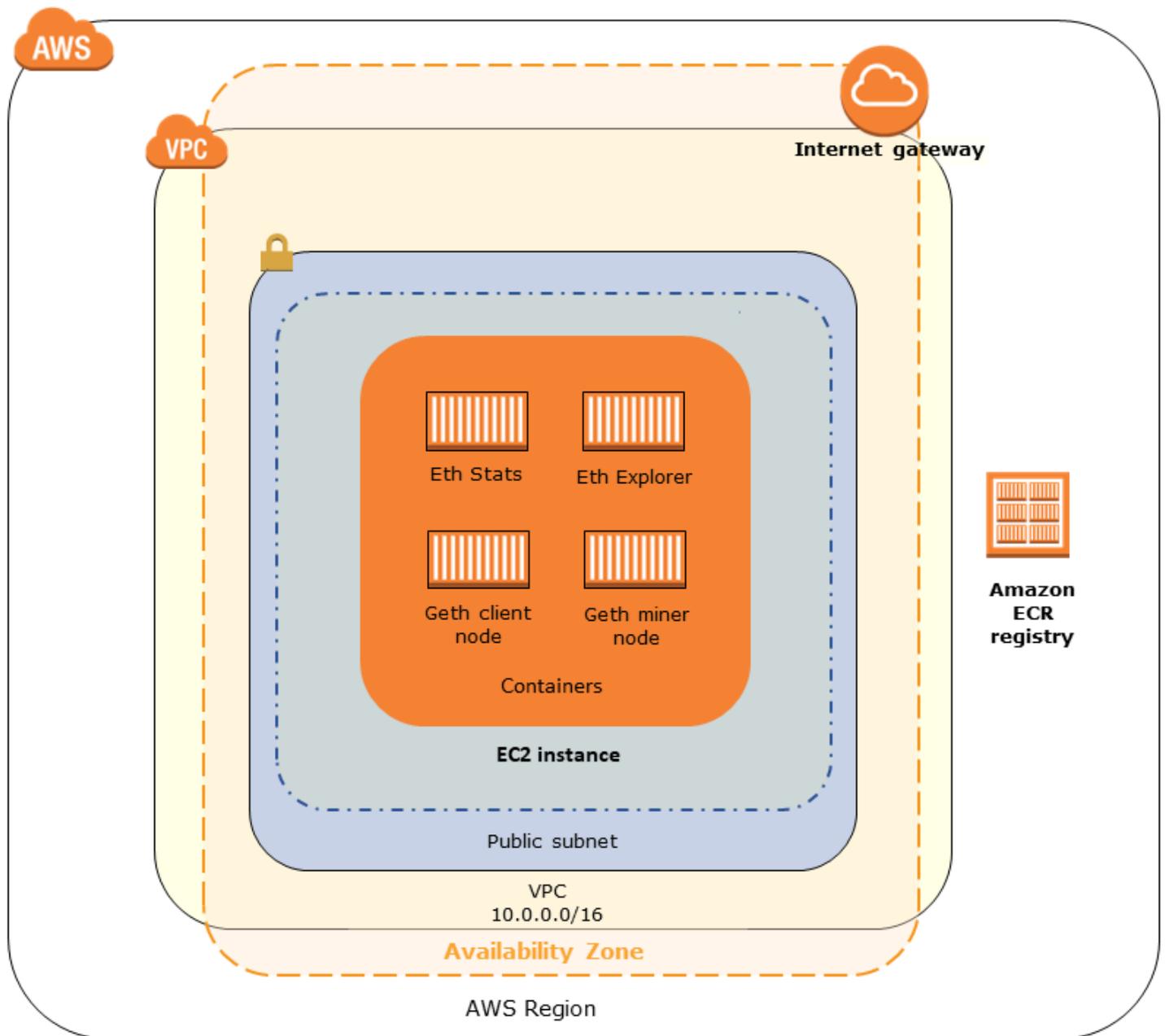
El siguiente diagrama muestra una red de Ethereum creada utilizando la plantilla con la opción de plataforma de contenedores ECS:



Uso de la plataforma docker-local

También puede lanzar contenedores de Ethereum dentro de una única instancia Amazon EC2. Todos los contenedores se ejecutan en una sola instancia EC2. Se trata de una configuración simplificada.

El siguiente diagrama muestra una red de Ethereum creada utilizando la plantilla con la opción de plataforma de contenedores docker-local:



Elección de una red de Ethereum pública o privada

Si se elige un valor distinto de 1-4 para Ethereum Network ID (ID de la red Ethereum), se crean nodos privados de Ethereum que se ejecutan dentro de la red que se defina, utilizando los parámetros de red privada que se especifiquen.

Si se elige un valor de 1-4 para ID de la red Ethereum, los nodos de Ethereum que se crean se unen a la red Ethereum pública. Puede hacer caso omiso de la configuración de red privada y de

sus valores predeterminados. Si decide unir los nodos de Ethereum a la red Ethereum pública, asegúrese de que es posible tener acceso a los servicios adecuados de la red a través de Internet.

Cambio de las cuentas predeterminadas y la frase mnemotécnica

Una frase mnemotécnica es un conjunto aleatorio de palabras que puede usar para generar carteras de Ethereum (es decir, pares de claves privadas/públicas) para cuentas asociadas en cualquier red. La frase mnemotécnica se puede utilizar para acceder a Ether para cuentas asociadas. Creamos una frase mnemotécnica predeterminada asociada a las cuentas predeterminadas que utiliza la plantilla de Ethereum.

Warning

Utilice las cuentas predeterminadas y la frase mnemotécnica asociada solo con fines de prueba. No envíe Ether real usando el conjunto predeterminado de cuentas porque cualquier persona con acceso a la frase mnemotécnica puede acceder o robar Ether de las cuentas. En su lugar, especifique cuentas personalizadas para fines de producción. La frase mnemotécnica asociada a la cuenta predeterminada es `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

Requisitos previos

Cuando se configura la red de Ethereum utilizando la AWS Blockchain Template para Ethereum, se deben cumplir los requisitos mínimos que se indican a continuación. La plantilla requiere los componentes de AWS enumerados para cada una de las categorías siguientes:

Temas

- [Requisitos previos para acceder a los recursos de Ethereum](#)
- [Requisitos previos de IAM](#)
- [Requisitos previos del grupo de seguridad](#)
- [Requisitos previos de VPC](#)
- [Ejemplo de permisos de IAM para el perfil de instancia EC2 y el rol de ECS](#)

Requisitos previos para acceder a los recursos de Ethereum

Requisito previo	Para la plataforma ECS	Para docker-local
Un par de claves de Amazon EC2 que se puede utilizar para obtener acceso a las instancias EC2. La clave debe existir en la misma región que el clúster y los demás recursos de ECS.	✓	✓
Un componente con conexión a Internet, como un host bastión o un equilibrador de carga con conexión a Internet, con una dirección interna desde la que se permite el tráfico en el equilibrador de carga de aplicaciones. Esto es necesario con la plataforma ECS porque la plantilla crea un equilibrador de carga interno por razones de seguridad. Esto es necesario con la plataforma docker-local cuando la instancia EC2 se encuentra en una subred privada, lo que recomendamos. Para obtener información sobre la configuración de un host bastión, consulte Creación de un host bastión .	✓	✓ (con subred privada)

Requisitos previos de IAM

Requisito previo	Para la plataforma ECS	Para docker-local
Una entidad principal (usuario o grupo) de IAM que tenga permisos para trabajar con todos los servicios relacionados.	✓	✓
Un perfil de instancia Amazon EC2 con los permisos adecuados para que las instancias EC2 interactúen con los demás servicios. Para obtener más información, consulte To create an EC2 instance profile .	✓	✓
Un rol de IAM con los permisos adecuados para que Amazon ECS interactúe con los demás servicios. Para obtener más información, consulte Creación del rol y los permisos de ECS .	✓	

Requisitos previos del grupo de seguridad

Requisito previo	Para la plataforma ECS	Para docker-local
Un grupo de seguridad para las instancias EC2, con los siguientes requisitos:	✓	✓
<ul style="list-style-type: none"> Reglas de salida que permitan el tráfico a 	✓	✓

Requisito previo	Para la plataforma ECS	Para docker-local
0.0.0.0/0 (valor predeterminado).		
<ul style="list-style-type: none"> Una regla de entrada que permite todo el tráfico de sí misma (el mismo grupo de seguridad). 	✓	✓
<ul style="list-style-type: none"> Una regla de entrada que permite todo el tráfico del grupo de seguridad para el equilibrador de carga de aplicaciones. 	✓	
<ul style="list-style-type: none"> Reglas de entrada que permiten HTTP (puerto 80), EthStats (servido en el puerto 8080), JSON RPC a través de HTTP (puerto 8545) y SSH (puerto 22) desde fuentes externas de confianza, como el CIDR IP del equipo cliente. 		✓

Requisito previo	Para la plataforma ECS	Para docker-local
<p>Un grupo de seguridad para el equilibrador de carga de aplicaciones, con los siguientes requisitos:</p> <ul style="list-style-type: none"> • Una regla de entrada que permite todo el tráfico de sí misma (el mismo grupo de seguridad). • Una regla de entrada que permita todo el tráfico desde el grupo de seguridad a las instancias EC2. • Las reglas de salida que permiten todo el tráfico únicamente al grupo de seguridad para las instancias EC2. Para obtener más información, consulte Creación de los grupos de seguridad. • Si asocia este mismo grupo de seguridad con un host bastión, una regla de entrada que permite el tráfico SSH (puerto 22) de orígenes de confianza. • Si el host bastión u otro componente con conexión a Internet se encuentra en un grupo de seguridad diferente, una regla de entrada que permite 	✓	

Requisito previo	Para la plataforma ECS	Para docker-local
el tráfico desde ese componente.		

Requisitos previos de VPC

Requisito previo	Para la plataforma ECS	Para docker-local
Una dirección IP elástica, que se utiliza para tener acceso a los servicios de Ethereum.	✓	✓
Una subred para ejecutar las instancias EC2. Recomendamos el uso de una subred privada.	✓	✓
Dos subredes con acceso público. Cada subred debe estar en distintas zonas de disponibilidad, con una en la misma zona de disponibilidad que la subred de las instancias EC2.	✓	

Ejemplo de permisos de IAM para el perfil de instancia EC2 y el rol de ECS

Debe especificar el ARN de un perfil de instancia EC2 como uno de los parámetros cuando utilice la plantilla. Si utiliza la plataforma de contenedores ECS, también debe especificar el ARN de un rol de ECS. Las políticas de permisos asociadas a estos roles permiten a los recursos de AWS y a las instancias del clúster interactuar con otros recursos de AWS. Para obtener más información, consulte [Roles de IAM](#) en la Guía del usuario de IAM. Utilice las instrucciones de política y los procedimientos siguientes como punto de partida para la creación de los permisos.

Ejemplo de política de permisos para el perfil de instancia EC2

La siguiente política de permisos demuestra las acciones permitidas para el perfil de instancia EC2 cuando se elige la plataforma de contenedores ECS. Se pueden utilizar las mismas instrucciones de política en una plataforma de contenedores docker-local, quitando las claves de contexto ecs para limitar el acceso.

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

Creación del rol y los permisos de ECS

Para los permisos asociados a la función de ECS, le recomendamos que comience con la política de permisos de AmazonEC2. ContainerServiceRole Utilice el procedimiento siguiente para crear un rol y asociarle esta política de permisos. Utilice la consola de IAM para ver la mayoría de los up-to-date permisos de esta política.

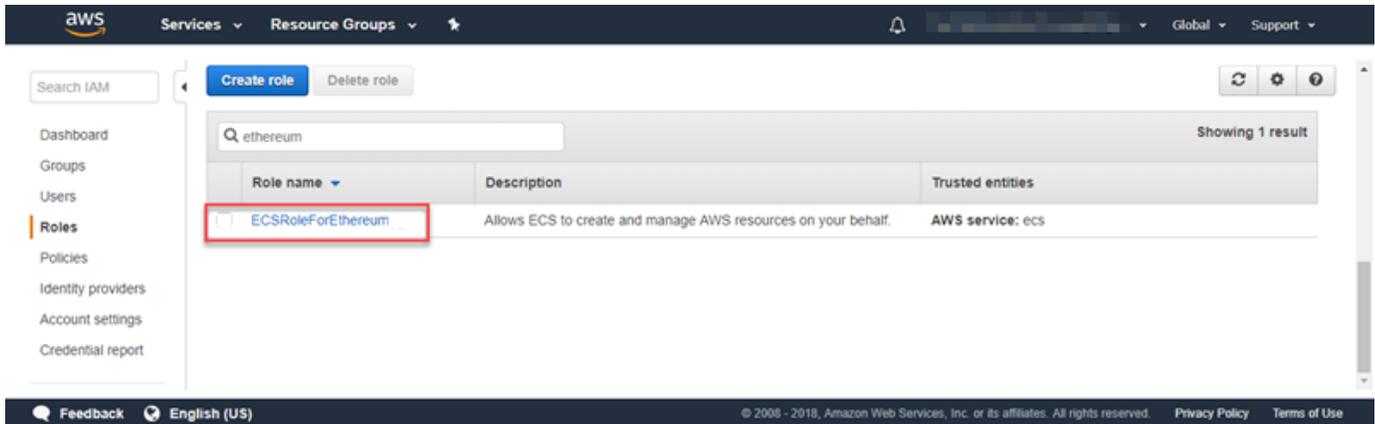
Crear el rol de IAM para Amazon ECS

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, elija Roles, Crear rol.
3. En Select type of trusted entity (Seleccionar tipo de entidad de confianza), elija AWS service.
4. En Choose the service that will use this role (Elegir el servicio que usará este rol), elija Elastic Container Service.
5. En Select your use case (Seleccione su caso de uso), elija Elastic Container Service (Servicio de contenedor elástico), Next:Permissions (Siguiente: Permisos).

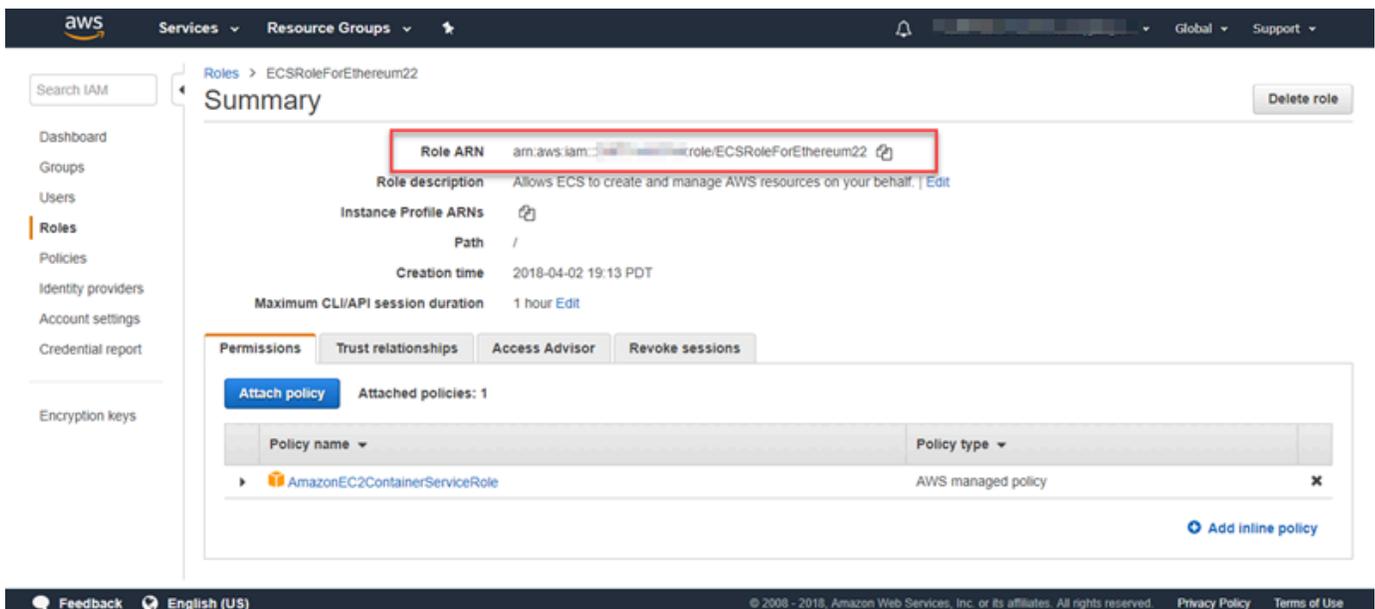
The screenshot shows the AWS IAM console 'Create role' wizard. The first step, 'Select type of trusted entity', has 'AWS service' selected. The second step, 'Choose the service that will use this role', displays a grid of services. 'Elastic Container Service' is highlighted with a red box. The third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

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

6. Para la política de permisos, deje seleccionada la política predeterminada (AmazonEC2ContainerServiceRole) y elija Next:Review.
7. En Nombre del rol, introduzca un valor que le ayude a identificar el rol, como ECS. RoleForEthereum En Role Description (Descripción del rol), escriba un breve resumen. Anote el nombre del rol para consultarlo más adelante.
8. Elija Crear rol.
9. Seleccione en la lista el rol que acaba de crear. Si su cuenta tiene varios roles, puede buscar el nombre de rol.



10. Copie el valor de Role ARN (ARN del rol) y guárdelo donde pueda copiarlo de nuevo. Necesitará este ARN al crear la red Ethereum.



Conexión a Ethereum Resources

Una vez que la pila raíz creada con la plantilla muestre CREATE_COMPLETE, puede conectarse a los recursos de Ethereum mediante la consola de AWS CloudFormation. La forma de conectarse depende de la plataforma de contenedores elegida, ECS o docker-local:

- ECS: la pestaña Salida de la pila raíz proporciona enlaces a los servicios que se ejecutan en el equilibrador de carga de aplicaciones. No es posible tener acceso directo a estas direcciones URL por motivos de seguridad. Para conectarse a ellas, puede configurar y utilizar un host bastión para que actúe como proxy durante la conexión. Para obtener más información, consulte [Conexiones proxy mediante un host bastión](#) más abajo.
- docker-local: puede conectarse utilizando la dirección IP de la instancia EC2 que aloja los servicios de Ethereum como se indica a continuación. Utilice la consola de EC2 para encontrar la *dirección-IP-de-ec2* de la instancia creada por la plantilla.
 - EthStats—*Utilice la dirección IP de EC2 http://*
 - EthExplorer— *Utilice la dirección IP http://EC2-IP:8080*
 - EthJsonRpc— Utilice la *dirección IP http://EC2-IP:8545*

Si ha especificado una subred pública para Ethereum Network Subnet ID (ID de subred de la red Ethereum) (List of VPC Subnets to use (Lista de subredes de VPC que se van a utilizar) dentro de la plantilla), puede conectarse directamente. El cliente debe ser un origen de confianza de tráfico entrante para SSH (puerto 22), así como los puertos de la lista. Esto lo determina el grupo de seguridad de EC2 que se especificó mediante la AWS Blockchain Template para Ethereum.

Si ha especificado una subred privada, puede configurar y utilizar un host bastión para que actúe como proxy en las conexiones con estas direcciones. Para obtener más información, consulte [Conexiones proxy mediante un host bastión](#) más abajo.

Conexiones proxy mediante un host bastión

En algunas configuraciones, es posible que los servicios de Ethereum no estén disponibles públicamente. En esos casos, puede conectarse a los recursos de Ethereum a través de un host bastión. Para obtener más información sobre los hosts bastión, consulte [Arquitectura de host bastión de Linux](#) en la Guía de inicio rápido del host bastión de Linux.

El host bastión es una instancia EC2. Asegúrese de que se cumplan los siguientes requisitos:

- La instancia EC2 del host bastión se encuentra dentro de una subred pública con la asignación automática de IP pública habilitada y que tiene una puerta de enlace de Internet.
- El host bastión tiene el par de claves que permite conexiones SSH.
- El host bastión está asociado a un grupo de seguridad que permite el tráfico SSH entrante desde los clientes que se conecten.
- El grupo de seguridad asignado a los hosts de Ethereum (por ejemplo, el equilibrador de carga de aplicaciones si ECS es la plataforma de contenedores, o la instancia EC2 del host si docker-local es la plataforma de contenedores) permite el tráfico entrante en todos los puertos desde los orígenes que se encuentren en la VPC.

Si tiene un host bastión configurado, asegúrese de que los clientes que se conecten utilicen el host bastión como proxy. En el siguiente ejemplo, se muestra cómo establecer una conexión de proxy mediante Mac OS. Sustituya *BastionIP* por la dirección IP de la instancia EC2 del host del bastión y *MySshKey.pem* por el archivo de pares de claves que copió en el host del bastión.

En la línea de comando, escriba lo siguiente:

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

Así se configura el reenvío de puertos del puerto 9001 de la máquina local al host bastión.

A continuación, configure su navegador o sistema para que utilice el proxy SOCKS para localhost:9001. Por ejemplo, si usa Mac OS, seleccione System preferences (Preferencias del Sistema), Network (Red), Advanced (Avanzado), seleccione SOCKS proxy y escriba localhost:9001.

Si utilizas la opción FoxyProxy Estándar con Chrome, selecciona Más herramientas y extensiones. En FoxyProxy Estándar, selecciona Detalles, Opciones de extensión y Añadir nuevo proxy. Seleccione Manual Proxy Configuration (Configuración manual del proxy). En Host or IP Address (Host o dirección IP) escriba localhost y en Port (Puerto) escriba 9001. Seleccione SOCKS Proxy?, Save (Guardar).

Ahora debería poder conectarse a las direcciones de host enumeradas en la salida de la plantilla.

Uso de AWS Blockchain Template para Hyperledger Fabric

Hyperledger Fabric es un marco de trabajo de cadena de bloques que ejecuta contratos inteligentes denominados código de cadena escritos en Go. Es posible crear una red privada con Hyperledger

Fabric para limitar los homólogos que se pueden conectar a la red y participar en ella. Para obtener más información sobre Hyperledger Fabric, consulte la documentación de [Hyperledger Fabric](#). Para obtener más información sobre los códigos de cadena, consulte el tema [Chaincode for Developers](#) en la documentación de [Hyperledger Fabric](#).

La AWS Blockchain Template para Hyperledger Fabric solo es compatible con una plataforma de contenedores docker-local, lo que significa que los contenedores de Hyperledger Fabric se implementan en una sola instancia EC2.

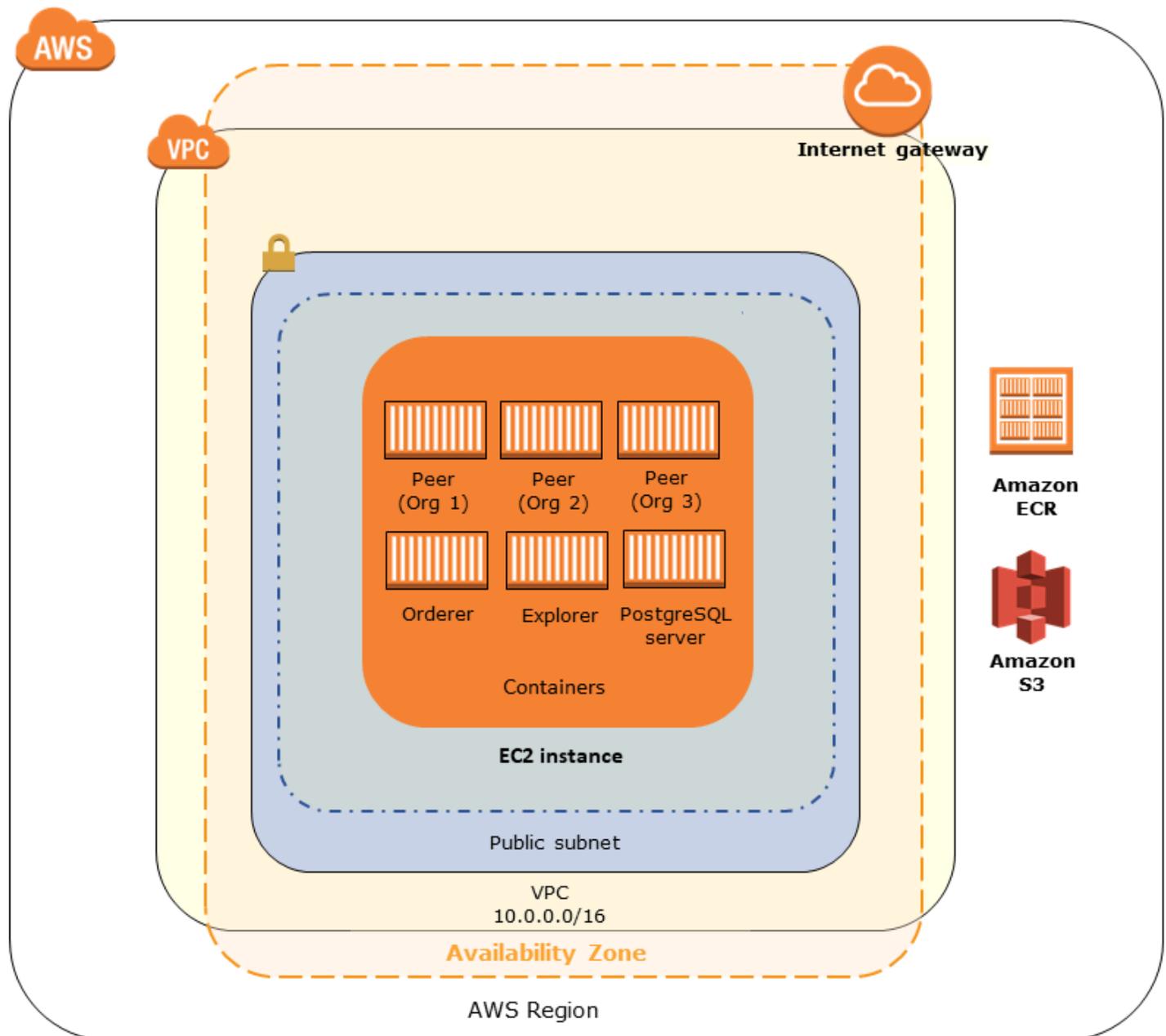
Enlaces para el lanzamiento

Consulte [Introducción a las plantillas de Blockchain de AWS](#) para ver los enlaces que permiten lanzar AWS CloudFormation en regiones específicas mediante las plantillas de Hyperledger Fabric.

AWS Blockchain Template para componentes de Hyperledger Fabric

La AWS Blockchain Template para Hyperledger Fabric crea una instancia EC2 con Docker y lanza una red Hyperledger Fabric utilizando los contenedores de dicha instancia. La red incluye un servicio de orden y tres organizaciones, cada una con un servicio de homólogos. La plantilla también lanza un contenedor de Hyperledger Explorer, que le permite examinar los datos de cadena de bloques. Se lanza un contenedor de servidor de PostgreSQL para admitir Hyperledger Explorer.

El siguiente diagrama muestra una red de Hyperledger Fabric creada con la plantilla:



Requisitos previos

Antes de lanzar una red de Hyperledger Fabric con la plantilla, asegúrese de que se cumplen los requisitos siguientes:

- La entidad principal (usuario o grupo) de IAM que utilice debe tener permiso para trabajar con todos los servicios relacionados.

- Se debe tener acceso a un par de claves que se puede utilizar para obtener acceso a las instancias EC2 (por ejemplo, mediante SSH). La clave debe existir en la misma región que la instancia.
- Se debe tener un perfil de instancia EC2 con una política de permisos asociada que permita el acceso a Amazon S3 y a Amazon Elastic Container Registry (Amazon ECR) para extraer contenedores. Si desea ver un ejemplo de política de permisos, consulte [Ejemplo de permisos de IAM para el perfil de instancia EC2](#).
- Debe tener una red de Amazon VPC con una subred pública o una subred privada con una puerta de enlace NAT y una dirección IP elástica para poder acceder a Amazon S3 AWS CloudFormation y Amazon ECR.
- Se debe tener un grupo de seguridad de EC2 con reglas de entrada que permitan el tráfico SSH (puerto 22) desde las direcciones IP que necesiten conectarse a la instancia mediante SSH, y lo mismo para los clientes que necesiten conectarse a Hyperledger Explorer (puerto 8080).

Ejemplo de permisos de IAM para el perfil de instancia EC2

Debe especificar el ARN de un perfil de instancia EC2 como uno de los parámetros cuando utilice la AWS Blockchain Template para Hyperledger Fabric. Utilice la siguiente instrucción de política como punto de partida para la política de permisos asociada a ese rol y ese perfil de instancia de 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": "*"
    }
  ]
}
```

```
    }  
  ]  
}
```

Conexión a los recursos de Hyperledger Fabric

Una vez que la pila raíz creada con la plantilla muestre `CREATE_COMPLETE`, puede conectarse a los recursos de Hyperledger Fabric de la instancia EC2. Si ha especificado una subred pública, puede conectarse a la instancia EC2 al igual que haría con cualquier otra instancia EC2. Para obtener más información, consulte [Conexión a su instancia de Linux mediante SSH](#) en la Guía del usuario de Amazon EC2.

Si ha especificado una subred privada, puede configurar y utilizar un host bastión para que actúe como proxy en las conexiones a los recursos de Hyperledger Fabric. Para obtener más información, consulte [Conexiones proxy mediante un host bastión](#) más abajo.

Note

Es posible que observe que la plantilla asigna una dirección IP pública a la instancia EC2 que aloja los servicios de Hyperledger Fabric; sin embargo, esta dirección IP no está accesible públicamente, ya que las políticas de direccionamiento de la subred privada especificada no permiten el tráfico entre esta dirección IP y los orígenes públicos.

Conexiones proxy mediante un host bastión

En algunas configuraciones, es posible que los servicios de Hyperledger Fabric no estén disponibles públicamente. En esos casos, puede conectarse a los recursos de Hyperledger Fabric a través de un host bastión. Para obtener más información sobre los hosts bastión, consulte [Arquitectura de host bastión de Linux](#) en la Guía de inicio rápido del host bastión de Linux.

El host bastión es una instancia EC2. Asegúrese de que se cumplan los siguientes requisitos:

- La instancia EC2 del host bastión se encuentra dentro de una subred pública con la asignación automática de IP pública habilitada y que tiene una puerta de enlace de Internet.
- El host bastión tiene el par de claves que permite conexiones SSH.
- El host bastión está asociado a un grupo de seguridad que permite el tráfico SSH entrante desde los clientes que se conecten.

- El grupo de seguridad asignado a los hosts de Hyperledger Fabric (por ejemplo, el equilibrador de carga de aplicaciones si ECS es la plataforma de contenedores, o la instancia EC2 del host si docker-local es la plataforma de contenedores) permite el tráfico entrante en todos los puertos desde los orígenes que se encuentren en la VPC.

Si tiene un host bastión configurado, asegúrese de que los clientes que se conecten utilicen el host bastión como proxy. En el siguiente ejemplo, se muestra cómo establecer una conexión de proxy mediante Mac OS. Sustituya *BastionIP* por la dirección IP de la instancia EC2 del host del bastión y *MySshKey.pem* por el archivo de pares de claves que copió en el host del bastión.

En la línea de comando, escriba lo siguiente:

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

Así se configura el reenvío de puertos del puerto 9001 de la máquina local al host bastión.

A continuación, configure su navegador o sistema para que utilice el proxy SOCKS para localhost:9001. Por ejemplo, si usa Mac OS, seleccione System preferences (Preferencias del Sistema), Network (Red), Advanced (Avanzado), seleccione SOCKS proxy y escriba localhost:9001.

Si utilizas la opción FoxyProxy Estándar con Chrome, selecciona Más herramientas y extensiones. En FoxyProxy Estándar, selecciona Detalles, Opciones de extensión y Añadir nuevo proxy. Seleccione Manual Proxy Configuration (Configuración manual del proxy). En Host or IP Address (Host o dirección IP) escriba localhost y en Port (Puerto) escriba 9001. Seleccione SOCKS Proxy?, Save (Guardar).

Ahora debería poder conectarse a las direcciones de host de Hyperledger Fabric enumeradas en la salida de la plantilla.

Historial de revisiones del documento

En la tabla siguiente, se describen los cambios realizados en esta guía.

Última actualización de la documentación: 1 de mayo de 2019

Cambio	Descripción	Fecha
Suspensión de AWS Blockchain Templates.	AWS Blockchain Templates se suspendió el 30 de abril de 2019. No habrá más actualizaciones de este servicio ni de esta documentación complementaria. Para disfrutar de la mejor experiencia de Managed Blockchain en AWS, le recomendamos que utilice Amazon Managed Blockchain (AMB) .	1 de mayo de 2019
Actualizaciones del host bastión.	Modificación del tutorial de introducción y de los requisitos previos de Ethereum para la adición de un host bastión, que permite el acceso a los recursos web servidos a través del equilibrador de carga interno cuando se utiliza la plataforma ECS y la instancia EC2 cuando se utiliza docker-local.	3 de mayo de 2018
Creación de la guía	Nueva guía para desarrolladores de la versión inicial de las AWS Blockchain Templates.	19 de abril de 2018

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.