



Entwicklerhandbuch

AWS Blockchain-Vorlagen



AWS Blockchain-Vorlagen: Entwicklerhandbuch

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

.....	iv
Was sind AWS-Blockchain-Templates?	1
Erste Schritte	2
Ich kenne mich mit Blockchain aus AWS	2
Ich kenne mich mit Blockchain aus AWS und bin neu darin	3
Ich bin ein Anfänger AWS und kenne mich gut mit Blockchain aus	3
Ich bin neu in Sachen AWS Blockchain	3
Verwandte Dienstleistungen	3
Einrichten	5
Anmelden bei AWS	5
Erstellen eines IAM-Benutzers	6
Erstellen eines Schlüsselpaares	8
Erste Schritte	10
Einrichten der Voraussetzungen	11
Erstellen von VPC und Subnetzen	11
Erstellen von Sicherheitsgruppen	15
Erstellen Sie eine IAM-Rolle für Amazon ECS und ein EC2 Instance-Profil	18
Erstellen eines Bastion-Host	24
Erstellen des Ethereum-Netzwerks	25
Connect zum Bastion-Host her EthStats und EthExplorer verwenden Sie ihn	28
Bereinigen von Ressourcen	31
AWS-Blockchain-Templates und -Funktionen	33
AWS-Blockchain-Template für Ethereum	33
Links für den Start	33
Ethereum-Optionen	34
Voraussetzungen	37
Verbindung zu Ethereum-Ressourcen herstellen	46
AWS-Blockchain-Template für Hyperledger Fabric	48
Links für den Start	49
AWS-Blockchain-Template für Hyperledger Fabric-Komponenten	49
Voraussetzungen	50
Verbindung zu Hyperledger Fabric-Ressourcen herstellen	52
Dokumentverlauf	54
AWS Glossar	56

AWS Blockchain Templates wurde am 30. April 2019 eingestellt. Es werden keine weiteren Aktualisierungen dieses Services oder dieser unterstützenden Dokumentation vorgenommen. Für ein optimales Managed Blockchain-Erlebnis empfehlen wir Ihnen AWS, [Amazon Managed Blockchain \(AMB\)](#) zu verwenden. Weitere Informationen zu den ersten Schritten mit Amazon Managed Blockchain finden Sie in unserem [Workshop zu Hyperledger Fabric](#) oder in unserem [Blog zur Bereitstellung eines Ethereum-Knotens](#). Wenn Sie Fragen zu AMB haben oder weitere Unterstützung benötigen, [wenden Sie sich an unser Support Account-Team](#). AWS

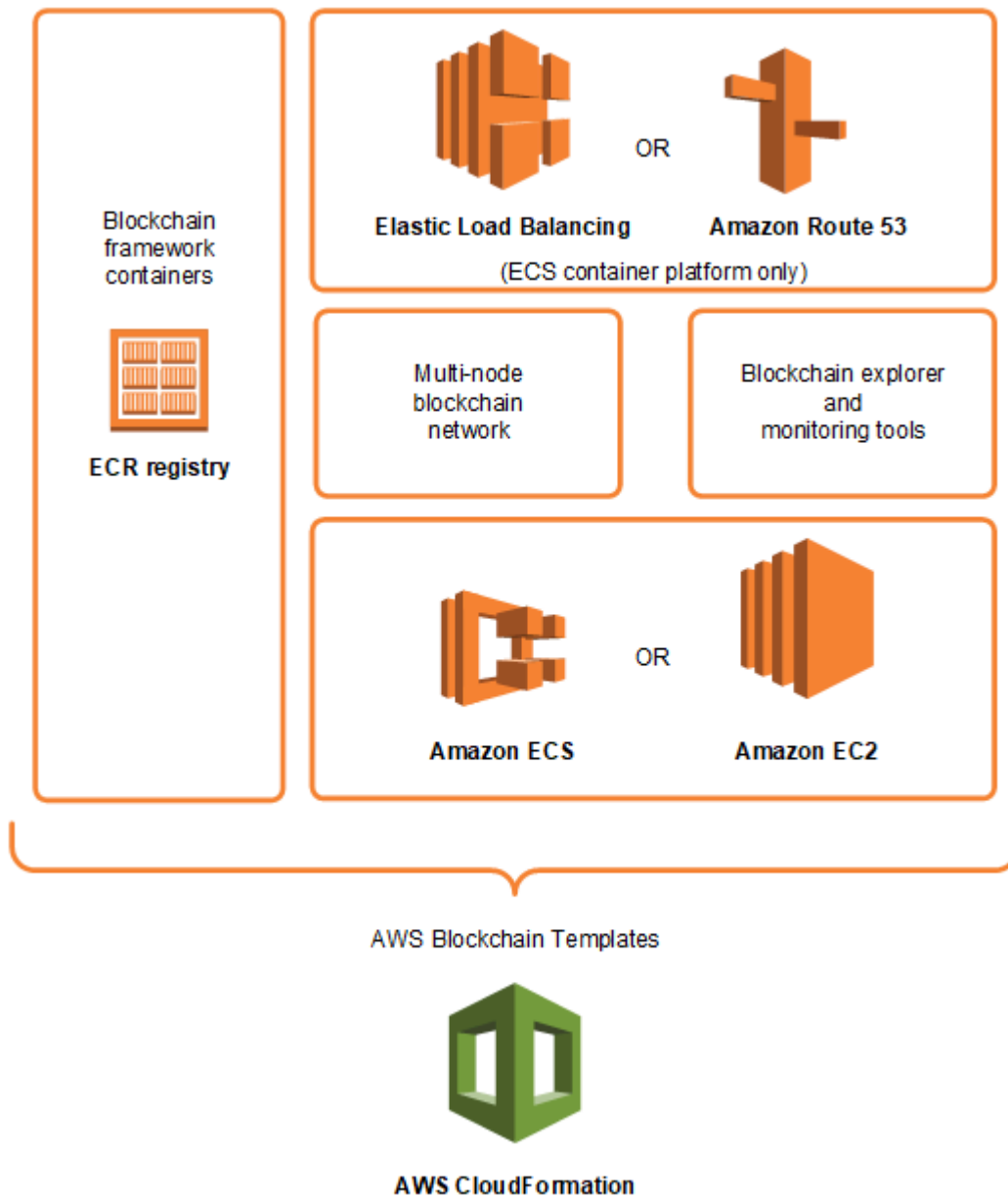
Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.

Was sind AWS-Blockchain-Templates?

Mit AWS Blockchain Templates können Sie AWS mithilfe verschiedener Blockchain-Frameworks schnell Blockchain-Netzwerke erstellen und bereitstellen. Blockchain ist eine dezentrale Datenbanktechnologie, die eine ständig wachsende Anzahl von Transaktionen und intelligenten Verträgen verwaltet, die gegen Manipulationen und Revisionen mittels Kryptographie geschützt sind.

Ein Blockchain-Netzwerk ist ein peer-to-peer Netzwerk, das die Effizienz und Unveränderlichkeit von Transaktionen für Geschäftsprozesse wie internationale Zahlungen, Lieferkettenmanagement, Grundbucheintragung, Crowdfunding, Unternehmensführung, Finanztransaktionen und mehr verbessert. Dies ermöglicht es Personen und Organisationen, die sich möglicherweise nicht kennen, dem Transaktionsprotokoll zu vertrauen und es unabhängig zu überprüfen.

Sie verwenden AWS Blockchain Templates, um CloudFormation Stacks zur Erstellung von Blockchain-Netzwerken zu konfigurieren und zu starten. Die AWS Ressourcen und Services, die Sie verwenden, hängen von der von Ihnen ausgewählten AWS-Blockchain-Template und den von Ihnen angegebenen Optionen ab. Informationen zu den verfügbaren Vorlagen und deren Funktionen finden Sie unter [AWS-Blockchain-Templates und -Funktionen](#). Die grundlegenden Komponenten eines Blockchain-Netzwerks, das mithilfe von AWS-Blockchain-Templates erstellt wurde, sind in der folgenden Abbildung dargestellt.



Erste Schritte

Der beste Ausgangspunkt hängt von Ihrem Kenntnisstand im Bereich Blockchain und AWS insbesondere von den Services im Zusammenhang mit AWS-Blockchain-Templates ab.

Ich kenne mich mit Blockchain aus AWS

Beginnen Sie mit dem Thema in [AWS-Blockchain-Templates und -Funktionen](#) über die Frameworks, die Sie verwenden möchten. Verwenden Sie die Links, um die AWS-Blockchain-Template zu starten

und das Blockchain-Netzwerk zu konfigurieren, oder laden Sie die Vorlagen herunter, um sie selbst auszuprobieren.

Ich kenne mich mit Blockchain aus AWS und bin neu darin

Beginnen Sie mit dem Tutorial [Erste Schritte mit AWS-Blockchain-Templates](#). Dies führt Sie durch die Erstellung eines einführenden Ethereum-Blockchain-Netzwerks mit Standardeinstellungen. Wenn Sie fertig sind, schauen Sie sich für eine Übersicht über Blockchain-Frameworks und Links die Seite [AWS-Blockchain-Templates und -Funktionen](#) an, um mehr über Konfigurationsmöglichkeiten und Funktionen zu erfahren.

Ich bin ein Anfänger AWS und kenne mich gut mit Blockchain aus

Beginnen Sie mit [AWS-Blockchain-Templates einrichten](#). Dies hilft Ihnen dabei, sich mit den Grundlagen wie einem Konto und einem Benutzerprofil vertraut zu machen. AWS Lesen Sie als nächstes das Tutorial [Erste Schritte mit AWS-Blockchain-Templates](#). Dieses Tutorial führt Sie durch die Erstellung eines einführenden Ethereum-Blockchain-Netzwerks. Auch wenn Sie Ethereum letztendlich nicht nutzen werden, erhalten Sie praktische Erfahrungen beim Einrichten verwandter Services. Diese Erfahrung ist für alle Blockchain-Frameworks nützlich. Das Thema finden Sie im Abschnitt [AWS-Blockchain-Templates und -Funktionen](#) zu Ihrem Framework.

Ich bin neu in Sachen AWS Blockchain

Beginnen Sie mit [AWS-Blockchain-Templates einrichten](#). Auf diese Weise können Sie sich mit den Grundlagen wie einem Konto und einem Benutzerprofil vertraut machen. AWS Lesen Sie als nächstes das Tutorial [Erste Schritte mit AWS-Blockchain-Templates](#). Dieses Tutorial führt Sie durch die Erstellung eines einführenden Ethereum-Blockchain-Netzwerks. Nehmen Sie sich Zeit, um die Links zu erkunden, um mehr über AWS Dienste und Ethereum zu erfahren.

Verwandte Dienstleistungen

Abhängig von den ausgewählten Optionen können AWS Blockchain Templates die folgenden AWS Services für die Bereitstellung von Blockchain verwenden:

- Amazon EC2 — Stellt Rechenkapazität für Ihr Blockchain-Netzwerk bereit. Weitere Informationen finden Sie im [EC2 Amazon-Benutzerhandbuch](#).
- Amazon ECS — Orchestriert die Container-Bereitstellung zwischen EC2 Instances in einem Cluster für Ihr Blockchain-Netzwerk, falls Sie es verwenden möchten. Weitere Informationen finden Sie im [Amazon Elastic Container Service-Entwicklerhandbuch](#).

- Amazon VPC — Bietet Netzwerkzugriff für die von Ihnen erstellten Ethereum-Ressourcen. Sie können die Konfiguration in puncto Zugänglichkeit und Sicherheit anpassen. Weitere Informationen finden Sie im [Amazon VPC Developer Guide](#).
- Application Load Balancing — Dient als zentrale Anlaufstelle für den Zugriff auf verfügbare Benutzeroberflächen und die interne Serviceerkennung, wenn Amazon ECS als Containerplattform verwendet wird. Weitere Informationen finden Sie unter [Was ist ein Application Load Balancer?](#) im Benutzerhandbuch für Application Load Balancers. .

AWS-Blockchain-Templates einrichten

Bevor Sie mit AWS Blockchain Templates beginnen, führen Sie die folgenden Aufgaben durch:

- [Anmelden bei AWS](#)
- [Erstellen eines IAM-Benutzers](#)
- [Erstellen eines Schlüsselpaares](#)

Dies sind grundlegende Voraussetzungen für alle Blockchain-Konfigurationen. Darüber hinaus können für das von Ihnen gewählte Blockchain-Netzwerk Voraussetzungen gegeben sein, die je nach gewünschter Umgebung und Konfiguration variieren. Weitere Informationen finden Sie im entsprechenden Abschnitt Ihrer Blockchain-Vorlage unter [AWS-Blockchain-Templates und -Funktionen](#).

step-by-step Anweisungen zur Einrichtung der Voraussetzungen für ein privates Ethereum-Netzwerk mithilfe eines Amazon ECS-Clusters finden Sie unter [Erste Schritte mit AWS-Blockchain-Templates](#).

Anmelden bei AWS

Wenn Sie sich für registrieren AWS, wird Ihr AWS Konto automatisch für alle Dienste registriert. Berechnet werden Ihnen aber nur die Services, die Sie nutzen.

Wenn Sie bereits ein AWS Konto haben, fahren Sie mit der nächsten Aufgabe fort. Wenn Sie kein AWS -Konto haben, führen Sie die folgenden Schritte zum Erstellen eines Kontos aus.

Um ein AWS Konto zu erstellen

1. Öffnen Sie <https://portal.aws.amazon.com/billing/die-Anmeldung>.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

Notieren Sie sich Ihre AWS Kontonummer. Sie benötigen sie, wenn Sie in der nächsten Aufgabe einen IAM-Benutzer erstellen.

Erstellen eines IAM-Benutzers

Dienste AWS erfordern, dass Sie beim Zugriff auf sie Anmeldeinformationen angeben, damit der Dienst feststellen kann, ob Sie über Berechtigungen für den Zugriff auf seine Ressourcen verfügen. Für die Konsole müssen Sie Ihr Passwort eingeben. Sie können Zugriffsschlüssel für Ihr AWS Konto erstellen, um auf die Befehlszeilenschnittstelle oder API zuzugreifen. Wir empfehlen jedoch nicht, dass Sie für den Zugriff die Anmeldeinformationen Ihres AWS Kontos AWS verwenden. Stattdessen empfehlen wir Ihnen, AWS Identity and Access Management (IAM) zu verwenden. Erstellen Sie einen IAM-Benutzer und fügen Sie den Benutzer einer IAM-Gruppe mit Administrator-Berechtigungen hinzu oder gewähren Sie diesem Benutzer Administrator-Berechtigungen. Sie können dann AWS mit einer speziellen URL und den Anmeldeinformationen für den IAM-Benutzer darauf zugreifen.

Wenn Sie sich für einen IAM-Benutzer registriert haben, AWS aber noch keinen für sich selbst erstellt haben, können Sie einen mit der IAM-Konsole erstellen. Wenn Sie bereits einen IAM-Benutzer haben, können Sie diesen Schritt überspringen.

Wählen Sie zum Erstellen eines Administratorbenutzers eine der folgenden Optionen aus.

Wählen Sie eine Möglichkeit zur Verwaltung Ihres Administrators aus.	Bis	Von	Sie können auch
Im IAM Identity Center (Empfohlen)	Verwendung von kurzfristigen Anmeldeinformationen für den Zugriff auf AWS.	Beachtung der Anweisungen unter Erste Schritte im AWS IAM Identity Center - Benutzerhandbuch.	Konfigurieren Sie den programmatischen Zugriff, indem Sie die Konfiguration AWS CLI für die Verwendung AWS IAM Identity Center im AWS Command Line

Wählen Sie eine Möglichkeit zur Verwaltung Ihres Administrators aus.	Bis	Von	Sie können auch
	Dies steht im Einklang mit den bewährten Methoden für die Sicherheit. Weitere Informationen zu bewährten Methoden finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.		Interface Benutzerhandbuch vornehmen.
In IAM (Nicht empfohlen)	Verwendung von langfristigen Anmeldeinformationen für den Zugriff auf AWS.	Folgen Sie den Anleitungen unter IAM-Benutzer für den Notfallzugriff erstellen im IAM-Benutzerhandbuch.	Sie konfigurieren den programmgesteuerten Zugriff unter Verwendung der Informationen unter Verwalten der Zugriffsschlüssel für IAM-Benutzer im IAM-Benutzerhandbuch.

Um sich als dieser neue IAM-Benutzer anzumelden, melden Sie sich von der ab und verwenden Sie dann die folgende URL, wobei `your_aws_account_id` Ihre AWS Kontonummer ohne Bindestriche ist (wenn Ihre Kontonummer beispielsweise lautet, lautet Ihre AWS Konto-ID): `AWS-Managementkonsole1234-5678-9012` `AWS 123456789012`

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

Geben Sie den IAM-Benutzernamen und das von Ihnen soeben erstellte Passwort ein. Nachdem Sie sich angemeldet haben, wird in der Navigationsleiste „your_user_name @ your_aws_account_id“ angezeigt.

Wenn Sie nicht möchten, dass die URL für Ihre Anmeldeseite Ihre Konto-ID enthält, können Sie AWS einen Kontoalias erstellen. Wählen Sie im IAM-Dashboard die Option Kontoalias erstellen und geben Sie einen Alias ein, z. B. Ihren Firmennamen. Nach dem Erstellen eines Konto-Alias verwenden Sie die folgende URL, um sich anzumelden:

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

Um den Anmeldelink der IAM-Benutzer Ihres Kontos zu verifizieren, öffnen Sie die IAM-Konsole und prüfen dies im Dashboard unter IAM users sign-in link.

Weitere Informationen finden Sie im [AWS Identity and Access Management-Benutzerhandbuch](#).

Erstellen eines Schlüsselpaares

AWS verwendet Public-Key-Kryptografie, um die Anmeldeinformationen für die Instanzen in einem Blockchain-Netzwerk zu sichern. Sie geben den Namen des key pair an, wenn Sie jede AWS-Blockchain-Template verwenden. Über das Schlüsselpaar können Sie dann direkt auf Instances zugreifen, z. B. um sich über SSH anzumelden.

Wenn Sie bereits ein Schlüsselpaar in der richtigen Region haben, können Sie diesen Schritt überspringen. Wenn Sie noch kein key pair erstellt haben, können Sie eines mit der EC2 Amazon-Konsole erstellen. Erstellen Sie das Schlüsselpaar in derselben Region, die Sie zum Start des Ethereum-Netzwerks verwenden. Weitere Informationen finden Sie unter [Regionen und Availability Zones](#) im EC2 Amazon-Benutzerhandbuch.

So erstellen Sie ein Schlüsselpaar

1. Öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie in der Navigationsleiste eine Region für das Schlüsselpaar aus. Sie können eine beliebige Region auswählen, die Ihnen zur Verfügung steht, und zwar unabhängig von Ihrem Standort. Schlüsselpaare sind jedoch spezifisch für eine Region. Wenn Sie beispielsweise planen, eine Instance in der Region USA Ost (Ohio) zu starten, müssen Sie ein key pair für die Instance in derselben Region erstellen.
3. Wählen Sie im Navigationsbereich Key Pairs (Schlüsselpaare) und Create Key Pair (Schlüsselpaar erstellen) aus.

4. Geben Sie unter Key pair name (Schlüsselpaarname) einen Namen für das neue Schlüsselpaar ein. Wählen Sie einen Namen, den Sie sich leicht merken können, z. B. Ihren IAM-Benutzernamen, gefolgt von `-key-pair`, plus dem Namen der Region. Beispiel: `me-key-pair-useast2`. Wählen Sie Erstellen aus.
5. Die private Schlüsseldatei wird von Ihrem Browser automatisch heruntergeladen. Der Basisdateiname ist der Name, den Sie als Namen für Ihr Schlüsselpaar festgelegt haben, und die Dateinamenerweiterung lautet `.pem`. Speichern Sie die Datei mit dem privaten Schlüssel an einem sicheren Ort.

 **Important**

Dies ist die einzige Möglichkeit, die private Schlüsseldatei zu speichern. Sie geben den Namen für Ihr Schlüsselpaar beim Starten des Ethereum-Netzwerks an.

Weitere Informationen finden Sie unter [Amazon EC2 Key Pairs](#) im EC2 Amazon-Benutzerhandbuch. Weitere Informationen zum Herstellen einer Verbindung zu EC2 Instances mithilfe des key pair finden Sie unter [Connect to Your Linux Instance](#) im EC2 Amazon-Benutzerhandbuch.

Erste Schritte mit AWS-Blockchain-Templates

Dieses Tutorial zeigt, wie Sie die AWS-Blockchain-Template für Ethereum verwenden, um ein privates Blockchain-Netzwerk auf AWS Through zu erstellen CloudFormation. Das Netzwerk, das Sie erstellen, hat zwei Ethereum-Clients und einen Miner, die auf EC2 Amazon-Instances in einem Amazon ECS-Cluster laufen. Amazon ECS führt diese Dienste in Docker-Containern aus, die aus Amazon ECR abgerufen wurden. Bevor Sie mit diesem Tutorial beginnen, ist es hilfreich, etwas über Blockchain-Netzwerke und die beteiligten AWS Dienste zu wissen, aber nicht erforderlich.

In diesem Tutorial wird davon ausgegangen, dass Sie die allgemeinen vorbereitenden Schritte, die unter [AWS-Blockchain-Templates einrichten](#) aufgeführt sind, ausgeführt haben. Darüber hinaus müssen Sie einige AWS Ressourcen einrichten, z. B. ein Amazon VPC-Netzwerk und spezifische Berechtigungen für IAM-Rollen, bevor Sie die Vorlage verwenden können.

In dem Tutorial wird gezeigt, wie man diese vorbereitenden Schritte durchführt. Wir haben Einrichtungsentscheidungen getroffen, diese sind aber nicht bindend. Solange Sie die Voraussetzungen erfüllen, können Sie je nach den Anforderungen Ihrer Anwendung und Umgebung andere Entscheidungen hinsichtlich der Konfiguration treffen. Informationen zu den Funktionen und allgemeinen Voraussetzungen für jede Vorlage sowie zum Herunterladen von Vorlagen oder zum direkten Aufruf in CloudFormation finden Sie unter [AWS-Blockchain-Templates und -Funktionen](#).

In diesem Tutorial verwenden Beispiele die Region USA West (Oregon) (us-west-2), aber Sie können jede Region verwenden, die AWS-Blockchain-Templates unterstützt:

- Region USA West (Oregon) (us-west-2)
- Region USA Ost (Nord-Virginia) (us-east-1)
- Region USA Ost (Ohio) (us-east-2)

Note

Wenn Sie eine Vorlage in einer Region ausführen, die oben nicht aufgeführt ist, werden Ressourcen in der Region USA Ost (Nord-Virginia) (us-east-1) gestartet.

Die AWS-Blockchain-Template für Ethereum, die Sie mithilfe dieses Tutorials konfigurieren, erstellt die folgenden Ressourcen:

- EC2 On-Demand-Instances des von Ihnen angegebenen Typs und der Anzahl. Das Tutorial verwendet den Instance-Standardtyp t2.medium.
- Ein interner Application Load Balancer.

Im Anschluss an das Tutorial werden Schritte zur Bereinigung der von Ihnen angelegten Ressourcen angeboten.

Themen

- [Einrichten der Voraussetzungen](#)
- [Erstellen des Ethereum-Netzwerks](#)
- [Connect zum Bastion-Host her EthStats und EthExplorer verwenden Sie ihn](#)
- [Bereinigen von Ressourcen](#)

Einrichten der Voraussetzungen

Für die Konfiguration von AWS Blockchain Template for Ethereum, die Sie in diesem Tutorial angeben, müssen Sie Folgendes tun:

- [Erstellen von VPC und Subnetzen](#)
- [Erstellen von Sicherheitsgruppen](#)
- [Erstellen Sie eine IAM-Rolle für Amazon ECS und ein EC2 Instance-Profil](#)
- [Erstellen eines Bastion-Host](#)

Erstellen von VPC und Subnetzen

Die AWS-Blockchain-Template für Ethereum startet Ressourcen in einem virtuellen Netzwerk, das Sie mithilfe von Amazon Virtual Private Cloud (Amazon VPC) definieren. Mit der in diesem Tutorial angegebenen Konfiguration wird ein Application Load Balancer erstellt, der zwei öffentliche Subnetze in verschiedenen Availability Zones benötigt. Darüber hinaus ist ein privates Subnetz für die Container-Instances erforderlich, und das Subnetz muss sich in der gleichen Availability Zone wie der Application Load Balancer befinden. Sie verwenden zunächst den VPC-Assistenten, um ein öffentliches und ein privates Subnetz in derselben Availability Zone zu erstellen. Anschließend erstellen Sie ein zweites öffentliches Subnetz innerhalb dieser VPC in einer anderen Availability Zone.

Weitere Informationen finden Sie unter [Was ist Amazon VPC?](#) im Amazon VPC-Benutzerhandbuch.

Verwenden Sie die Amazon VPC-Konsole (<https://console.aws.amazon.com/vpc/>), um die Elastic IP-Adresse, die VPC und das Subnetz wie unten beschrieben zu erstellen.

Erstellen einer Elastic IP-Adresse

1. Öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie Elastic IPs, Neue Adresse zuweisen, Allocate aus.
3. Notieren Sie sich die Elastic IP-Adresse und wählen Sie dann Close (Schließen).
4. Suchen Sie in der Liste der Elastic IP-Adressen die Allocation ID (Zuordnungs-ID) für die zuvor erstellte Elastic IP-Adresse. Diese verwenden Sie beim Erstellen der VPC.

So erstellen Sie die VPC

1. Wählen Sie in der Navigationsleiste eine Region für die VPC aus. VPCs sind spezifisch für eine Region, wählen Sie also dieselbe Region aus, in der Sie Ihr key pair erstellt haben und in der Sie den Ethereum-Stack starten. Weitere Informationen finden Sie unter [Erstellen eines Schlüsselpaares](#).
2. Wählen Sie im VPC-Dashboard Start VPC Wizard aus.
3. Wählen Sie auf der Seite Step 1: Select a VPC Configuration (Schritt 1: Wählen Sie eine VPC-Konfiguration aus) die Option VPC with Public and Private Subnets (VPC mit öffentlichen und privaten Subnetzen), Select (Auswählen).
4. Belassen Sie auf der Seite Schritt 2: VPC mit öffentlichen und privaten Subnetzen die Standardwerte für IPv4 CIDR-Block und IPv6 CIDR-Block. Geben Sie unter VPC name (VPC-Name) einen Anzeigenamen ein.
5. Behalten Sie für IPv4 CIDR des öffentlichen Subnetzes den Standardwert bei. Wählen Sie für Availability Zone eine Zone aus. Geben Sie für Public subnet name (Öffentlicher Subnetz-Namen) einen Anzeigenamen ein.

Sie geben dieses Subnetz als eines der ersten beiden Subnetze für den Application Load Balancer an, wenn Sie die Vorlage verwenden.

Notieren Sie sich die Availability Zone dieses Subnetzes, da Sie für das private Subnetz die gleiche Availability Zone und für das andere öffentliche Subnetz eine andere auswählen.

6. Behalten Sie für IPv4 CIDR des privaten Subnetzes den Standardwert bei. Wählen Sie unter Availability Zone, dieselbe Availability Zone wie im vorherigen Schritt aus. Geben Sie für Private subnet name (Privater Subnetz-Namen) einen Anzeigenamen ein.
7. Wählen Sie unter Elastic IP Allocation ID (Elastische IP-Zuordnungs-ID) die von Ihnen zuvor erstellte Elastic IP-Adresse aus.
8. Behalten Sie die Standardwerte für die anderen Einstellungen bei.
9. Wählen Sie VPC erstellen aus.

Das folgende Beispiel zeigt eine EthereumNetworkVPC-VPC mit einem öffentlichen Subnetz EthereumPubSub1 und einem privaten Subnetz 1. EthereumPvtSub Das öffentliche Subnetz verwendet die Availability Zone 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

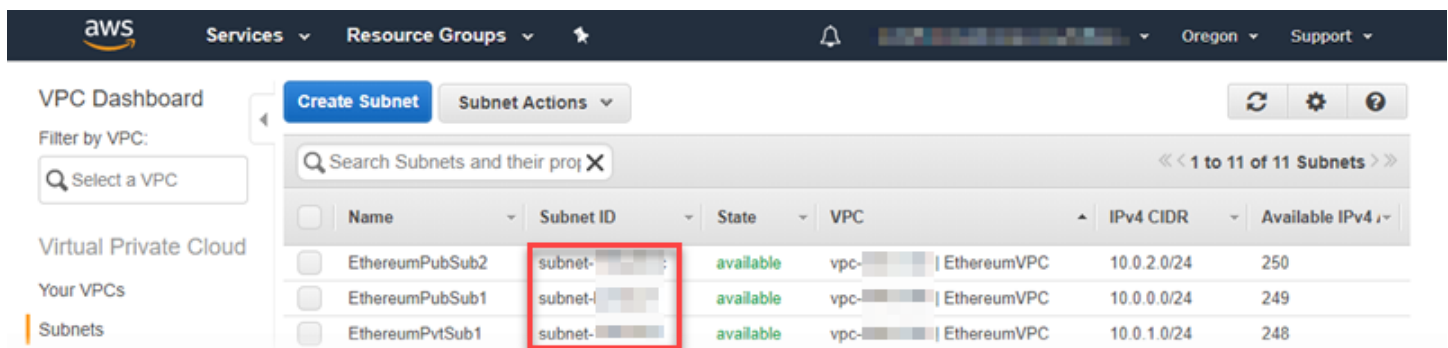
Hardware tenancy:* ▼

So erstellen Sie das zweite öffentliche Subnetz in einer anderen Availability Zone

1. Wählen Sie die Option Subnets (Subnetze) und dann das von Ihnen zuvor erstellte öffentliche Subnetz aus der Liste aus. Klicken Sie auf die Registerkarte Route Table (Routing-Tabelle) und notieren Sie sich die ID der Route table (Routing-Tabelle). Dieselbe Routing-Tabelle geben Sie unten für das zweite öffentliche Subnetz an.

2. Wählen Sie Create Subnet aus.
3. Geben Sie unter Name tag (Namensbezeichnung) einen Namen für das Subnetz ein. Sie verwenden diesen Namen später, wenn Sie den Bastions-Host in diesem Netzwerk erstellen.
4. Wählen Sie als VPC die zuvor erstellte VPC aus.
5. Wählen Sie als Availability Zone eine andere Zone als die Zone aus, die Sie für das erste öffentliche Subnetz ausgewählt haben.
6. Geben Sie für IPv4 CIDR-Block 10.0.2.0/24 ein.
7. Wählen Sie Ja, erstellen aus. Das Subnetz wird der Liste der Subnetze hinzugefügt.
8. Wenn das Subnetz aus der Liste ausgewählt ist, wählen Sie Subnet Actions (Subnetz-Aktionen), Modify auto-assign IP settings (Einstellungen für automatische IP-Zuweisung ändern). Wählen Sie Automatisch IPs zuweisen, Speichern, Schließen aus. Dadurch kann der Bastions-Host eine öffentliche IP-Adresse abrufen, wenn Sie sie in diesem Subnetz erstellen.
9. Klicken Sie auf der Registerkarte Route Table (Routing-Tabelle) auf Edit (Bearbeiten). Unter Change to (Ändern in) wählen Sie die zuvor notierte ID der Routing-Tabelle aus und klicken dann auf Save (Speichern).

Sie sollten jetzt drei Subnetze für die VPC sehen, die Sie zuvor erstellt haben. Notieren Sie sich die Subnetznamen, IDs damit Sie sie mithilfe der Vorlage angeben können.



The screenshot shows the AWS VPC Dashboard with a table of subnets. The table has columns for Name, Subnet ID, State, VPC, IPv4 CIDR, and Available IPv4. Three subnets are listed, and their Subnet IDs are highlighted with a red box.

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

Erstellen von Sicherheitsgruppen

Sicherheitsgruppen fungieren als Firewall und steuern den bei den Ressourcen ein- und ausgehenden Datenverkehr. Wenn Sie die Vorlage verwenden, um ein Ethereum-Netzwerk auf einem Amazon ECS-Cluster zu erstellen, geben Sie zwei Sicherheitsgruppen an:

- Eine Sicherheitsgruppe für EC2 Instances, die den Verkehr zu und von EC2 Instances im Cluster steuert

- Eine Sicherheitsgruppe für den Application Load Balancer, die den Verkehr zwischen dem Application Load Balancer, EC2 Instances und dem Bastion-Host steuert. Sie ordnen diese Sicherheitsgruppe auch dem Bastion-Host zu.

Jede Sicherheitsgruppe hat Regeln, die die Kommunikation zwischen dem Application Load Balancer und den EC2 Instances ermöglichen, sowie andere Mindestregeln. Dies setzt voraus, dass die Sicherheitsgruppen gegenseitig aufeinander verweisen. Aus diesem Grund erstellen Sie zunächst die Sicherheitsgruppen und aktualisieren diese dann mit entsprechenden Regeln.

So erstellen Sie zwei Sicherheitsgruppen

1. Öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Security Groups und dann auf Create Security Group.
3. Geben Sie unter Sicherheitsgruppenname einen Namen für die Sicherheitsgruppe ein, der leicht zu identifizieren ist und sie von den anderen unterscheidet, z. B. Ethereum EC2 -SG oder EthereumALB-SG. Sie werden diese Namen in einem späteren Schritt noch verwenden. Geben Sie unter Description (Beschreibung) eine kurze Zusammenfassung ein.
4. Wählen Sie als VPC die zuvor erstellte VPC aus.
5. Wählen Sie Erstellen aus.
6. Wiederholen Sie die obigen Schritte, um die andere Sicherheitsgruppe zu erstellen.

Fügen Sie der Sicherheitsgruppe für Instances Regeln für eingehenden Datenverkehr hinzu EC2

1. Wählen Sie die Sicherheitsgruppe für EC2 Instances aus, die Sie zuvor erstellt haben
2. Klicken Sie auf die Registerkarte Inbound und wählen Sie Edit aus.
3. Wählen Sie für Type (Typ) die Option All traffic (Gesamter Datenverkehr) aus. Lassen Sie für Quelle die Option Benutzerdefiniert ausgewählt und wählen Sie dann die Sicherheitsgruppe, die Sie gerade bearbeiten, aus der Liste aus, zum Beispiel Ethereum EC2 -SG. Dadurch können die EC2 Instances in der Sicherheitsgruppe miteinander kommunizieren.
4. Klicken Sie auf Add Rule (Regel hinzufügen).
5. Wählen Sie für Type (Typ) die Option All traffic (Gesamter Datenverkehr) aus. Lassen Sie bei Source (Quelle) die Option Custom (Benutzerdefiniert) ausgewählt, und wählen Sie anschließend die Sicherheitsgruppe für den Application Load Balancer aus der Liste aus, z. B. EthereumALB-SG. Dadurch können die EC2 Instances in der Sicherheitsgruppe mit dem Application Load Balancer kommunizieren.

6. Wählen Sie Speichern.

Fügen Sie der Sicherheitsgruppe für den Application Load Balancer Regeln für eingehenden Datenverkehr hinzu und bearbeiten Sie die Regeln für ausgehenden Datenverkehr

1. Wählen Sie die Sicherheitsgruppe für Application Load Balancer aus, die Sie zuvor erstellt haben
2. Wählen Sie auf der Registerkarte Inbound Rules (Regeln für eingehenden Datenverkehr) die Option Edit (Bearbeiten) aus und fügen Sie die folgenden Regeln für eingehenden Datenverkehr hinzu:
 - a. Wählen Sie für Type (Typ) die Option All traffic (Gesamter Datenverkehr) aus. Lassen Sie bei Source (Quelle) die Option Custom (Benutzerdefiniert) ausgewählt, und wählen Sie anschließend die Sicherheitsgruppe aus der Liste aus, die Sie gerade bearbeiten, z. B. EthereumALB-SG. Dadurch kann der Application Load Balancer mit sich selbst und mit dem Bastion-Host kommunizieren.
 - b. Klicken Sie auf Add Rule (Regel hinzufügen).
 - c. Wählen Sie für Type (Typ) die Option All traffic (Gesamter Datenverkehr) aus. Lassen Sie für Source die Option Benutzerdefiniert ausgewählt und wählen Sie dann die Sicherheitsgruppe für EC2 Instances aus der Liste aus, zum Beispiel Ethereum EC2 -SG. Dadurch können die EC2 Instances in der Sicherheitsgruppe mit dem Application Load Balancer und dem Bastion-Host kommunizieren.
 - d. Klicken Sie auf Add Rule (Regel hinzufügen).
 - e. Wählen Sie unter Typ die Option SSH aus. Als Source (Quelle) wählen Sie My IP (Meine IP), wodurch das IP CIDR Ihres Computers erkannt und eingegeben wird.

Important

Diese Regel ermöglicht es dem Bastion-Host, SSH-Verkehr von Ihrem Computer zu akzeptieren, sodass Ihr Computer den Bastion-Host verwenden kann, um Webschnittstellen anzuzeigen und eine Verbindung zu EC2 Instanzen im Ethereum-Netzwerk herzustellen. Um anderen zu ermöglichen, sich mit dem Ethereum-Netzwerk zu verbinden, fügen Sie sie als Quellen zu dieser Regel hinzu. Lassen Sie nur eingehenden Datenverkehr zu vertrauenswürdigen Quellen zu.

- f. Wählen Sie Speichern.

3. Wählen Sie auf der Registerkarte Outbound (Regeln für ausgehenden Datenverkehr) die Option Edit (Bearbeiten) und löschen Sie die automatisch erstellte Regel, um ausgehenden Datenverkehr zu allen IP-Adressen zuzulassen.
4. Klicken Sie auf Add Rule (Regel hinzufügen).
5. Wählen Sie für Type (Typ) die Option All traffic (Gesamter Datenverkehr) aus. Lassen Sie für Destination die Option Benutzerdefiniert ausgewählt und wählen Sie dann die Sicherheitsgruppe für EC2 Instances aus der Liste aus. Dies ermöglicht ausgehende Verbindungen vom Application Load Balancer und dem Bastion-Host zu EC2 Instances im Ethereum-Netzwerk.
6. Klicken Sie auf Add Rule (Regel hinzufügen).
7. Wählen Sie für Type (Typ) die Option All traffic (Gesamter Datenverkehr) aus. Lassen Sie bei Destination (Ziel) die Option Custom (Benutzerdefiniert) ausgewählt, und wählen Sie anschließend die Sicherheitsgruppe aus der Liste aus, die Sie gerade bearbeiten, z. B. EthereumALB-SG. Dadurch kann der Application Load Balancer mit sich selbst und mit dem Bastion-Host kommunizieren.
8. Wählen Sie Speichern.

Erstellen Sie eine IAM-Rolle für Amazon ECS und ein EC2 Instance-Profil

Wenn Sie diese Vorlage verwenden, geben Sie eine IAM-Rolle für Amazon ECS und ein EC2 Instance-Profil an. Die diesen Rollen zugeordneten Berechtigungsrichtlinien gestatten den AWS-Ressourcen und Instances in Ihrem Cluster, mit anderen AWS-Ressourcen zu interagieren. Weitere Informationen finden Sie unter [IAM-Rollen](#) im IAM-Benutzerhandbuch. Sie richten die IAM-Rolle für Amazon ECS und das EC2 Instance-Profil mithilfe der IAM-Konsole (<https://console.aws.amazon.com/iam/>) ein.

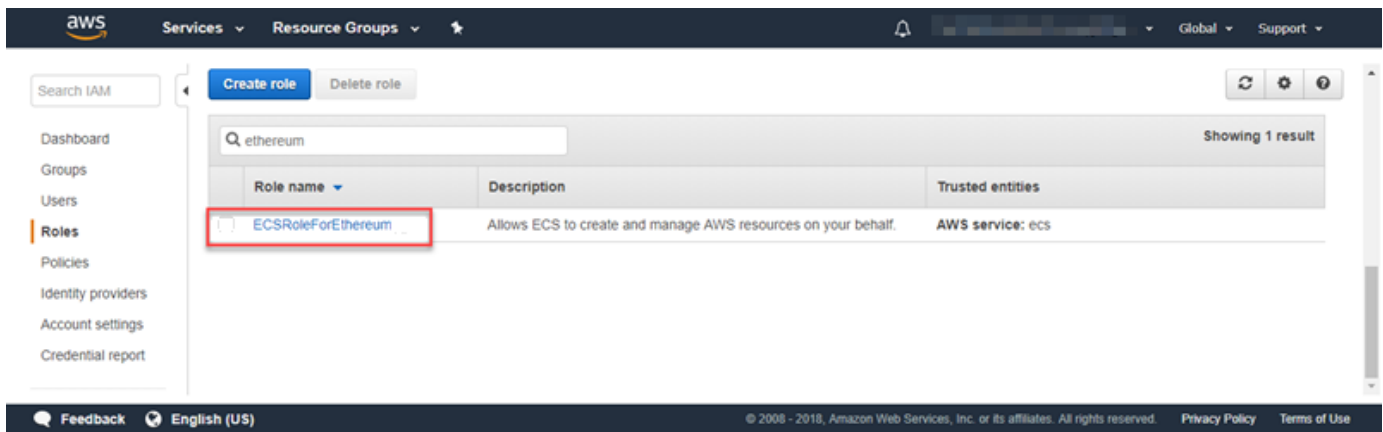
So erstellen Sie die IAM-Rolle für Amazon ECS

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und Create Role (Rolle erstellen) aus.
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS service (AWS-Service) aus.
4. Wählen Sie für Choose the service that will use this role (Service auswählen, der diese Rolle verwendet) die Option Elastic Container Service aus.

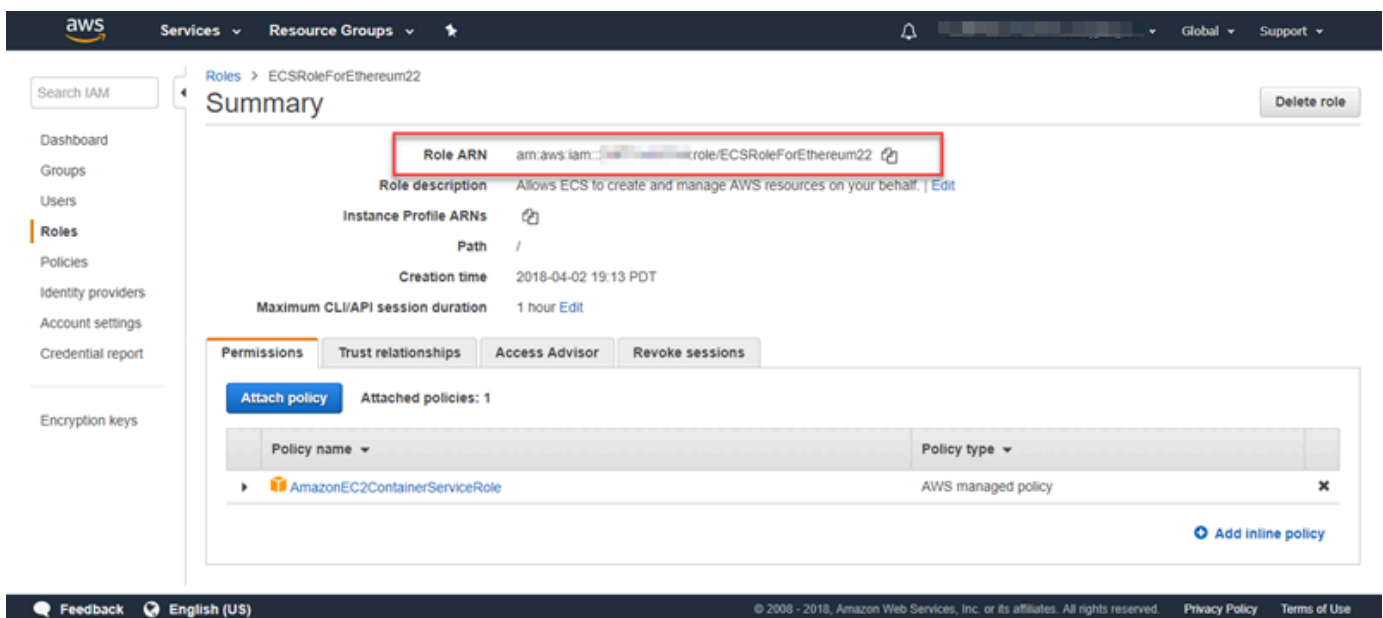
- Wählen Sie unter Select your use case (Auswahl Ihres Anwendungsfalls) die Optionen Elastic Container Service (Elastischer Container-Service), Next:Permissions (Weiter: Berechtigungen) aus.

The screenshot shows the AWS IAM console 'Create role' page. The 'Select type of trusted entity' section has 'AWS service' selected. Below this, a grid of services is shown, with 'Elastic Container Service' highlighted in a red box. In the 'Select your use case' section, the 'Elastic Container Service' option is also highlighted in a red box. The 'Next: Permissions' button is visible at the bottom right.

- Lassen Sie für die Berechtigungsrichtlinie die Standardrichtlinie (Amazon EC2 ContainerServiceRole) ausgewählt und wählen Sie Weiter: Überprüfen.
- Geben Sie unter Rollename einen Wert ein, anhand dessen Sie die Rolle leichter identifizieren können, z. B. ECSRoleForEthereum. Geben Sie unter Role Description (Rollenbeschreibung) eine kurze Zusammenfassung ein. Notieren Sie sich den Rollennamen für später.
- Wählen Sie Rolle erstellen aus.
- Wählen Sie die gerade erstellte Rolle aus der Liste aus. Wenn Ihr Konto viele Rollen hat, können Sie nach dem Rollennamen suchen.



10. Kopieren Sie den Wert Role ARN (Rollen-ARN) und speichern Sie ihn, sodass Sie ihn jederzeit erneut kopieren können. Sie benötigen diesen ARN beim Erstellen des Ethereum-Netzwerks.



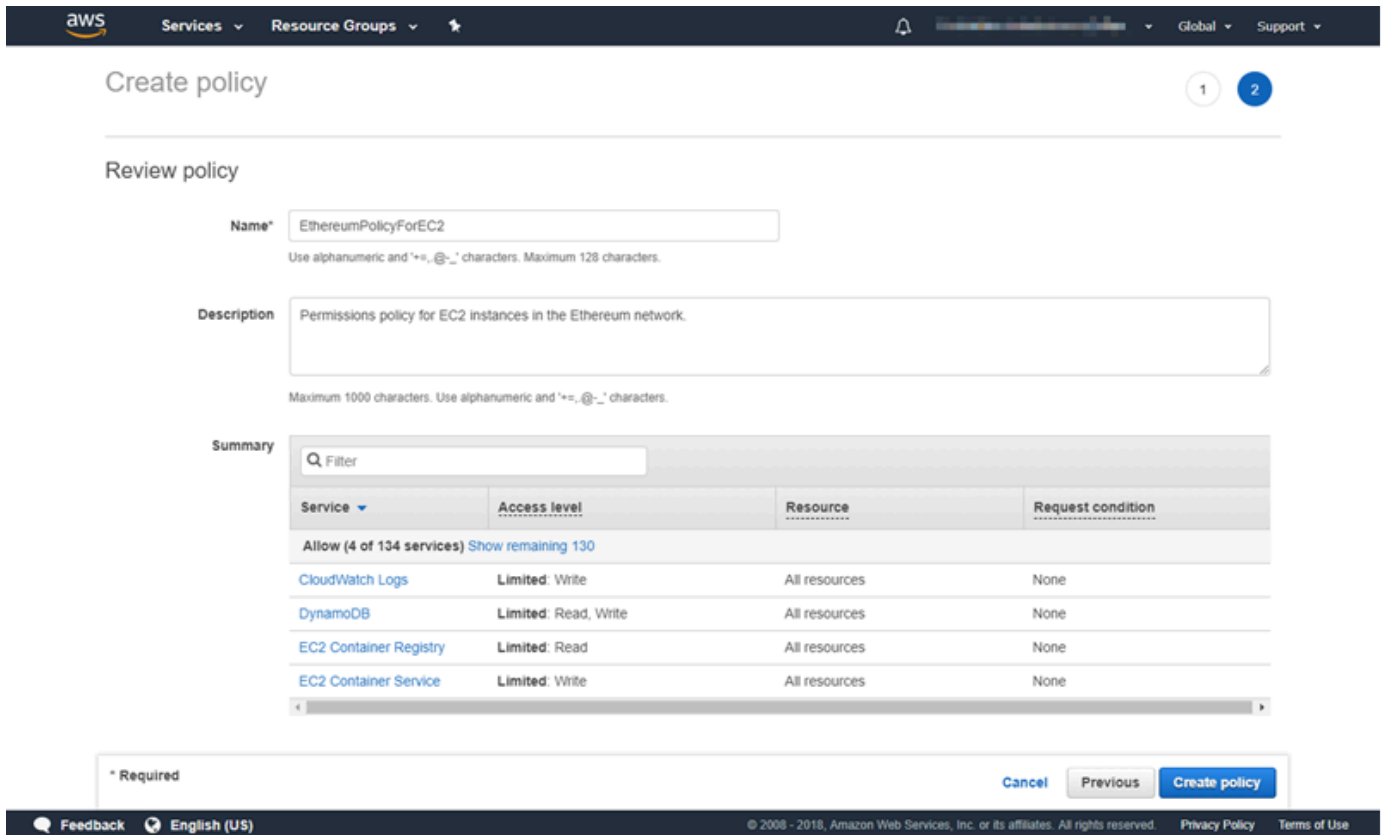
Das EC2 Instanzprofil, das Sie in der Vorlage angeben, wird von EC2 Instanzen im Ethereum-Netzwerk für die Interaktion mit anderen AWS Diensten verwendet. Sie erstellen eine Berechtigungsrichtlinie für die Rolle, erstellen die Rolle (die automatisch ein gleichnamiges Instance-Profil erstellt) und fügen dann die Berechtigungsrichtlinie an die Rolle an.

Um ein EC2 Instanzprofil zu erstellen

1. Wählen Sie im Navigationsbereich Policies und Create policy aus.
2. Wählen Sie JSON und ersetzen Sie die Standard-Richtlinienanweisung durch die folgende JSON-Richtlinie:

```
{
  "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. Wählen Sie Richtlinie prüfen.
4. Geben Sie unter Name einen Wert ein, anhand dessen Sie beispielsweise diese Berechtigungsrichtlinie identifizieren können EthereumPolicyForEC2. Geben Sie unter Description (Beschreibung) eine kurze Zusammenfassung ein. Wählen Sie Richtlinie erstellen aus.



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

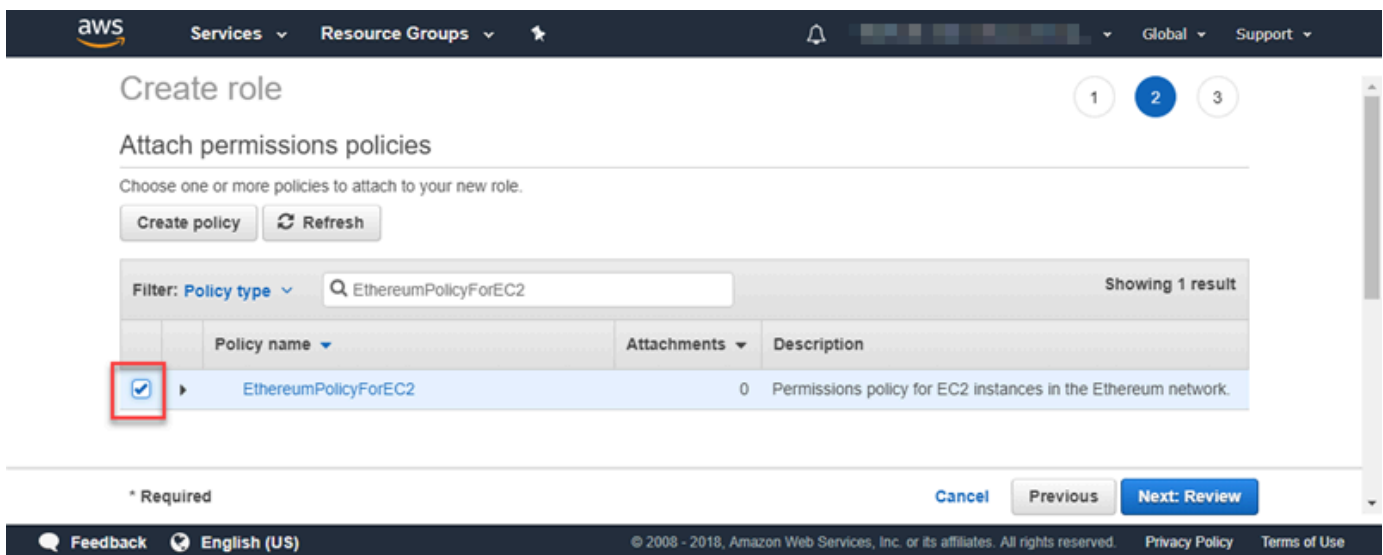
Filter

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. Wählen Sie Roles (Rollen), Create role (Rolle erstellen) aus.
6. Wählen EC2 Sie Weiter: Berechtigungen.
7. Geben Sie im Suchfeld beispielsweise den Namen der Berechtigungsrichtlinie ein, die Sie zuvor erstellt haben EthereumPolicyForEC2.
8. Wählen Sie das Häkchen für die Richtlinie, die Sie zuvor erstellt haben, und wählen Sie Next: Review (Weiter: Überprüfen).



Create role 1 2 3

Attach permissions policies

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

[Create policy](#) [Refresh](#)

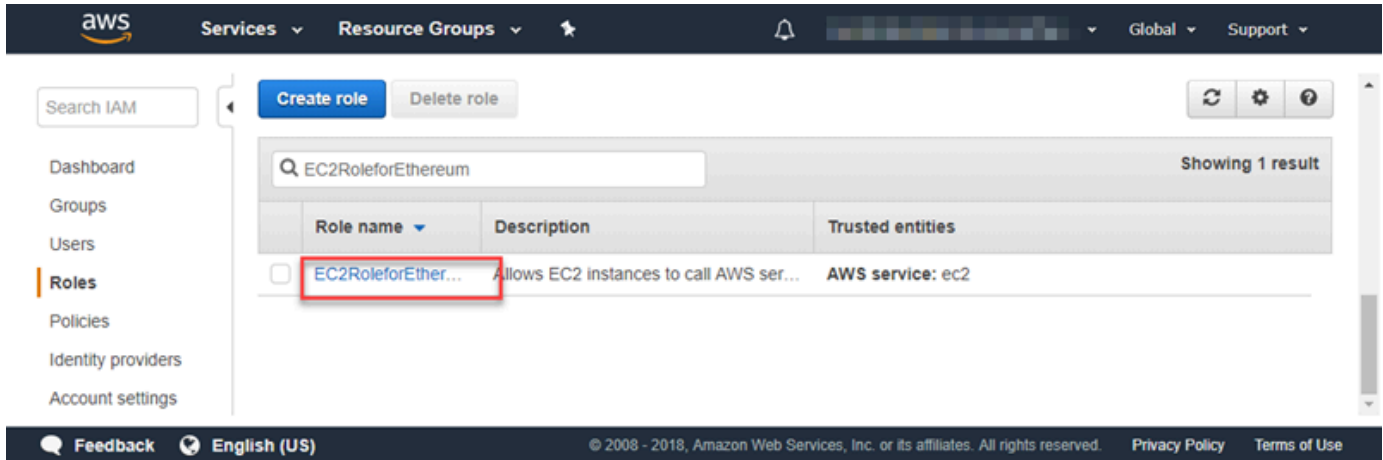
Filter: Policy type Showing 1 result

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

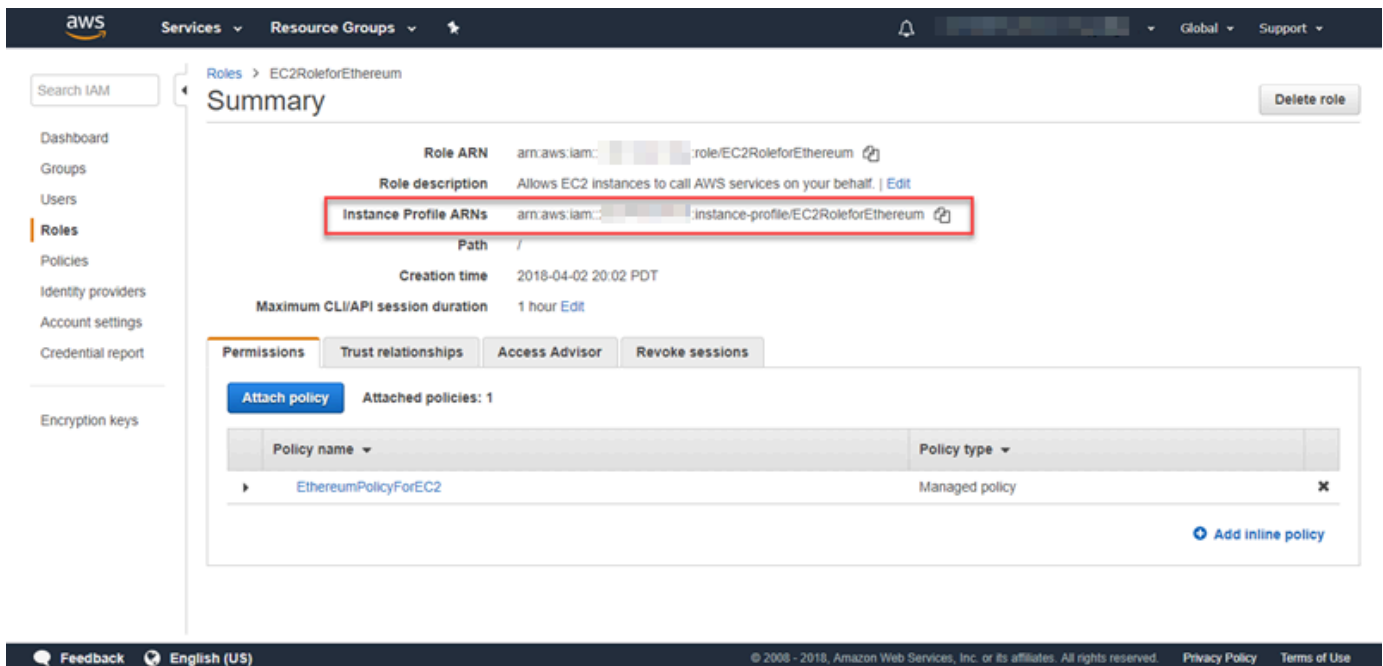
* Required Cancel Previous **Next: Review**

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

9. Geben Sie für Rollenname einen Wert ein, anhand dessen Sie beispielsweise die Rolle identifizieren können EC2RoleForEthereum. Geben Sie unter Role description (Rollenbeschreibung) eine kurze Zusammenfassung ein. Klicken Sie anschließend auf Create role (Rolle erstellen).
10. Wählen Sie die gerade erstellte Rolle aus der Liste aus. Verfügt Ihr Konto über mehrere Rollen, können Sie den Rollennamen im Feld Search (Suche) eingeben.



11. Kopieren Sie den Wert des Instance Profile ARN (Instance-Profil-ARN) und speichern Sie ihn, damit Sie ihn wieder kopieren können. Sie benötigen diesen ARN beim Erstellen des Ethereum-Netzwerks.



Erstellen eines Bastion-Host

In diesem Tutorial erstellen Sie einen Bastion-Host. Dies ist eine EC2 Instanz, die Sie verwenden, um eine Verbindung zu den Webschnittstellen und Instanzen in Ihrem Ethereum-Netzwerk herzustellen. Sein alleiniger Zweck besteht darin, SSH-Verkehr von vertrauenswürdigen Clients außerhalb der VPC weiterzuleiten, damit diese auf Ethereum-Netzwerkressourcen zugreifen können.

Sie richten den Bastion-Host ein, weil der Application Load Balancer, der über die Vorlage erstellt wird, intern ist, d. h., er leitet nur interne IP-Adressen weiter. Der Bastion-Host:

- verfügt über eine interne IP-Adresse, die der Application Load Balancer erkennt, weil Sie ihn im zweiten öffentlichen Subnetz starten, das Sie zuvor erstellt haben.
- verfügt über eine öffentliche IP-Adresse, die das Subnetz zuweist, auf die vertrauenswürdige Quellen außerhalb der VPC zugreifen können.
- ist der Sicherheitsgruppe für den zuvor erstellten Application Load Balancer zugeordnet, der über eine eingehende Regel verfügt, die SSH-Datenverkehr (Port 22) von vertrauenswürdigen Clients zulässt.

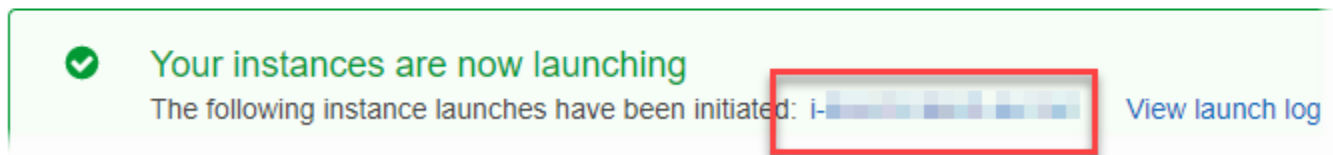
Um auf das Ethereum-Netzwerk zugreifen zu können, müssen vertrauenswürdige Clients für die Verbindung über den Bastion-Host eingerichtet werden. Weitere Informationen finden Sie unter [Connect zum Bastion-Host her EthStats und EthExplorer verwenden Sie ihn](#). Ein Bastion-Host ist ein Ansatz. Sie können jeden Ansatz verwenden, der den Zugriff von vertrauenswürdigen Clients auf private Ressourcen innerhalb einer VPC ermöglicht.

So erstellen Sie einen Bastion-Host

1. Folgen Sie den ersten fünf Schritten zum [Starten einer Instance](#) im EC2 Amazon-Benutzerhandbuch.
2. Wählen Sie "Edit Instance Details. Wählen Sie für Network (Netzwerk) die zuvor erstellte VPC und für Subnet (Subnetz) das zuvor erstellte zweite öffentliche Subnetz. Belassen Sie alle anderen Einstellungen auf ihren Standardwerten.
3. Bestätigen Sie die Änderung, wenn Sie dazu aufgefordert werden, und wählen Sie dann Review and Launch (Überprüfen und starten).
4. Klicken Sie auf Edit Security Groups (Sicherheitsgruppen bearbeiten). Wählen Sie für Sicherheitsgruppe zuweisen die Option Existierende Sicherheitsgruppe auswählen aus.

5. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe für den Application Load Balancer aus, die Sie zuvor erstellt haben, und wählen Sie dann Review and Launch (Prüfen und starten).
6. Wählen Sie Launch (Starten) aus.
7. Notieren Sie sich die Instance-ID. Sie benötigen sie später zum [Connect zum Bastion-Host her](#) [EthStats und EthExplorer verwenden Sie ihn](#).

Launch Status



Erstellen des Ethereum-Netzwerks

Das Ethereum-Netzwerk, das Sie mithilfe der Vorlage in diesem Thema angeben, startet einen CloudFormation Stack, der einen Amazon EC2 ECS-Instance-Cluster für das Ethereum-Netzwerk erstellt. Die Vorlage stützt sich auf die Ressourcen, die Sie zuvor in [Einrichten der Voraussetzungen](#) erstellt haben.

Wenn Sie den CloudFormation Stack mithilfe der Vorlage starten, erstellt er verschachtelte Stapel für einige Aufgaben. Nachdem diese erstellt wurden, können Sie eine Verbindung zu Ressourcen herstellen, die über den Application Load Balancer des Netzwerks mit dem Bastion-Host bereitgestellt werden, um zu verifizieren, ob Ihr Ethereum-Netzwerk läuft und zugänglich ist.

Um das Ethereum-Netzwerk mit der AWS-Blockchain-Template für Ethereum zu erstellen

1. Weitere Informationen finden Sie unter [Erste Schritte mit AWS-Blockchain-Templates](#) und öffnen Sie die neueste AWS-Blockchain-Template für Ethereum in der CloudFormation Konsole über die Quicklinks für Ihre AWS-Region.
2. Geben Sie die Werte entsprechend der folgenden Richtlinien ein:
 - Geben Sie unter Stack name (Stack-Name) einen Namen ein, den Sie leicht identifizieren können. Dieser Name wird innerhalb der Namen der Ressourcen verwendet, die durch den Stack erstellt werden.

- Behalten Sie unter Ethereum Network Parameters (Ethereum-Netzwerkparameter) und Private Ethereum Network Parameters (Private Ethereum-Netzwerkparameter) die Standardeinstellungen bei.

⚠ Warning

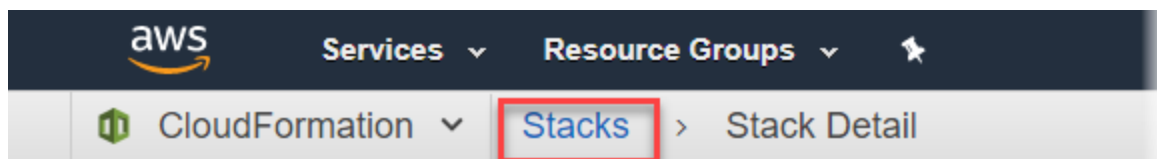
Verwenden Sie die Standardkonten und die zugehörige mnemonische Phrase nur zu Testzwecken. Senden Sie keinen echten Ether mit dem Standardkontensatz, da jeder Benutzer mit Zugriff auf die mnemonische Phrase auf Ether zugreifen oder aus den Konten stehlen kann. Geben Sie stattdessen benutzerdefinierte Konten für Produktionszwecke an. Die dem Standardkonto zugeordnete mnemonische Phrase lautet `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

- Behalten Sie unter Plattformkonfiguration die Standardeinstellungen bei, wodurch ein Amazon EC2 ECS-Instance-Cluster erstellt wird. Die Alternative, Docker-Local, erstellt ein Ethereum-Netzwerk mit einer einzigen EC2 Instanz.
- Wählen Sie unter EC2 Konfiguration die Optionen gemäß den folgenden Richtlinien aus:
 - Wählen Sie für EC2 key pair ein Schlüsselpaar aus. Weitere Informationen zum Erstellen eines Schlüsselpaars finden Sie unter [Erstellen eines Schlüsselpaars](#).
 - Wählen Sie für EC2 Sicherheitsgruppe die Sicherheitsgruppe aus, die Sie zuvor in erstellt haben [Erstellen von Sicherheitsgruppen](#).
 - Geben Sie für EC2 Instanzprofil ARN den ARN des Instanzprofils ein, das Sie zuvor in erstellt haben [Erstellen Sie eine IAM-Rolle für Amazon ECS und ein EC2 Instance-Profil](#).
- Wählen Sie unter VPC network configuration (VPC-Netzwerkkonfiguration) die Optionen gemäß den folgenden Richtlinien aus:
 - Wählen Sie für VPC ID die VPC aus, die Sie zuvor in [Erstellen von VPC und Subnetzen](#) erstellt haben.
 - Wählen Sie für Ethereum Network Subnet IDs das einzelne private Subnetz aus, das Sie zuvor im Verfahren erstellt haben. [To create the VPC](#)
- Behalten Sie unter ECS cluster configuration (ECS-Cluster-Konfiguration) die Standardwerte bei. Dadurch wird ein ECS-Cluster aus drei EC2 Instanzen erstellt.
- Wählen Sie unter Application Load Balancer configuration (Konfiguration des Application Load Balancer) (ECS only/Nur ECS) die Optionen gemäß den folgenden Richtlinien aus:

- Wählen Sie für Application Load Balancer Subnet IDs zwei öffentliche Subnetze aus den bereits genannten [list of subnets](#) aus.
 - Wählen Sie bei Application Load Balancer Security Group (Sicherheitsgruppe für Application Load Balancer) die Sicherheitsgruppe für den Application Load Balancer aus, die Sie zuvor in [Erstellen von Sicherheitsgruppen](#) erstellt haben.
 - Geben Sie für IAM-Rolle den ARN der ECS-Rolle ein, die Sie zuvor in [Erstellen Sie eine IAM-Rolle für Amazon ECS und ein EC2 Instance-Profil](#) erstellt haben.
 - Wählen Sie EthStatsunter Optionen gemäß den folgenden Richtlinien aus:
 - Behalten Sie für Deploy EthStats die Standardeinstellung bei, die wahr ist.
 - Geben Sie für EthStats Connection Secret einen beliebigen Wert ein, der mindestens sechs Zeichen lang ist.
 - EthExplorerBehalten Sie unter die Standardeinstellung für Deploy bei EthExplorer, die wahr ist.
 - Behalten Sie unter Other parameters (Andere Parameter) den Standardwert für Nested Template S3 URL Prefix (Verschachteltes Vorlagen-S3-URL-Präfix) bei und notieren Sie diesen. Hier finden Sie verschachtelte Vorlagen.
3. Behalten Sie für alle anderen Einstellungen den Standardwert bei, markieren Sie zur Bestätigung das Kontrollkästchen und klicken Sie dann auf Create (Erstellen).

Die Stack-Detailseite für den Root-Stack, der CloudFormation gestartet wird, wird angezeigt.

4. Um den Fortschritt des Root-Stack und der verschachtelten Stacks zu überwachen, wählen Sie Stacks aus.



MyFirstEthereumStack

Stack name: MyFirstEthereumStack

5. Wenn alle Stacks CREATE_COMPLETE als Status anzeigen, können Sie eine Verbindung zu den Benutzeroberflächen von Ethereum herstellen, um zu überprüfen, ob das Netzwerk läuft und zugänglich ist. Wenn Sie die ECS-Containerplattform URLs für die Verbindung zu EthStats EthExplorer, und EthJson RPC über den Application Load Balancer verwenden, sind sie auf der Registerkarte Outputs des Root-Stacks verfügbar.

⚠️ Wichtig

Sie können keine direkte Verbindung zu diesen URLs oder SSH herstellen, bis Sie eine Proxyverbindung über den Bastion-Host auf Ihrem Client-Computer eingerichtet haben. Weitere Informationen finden Sie unter [Connect zum Bastion-Host her EthStats und EthExplorer verwenden Sie ihn](#).

The screenshot shows the AWS CloudFormation console. At the top, there are navigation menus for 'Services', 'Resource Groups', and 'Stacks'. Below that, 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, the 'Outputs' tab is active, showing a table of stack outputs. The 'EthStatsURL' output is highlighted with a red box, showing a URL that has been partially redacted with a blue 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 ...	

Connect zum Bastion-Host her EthStats und EthExplorer verwenden Sie ihn

Um eine Verbindung zu den Ethereum-Ressourcen in diesem Tutorial herzustellen, richten Sie eine SSH-Port-Weiterleitung (SSH-Tunneling) über den Bastion-Host ein. Die folgenden Anweisungen zeigen, wie Sie dies tun können, damit Sie eine Verbindung zu einem Browser herstellen EthStats und EthExplorer URLs ihn verwenden können. In den folgenden Anweisungen richten Sie zunächst einen SOCKS-Proxy auf einem lokalen Port ein. Anschließend verwenden Sie

eine Browsererweiterung, [FoxyProxy](#), um diesen weitergeleiteten Port für Ihr Ethereum-Netzwerk zu verwenden URLs.

Wenn Sie Mac OS oder Linux nutzen, verwenden Sie einen SSH-Client, um die SOCKS-Proxy-Verbindung zum Bastion-Host einzurichten. Wenn Sie Windows nutzen, verwenden Sie PuTTY. Bevor Sie die Verbindung herstellen, bestätigen Sie, dass der von Ihnen verwendete Client-Computer in der Sicherheitsgruppe für den Application Load Balancer, die Sie zuvor eingerichtet haben, als zulässige Quelle für eingehenden SSH-Verkehr angegeben ist.

So stellen Sie eine Verbindung zum Bastion-Host mit SSH-Portweiterleitung über SSH her

- Folgen Sie den Anweisungen unter [Herstellen einer Verbindung zu Ihrer Linux-Instance mithilfe von SSH](#) im EC2 Amazon-Benutzerhandbuch. Fügen Sie für Schritt 4 des Verfahrens [Verbinden mit Ihrer Linux-Instance](#) den SSH-Befehl `-D 9001` hinzu, geben Sie dasselbe key pair an, das Sie in der Konfiguration AWS Blockchain Template for Ethereum angegeben haben, und geben Sie den DNS-Namen des Bastion-Hosts an.

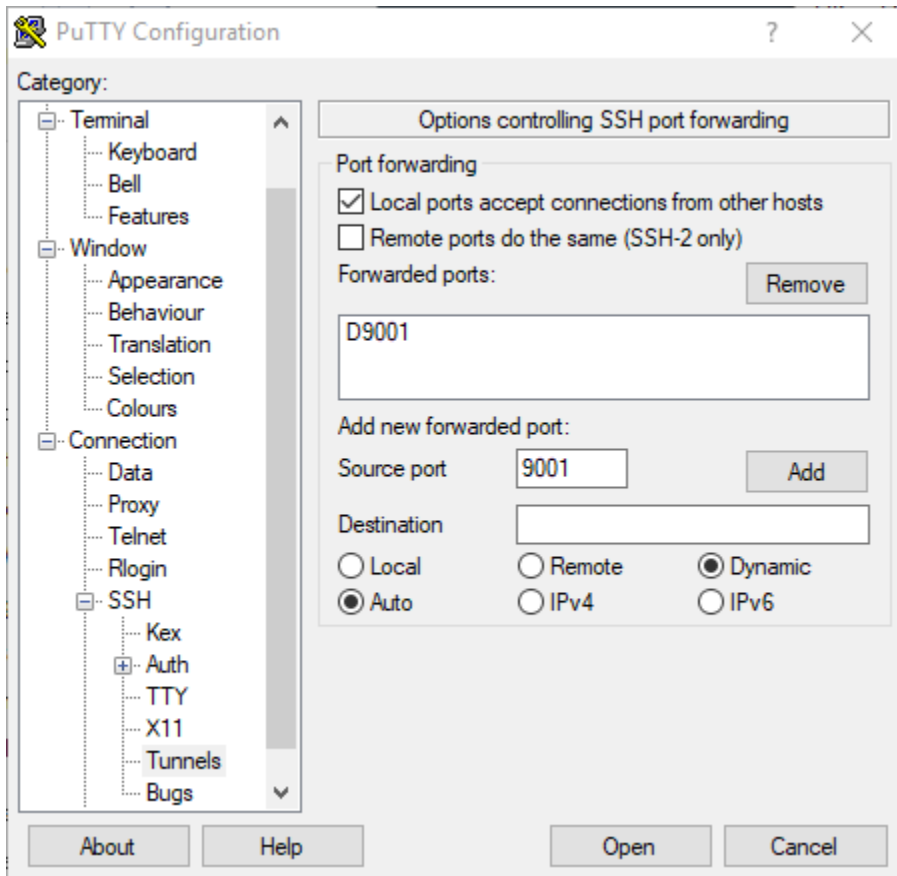
```
ssh -i /path/my-template-key-pair.pem ec2-user@bastion-host-dns -D 9001
```

So stellen Sie eine Verbindung zum Bastion-Host mit SSH-Portweiterleitung über PuTTY (Windows) her

1. Folgen Sie den Anweisungen unter [Herstellen einer Verbindung zu Ihrer Linux-Instance von Windows mithilfe von PuTTY](#) im EC2 Amazon-Benutzerhandbuch bis Schritt 7 des Verfahrens [Starten einer PuTTY-Sitzung](#) und verwenden Sie dabei dasselbe key pair, das Sie in der Konfiguration AWS Blockchain Template for Ethereum angegeben haben.
2. Wählen Sie in PuTTY unter Category (Kategorie) die Optionen Connection (Verbindung), SSH und Tunnels (Tunnel).
3. Wählen Sie für Port forwarding (Portweiterleitung) die Option Local ports accept connections from other hosts (Lokale Ports akzeptieren Verbindungen von anderen Hosts)
4. Gehen Sie unter Add new forwarded port (Neuen weitergeleiteten Port hinzufügen) wie folgt vor:
 - a. Geben Sie für Source port (Quell-Port) 9001 ein. Dies ist ein willkürlich gewählter, ungenutzter Port, den wir ausgewählt haben. Sie können bei Bedarf einen anderen Port auswählen.
 - b. Lassen Sie Destination (Ziel) leer.
 - c. Wählen Sie Dynamic (Dynamisch) aus.

- d. Wählen Sie Hinzufügen aus.

Für Forwarded ports (Weitergeleitete Ports) sollte D9001 wie unten dargestellt angezeigt werden.



5. Wählen Sie Open (Öffnen) und authentifizieren Sie sich dann bei dem Bastion-Host, wie es Ihre Schlüsselkonfiguration erfordert. Lassen Sie die Verbindung offen.

Wenn die PuTTY-Verbindung geöffnet ist, konfigurieren Sie jetzt Ihr System oder eine Browsererweiterung so, dass der weitergeleitete Port für Ihr Ethereum-Netzwerk URLs verwendet wird. Die folgenden Anweisungen basieren auf der Verwendung von FoxyProxy Standard zur Weiterleitung von Verbindungen auf der Grundlage des URL-Musters von EthStats und EthExplorer und des Ports 9001, den Sie zuvor als weitergeleiteten Port eingerichtet haben. Sie können jedoch jede Methode verwenden, die Sie bevorzugen.

Um die Verwendung des SSH-Tunnels für das Ethereum-Netzwerk zu konfigurieren FoxyProxy URLs

Dieses Verfahren wurde basierend auf Chrome geschrieben. Wenn Sie einen anderen Browser verwenden, übersetzen Sie die Einstellungen und die Reihenfolge in die Version von FoxyProxy für diesen Browser.

1. Laden Sie die FoxyProxy Standard-Browsererweiterung herunter, installieren Sie sie und öffnen Sie dann die Optionen gemäß den Anweisungen für Ihren Browser.
2. Klicken Sie auf Add New Proxy (Neuen Proxy hinzufügen).
3. Stellen Sie auf der Registerkarte General (Allgemein) sicher, dass der Proxy auf Enabled (Aktiviert) steht und geben Sie einen Proxy Name (Proxy-Namen) und Proxy Notes (Proxy-Notizen) ein, die Ihnen helfen, diese Proxy-Konfiguration zu identifizieren.
4. Wählen Sie auf der Registerkarte Proxy Details die Option Manual Proxy Configuration (Manuelle Proxy-Konfiguration). Für Host or IP Address (Host oder IP-Adresse) (oder Server or IP Address (Server oder IP-Adresse) in einigen Versionen) geben Sie localhost ein. Geben Sie im Feld Port 9001 ein. Wählen Sie SOCKS proxy? (SOCKS-Proxy?).
5. Wählen Sie auf der Registerkarte URL Pattern (URL-Muster) die Option Add New Pattern (Neues Muster hinzufügen) aus.
6. Geben Sie als Musternamen einen Namen ein, der leicht zu identifizieren ist, und geben Sie für URL-Muster ein Muster ein, das mit allen Ethereum-Ressourcen übereinstimmt, die URLs Sie mit der Vorlage erstellt haben, zum Beispiel `http://internal - MyUser -loadB-*`. Informationen zum Ansehen URLs finden Sie unter [Ethereum URLs](#)
7. Behalten Sie die Standardauswahl für die anderen Einstellungen bei und wählen Sie Save (Speichern).

Sie können jetzt eine Verbindung zu Ethereum herstellen URLs, die auf der CloudFormation Konsole verfügbar sind. Verwenden Sie dazu den Tab Outputs des Root-Stacks, den Sie mit der Vorlage erstellt haben.

Bereinigen von Ressourcen

CloudFormation macht es einfach, Ressourcen zu bereinigen, die der Stack erstellt hat. Wenn Sie den Stack löschen, werden alle Ressourcen, die der Stack erstellt hat, gelöscht.

So löschen Sie Ressourcen, die die Vorlage erstellt hat

- Öffnen Sie die CloudFormation Konsole, wählen Sie den Root-Stack aus, den Sie zuvor erstellt haben, und wählen Sie Aktionen, Löschen.

Der Status des zuvor angelegten Root-Stack und die zugehörigen verschachtelten Stacks werden auf DELETE_IN_PROGRESS aktualisiert.

Sie können die vorbereitenden Schritte, die Sie für das Ethereum-Netzwerk erstellt haben, löschen.

Löschen der VPC

- Öffnen Sie die Amazon VPC-Konsole, wählen Sie die VPC aus, die Sie zuvor erstellt haben, und wählen Sie dann Aktionen, VPC löschen. Dadurch werden auch die Subnetze, Sicherheitsgruppen und das mit der VPC verbundene NAT-Gateway gelöscht.

Löschen Sie die IAM-Rolle und das Instance-Profil EC2

- Öffnen Sie die IAM-Konsole und wählen Sie Rollen aus. Wählen Sie die Rolle für ECS und die Rolle aus EC2 , die Sie zuvor erstellt haben, und klicken Sie auf Löschen.

Beenden Sie die EC2 Instanz für den Bastion-Host

- Öffnen Sie das EC2 Amazon-Dashboard, wählen Sie Running Instances, wählen Sie die EC2 Instance aus, die Sie für den Bastion-Host erstellt haben, wählen Sie Actions, Instance State, Terminate aus.

AWS-Blockchain-Templates und -Funktionen

Dieser Abschnitt enthält Links, über die Sie sofort mit der Erstellung eines Blockchain-Netzwerks beginnen können, sowie Informationen zu Konfigurationsmöglichkeiten und Voraussetzungen für die Einrichtung des Netzwerks auf AWS.

Die folgenden Vorlagen sind verfügbar:

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

AWS Blockchain Templates ist in den folgenden Regionen verfügbar:

- Region USA West (Oregon) (us-west-2)
- Region USA Ost (Nord-Virginia) (us-east-1)
- Region USA Ost (Ohio) (us-east-2)

Note

Wenn Sie eine Vorlage in einer Region ausführen, die oben nicht aufgeführt ist, werden Ressourcen in der Region USA Ost (Nord-Virginia) (us-east-1) gestartet.

Verwenden der AWS-Blockchain-Template für Ethereum

Ethereum ist ein Blockchain-Framework, das intelligente Verträge mit Solidity, einer Ethereum-spezifischen Sprache, ausführt. Homestead ist die neueste Version von Ethereum. Weitere Informationen finden Sie in der [Ethereum Homestead-Dokumentation und der Solidity-Dokumentation](#).

Links für den Start

Links zum [Start CloudFormation in bestimmten Regionen mithilfe der Ethereum-Vorlagen finden Sie unter Erste Schritte mit AWS-Blockchain-Templates](#).

Ethereum-Optionen

Wenn Sie das Ethereum-Netzwerk mithilfe der Vorlage konfigurieren, treffen Sie Entscheidungen, die die nachfolgenden Anforderungen bestimmen:

- [Auswahl der Container-Plattform](#)
- [Auswahl eines privaten oder öffentlichen Ethereum-Netzwerks](#)
- [Ändern der Standardkonten und der mnemonischen Phrase](#)

Auswahl der Container-Plattform

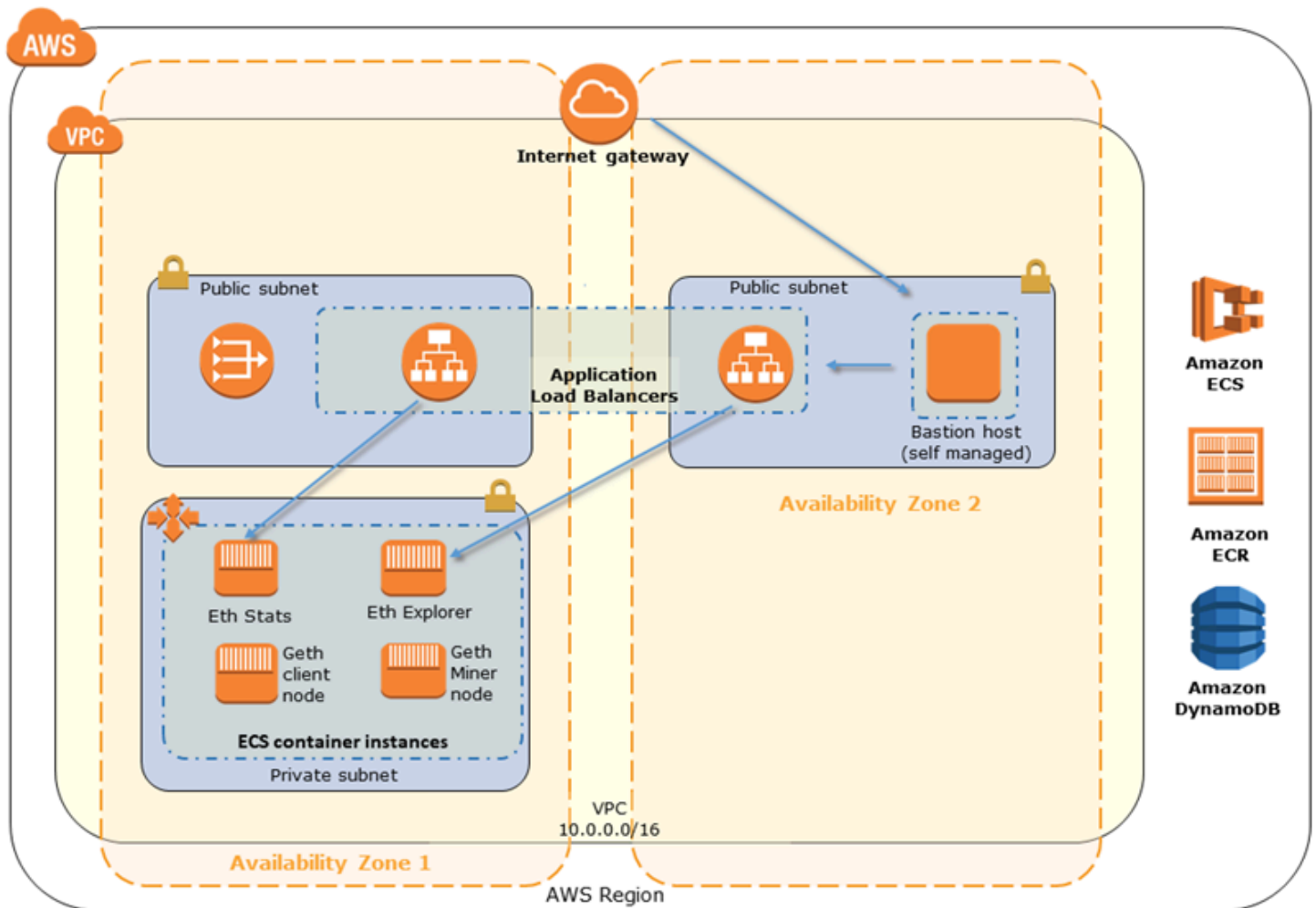
AWS-Blockchain-Templates verwenden Docker-Container, die in Amazon ECR gespeichert sind, um Blockchain-Software bereitzustellen. Das AWS-Blockchain-Template für Ethereum bietet zwei Optionen für die Container-Plattform:

- `ecs` — Gibt an, dass Ethereum auf einem Amazon ECS-Cluster von EC2 Amazon-Instances läuft.
- `docker-local` — Gibt an, dass Ethereum auf einer einzigen Instanz ausgeführt wird. EC2

Verwendung der Amazon ECS-Container-Plattform

Mit Amazon ECS erstellen Sie Ihr Ethereum-Netzwerk auf einem ECS-Cluster, der aus mehreren EC2 Instances besteht, mit einem Application Load Balancer und zugehörigen Ressourcen. Weitere Informationen zur Verwendung der Amazon ECS-Konfiguration finden Sie im [Erste Schritte mit AWS-Blockchain-Templates](#) Tutorial.

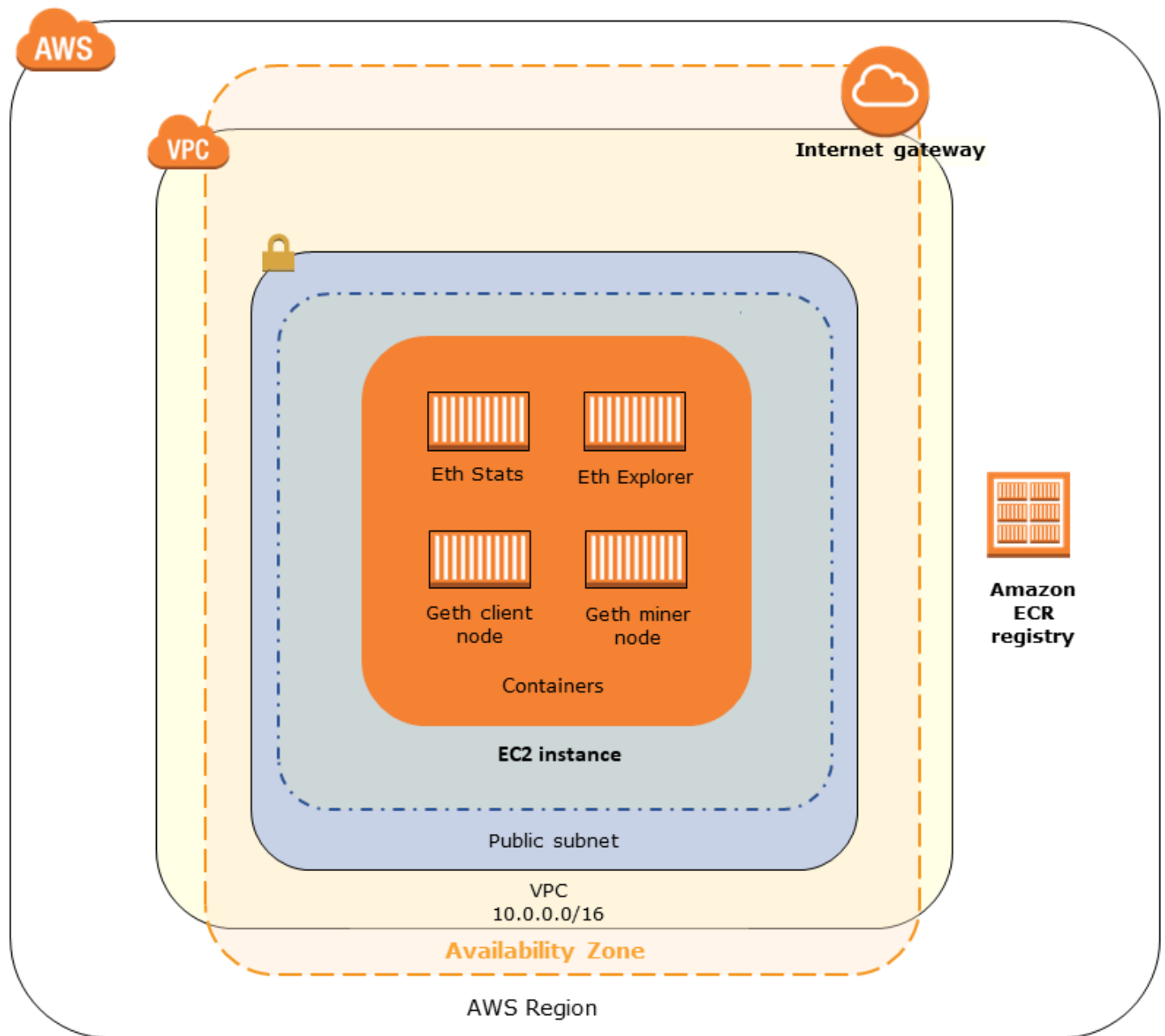
Das folgende Diagramm zeigt ein Ethereum-Netzwerk, das mithilfe der Vorlage mit der ECS-Container-Platfformoption erstellt wurde:



Verwendung der Docker-Local-Plattform

Alternativ können Sie Ethereum-Container innerhalb einer einzigen EC2 Amazon-Instance starten. Alle Container laufen auf einer einzigen EC2 Instance. Dies ist eine vereinfachte Einrichtung.

Das folgende Diagramm zeigt ein Ethereum-Netzwerk, das mithilfe der Vorlage mit der Docker-Local-Container-Plattformoption erstellt wurde:



Auswahl eines privaten oder öffentlichen Ethereum-Netzwerks

Wenn Sie einen anderen Ethereum-Netzwerk-ID-Wert als 1-4 wählen, werden private Ethereum-Knoten erstellt, die innerhalb eines von Ihnen definierten Netzwerks unter Verwendung der von Ihnen angegebenen privaten Netzwerkparameter ausgeführt werden.

Wenn Sie eine Ethereum-Netzwerk-ID von 1—4 wählen, werden die von Ihnen erstellten Ethereum-Knoten mit dem öffentlichen Ethereum-Netzwerk verbunden. Sie können private Netzwerkeinstellungen und deren Standardeinstellungen ignorieren. Wenn Sie sich dafür

entscheiden, Ethereum-Knoten mit dem öffentlichen Ethereum-Netzwerk zu verbinden, stellen Sie sicher, dass die entsprechenden Dienste in Ihrem Netzwerk über das Internet zugänglich sind.

Ändern der Standardkonten und der mnemonischen Phrase

Eine mnemonische Phrase ist eine zufällige Wortgruppe, die Sie verwenden können, um Ethereum-Wallets (d. h. private/public Schlüsselpaare) für zugehörige Konten in einem beliebigen Netzwerk zu generieren. Die mnemonische Phrase kann verwendet werden, um auf Ether für verknüpfte Konten zuzugreifen. Wir haben eine Standard-Mnemonik erstellt, die den Standardkonten zugeordnet ist, die von der Ethereum-Vorlage verwendet werden.

Warning

Verwenden Sie die Standardkonten und die zugehörige mnemonische Phrase nur zu Testzwecken. Senden Sie keinen echten Ether mit dem Standardkontensatz, da jeder Benutzer mit Zugriff auf die mnemonische Phrase auf Ether zugreifen oder aus den Konten stehlen kann. Geben Sie stattdessen benutzerdefinierte Konten für Produktionszwecke an. Die dem Standardkonto zugeordnete mnemonische Phrase lautet `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`.

Voraussetzungen

Wenn Sie Ihr Ethereum-Netzwerk mithilfe der AWS Blockchain Template for Ethereum einrichten, müssen die unten aufgeführten Mindestanforderungen erfüllt sein. Für die Vorlage sind die AWS Komponenten erforderlich, die für jede der folgenden Kategorien aufgeführt sind:

Themen

- [Voraussetzungen für den Zugriff auf Ethereum-Ressourcen](#)
- [IAM-Voraussetzungen](#)
- [Sicherheitsgruppen-Voraussetzungen](#)
- [VPC-Voraussetzungen](#)
- [Beispiel für IAM-Berechtigungen für das EC2 Instanzprofil und die ECS-Rolle](#)

Voraussetzungen für den Zugriff auf Ethereum-Ressourcen

Voraussetzung	Für ECS-Plattform	Für Docker-Local
Ein EC2 Amazon-Schlüsselpaar, mit dem Sie auf EC2 Instances zugreifen können. Der Schlüssel muss sich in derselben Region wie der ECS-Cluster und die anderen Ressourcen befinden.	✓	✓
Eine mit dem Internet verbundene Komponente, z. B. ein Bastion-Host oder ein mit dem Internet verbundener Load Balancer, mit einer internen Adresse, von der aus Datenverkehr in den Application Load Balancer zugelassen wird. Dies ist bei der ECS-Plattform erforderlich, da die Vorlage aus Sicherheitsgründen einen internen Load Balancer erstellt. Dies ist bei der Docker-Local-Plattform erforderlich, wenn sich die EC2 Instance in einem privaten Subnetz befindet, was wir empfehlen. Weitere Informationen zum Konfigurieren eines Bastion-Hosts finden Sie unter Erstellen eines Bastion-Hosts .	✓	✓ (mit privatem Subnetz)

IAM-Voraussetzungen

Voraussetzung	Für ECS-Plattform	Für Docker-Local
Ein IAM-Prinzipal (Benutzer oder Gruppe), der berechtigt ist, mit allen zugehörigen Diensten zu arbeiten.	✓	✓
Ein EC2 Amazon-Instance-Profil mit entsprechenden Berechtigungen für EC2 Instances zur Interaktion mit anderen Services. Weitere Informationen finden Sie unter To create an EC2 instance profile .	✓	✓
Eine IAM-Rolle mit Berechtigungen für Amazon ECS zur Interaktion mit anderen Services. Weitere Informationen finden Sie unter Erstellen der ECS-Rolle und Berechtigungen .	✓	

Sicherheitsgruppen-Voraussetzungen

Voraussetzung	Für ECS-Plattform	Für Docker-Local
Eine Sicherheitsgruppe für EC2 Instances mit den folgenden Anforderungen:	✓	✓
<ul style="list-style-type: none"> Die Regeln für ausgehenden Datenverkehr, die erlauben, dass der 	✓	✓

Voraussetzung	Für ECS-Plattform	Für Docker-Local
Datenverkehr 0.0.0.0/0 (Standard).		
<ul style="list-style-type: none"> Eine Regel für eingehenden Datenverkehr, die den gesamten Datenverkehr von sich aus (von derselben Sicherheitsgruppe) zulässt. 	✓	✓
<ul style="list-style-type: none"> Eine Regel für eingehenden Datenverkehr, die den gesamten Datenverkehr von der Sicherheitsgruppe für den Application Load Balancer zulässt. 	✓	
<ul style="list-style-type: none"> Regeln für eingehenden Datenverkehr, die HTTP (Port 80), EthStats (bereitgestellt auf Port 8080), JSON RPC über HTTP (Port 8545) und SSH (Port 22) von vertrauenswürdigen externen Quellen zulassen, wie z. B. dem IP-CIDR Ihres Client-Computers. 		✓

Voraussetzung	Für ECS-Plattform	Für Docker-Local
<p>Eine Sicherheitsgruppe für den Application Load Balancer mit den folgenden Voraussetzungen:</p> <ul style="list-style-type: none">• Eine Regel für eingehenden Datenverkehr, die den gesamten Datenverkehr von sich aus (von derselben Sicherheitsgruppe) zulässt.• Eine Regel für eingehenden Datenverkehr, die den gesamten Datenverkehr aus der Sicherheitsgruppe für Instances zulässt. EC2• Regeln für ausgehenden Datenverkehr, die den gesamten Datenverkehr nur an die Sicherheitsgruppe für EC2 Instances zulassen. Weitere Informationen finden Sie unter Erstellen von Sicherheitsgruppen.• Wenn dieselbe Sicherheitsgruppe einem Bastion-Host zugeordnet wird, eine Regel für eingehenden Datenverkehr, die SSH-Datenverkehr (Port 22) von vertrauenswürdigen Quellen zulässt.• Wenn der Bastion-Host oder eine andere mit dem Internet verbunden	✓	

Voraussetzung	Für ECS-Plattform	Für Docker-Local
e Komponente zu einer anderen Sicherheitsgruppe gehört, eine Regel für eingehenden Datenverkehr, die den Datenverkehr von dieser Komponente zulässt.		

VPC-Voraussetzungen

Voraussetzung	Für ECS-Plattform	Für Docker-Local
Eine elastische IP-Adresse, die für den Zugriff auf Ethereum-Dienste verwendet wird.	✓	✓
Ein Subnetz zum Ausführen von EC2 Instances. Wir empfehlen Ihnen dringend, ein privates Subnetz zu verwenden.	✓	✓
Zwei öffentlich zugängliche Subnetze. Jedes Subnetz muss sich in unterschiedlichen Availability Zones befinden, wobei sich eine in derselben Availability Zone wie das Subnetz für Instances befinden muss. EC2	✓	

Beispiel für IAM-Berechtigungen für das EC2 Instanzprofil und die ECS-Rolle

Sie geben einen ARN für ein EC2 Instanzprofil als einen der Parameter an, wenn Sie die Vorlage verwenden. Wenn Sie die ECS-Container-Plattform verwenden, geben Sie zusätzlich einen ARN für die ECS-Rolle an. Die diesen Rollen zugeordneten Berechtigungsrichtlinien gestatten den AWS-Ressourcen und Instances in Ihrem Cluster, mit anderen AWS-Ressourcen zu interagieren. Weitere Informationen finden Sie unter [IAM-Rollen](#) im IAM-Benutzerhandbuch. Verwenden Sie die folgenden Richtlinienanweisungen und Verfahren als Basis für die Erstellung von Berechtigungen.

Beispiel für eine Berechtigungsrichtlinie für das EC2 Instanzprofil

Die folgende Berechtigungsrichtlinie zeigt, welche Aktionen für das EC2 Instance-Profil zulässig sind, wenn Sie sich für die ECS-Containerplattform entscheiden. Dieselben Richtlinienanweisungen können auf einer Docker-Local-Container-Plattform verwendet werden, wobei `ecs`-Kontextschlüssel entfernt werden, um den Zugriff einzuschränken.

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

Erstellen der ECS-Rolle und Berechtigungen

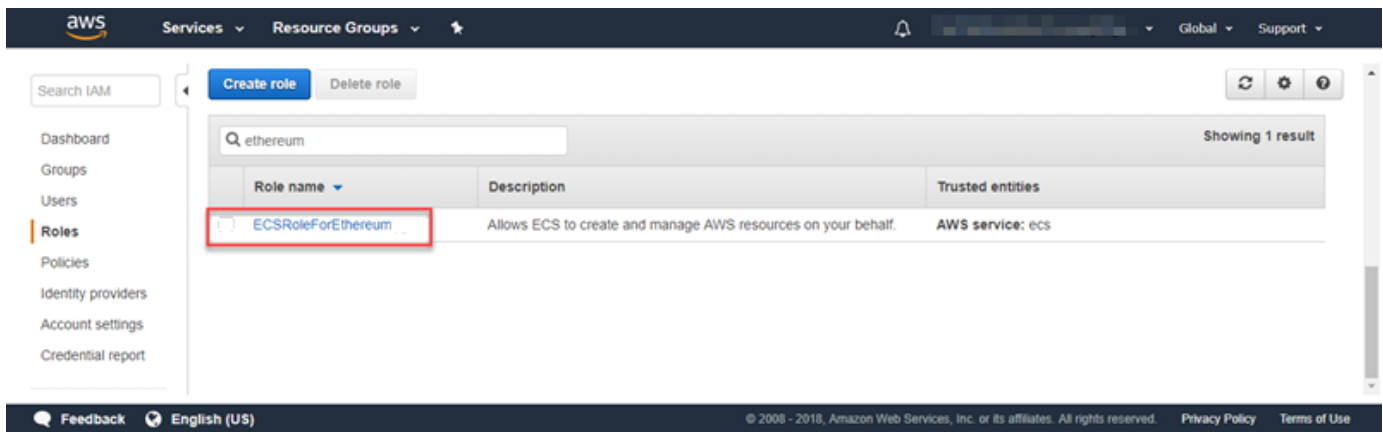
Für die mit der ECS-Rolle verknüpften Berechtigungen empfehlen wir, mit der EC2ContainerServiceRoleAmazon-Berechtigungsrichtlinie zu beginnen. Führen Sie die folgenden Schritte aus, um eine Rolle zu erstellen und diese Berechtigungsrichtlinie anzufügen. Verwenden Sie die IAM-Konsole, um die meisten up-to-date Berechtigungen in dieser Richtlinie einzusehen.

So erstellen Sie die IAM-Rolle für Amazon ECS

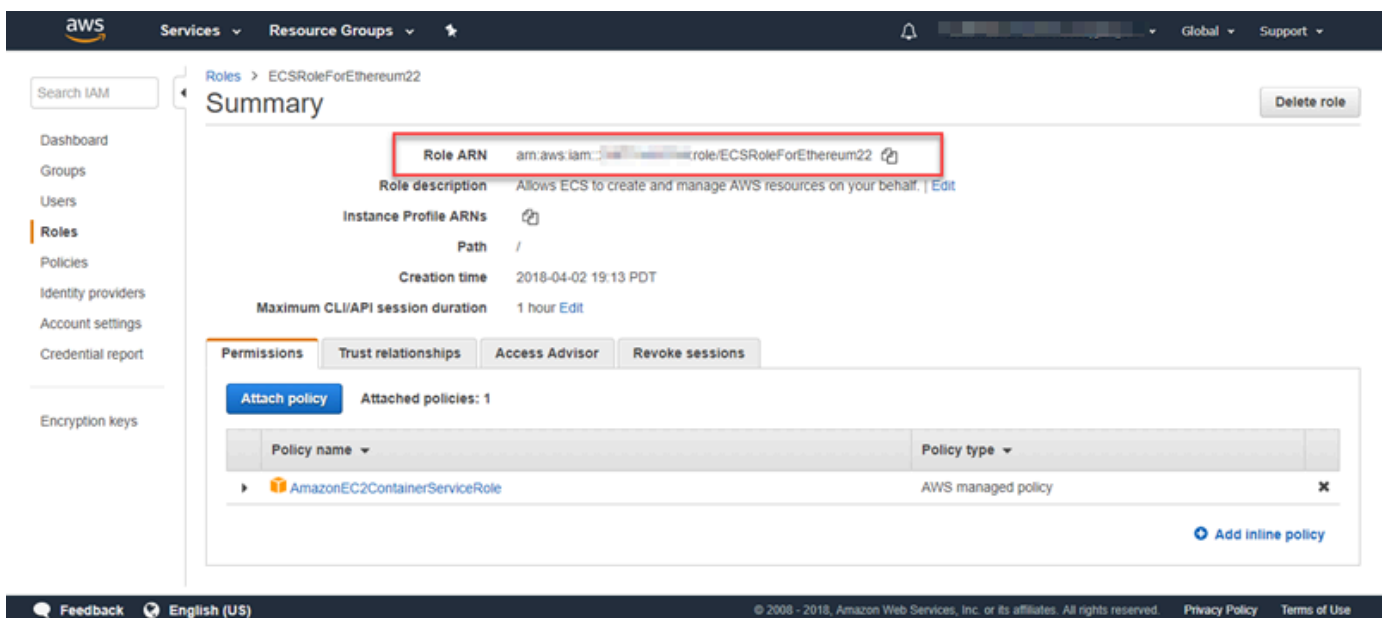
1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und Create Role (Rolle erstellen) aus.
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS service (AWS-Service) aus.
4. Wählen Sie für Choose the service that will use this role (Service auswählen, der diese Rolle verwendet) die Option Elastic Container Service aus.
5. Wählen Sie unter Select your use case (Auswahl Ihres Anwendungsfalls) die Optionen Elastic Container Service (Elastischer Container-Service), Next:Permissions (Weiter: Berechtigungen) aus.

The screenshot shows the AWS IAM console 'Create role' page. The 'Select type of trusted entity' section has '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. Lassen Sie für die Berechtigungsrichtlinie die Standardrichtlinie (Amazon EC2 ContainerServiceRole) ausgewählt und wählen Sie Weiter: Überprüfen.
7. Geben Sie unter Rollename einen Wert ein, anhand dessen Sie die Rolle leichter identifizieren können, z. B. ECSRoleForEthereum Geben Sie unter Role Description (Rollenbeschreibung) eine kurze Zusammenfassung ein. Notieren Sie sich den Rollennamen für später.
8. Wählen Sie Rolle erstellen aus.
9. Wählen Sie die gerade erstellte Rolle aus der Liste aus. Wenn Ihr Konto viele Rollen hat, können Sie nach dem Rollennamen suchen.



10. Kopieren Sie den Wert Role ARN (Rollen-ARN) und speichern Sie ihn, sodass Sie ihn jederzeit erneut kopieren können. Sie benötigen diesen ARN beim Erstellen des Ethereum-Netzwerks.



Verbindung zu Ethereum-Ressourcen herstellen

Nachdem der Root-Stack, den Sie mit der Vorlage erstellen, CREATE_COMPLETE anzeigt, können Sie über die Konsole eine Verbindung zu Ethereum-Ressourcen herstellen. CloudFormation Die Art der Verbindung hängt von der Container-Plattform ab, die Sie wählen: ECS oder Docker-Local:

- ECS — Die Registerkarte „Ausgabe“ des Root-Stacks enthält Links zu Diensten, die auf dem Application Load Balancer ausgeführt werden. Auf URLs diese kann aus Sicherheitsgründen nicht direkt zugegriffen werden. Um eine Verbindung herzustellen, können Sie einen Bastion-Host für Proxy-Verbindungen zu ihnen einrichten und verwenden. Weitere Informationen hierzu finden Sie unter [Proxy-Verbindungen unter Verwendung eines Bastion-Hosts](#) weiter unten.

- `docker-local` — Sie stellen eine Verbindung mit der IP-Adresse der EC2 Instanz her, die die Ethereum-Dienste hostet, wie unten aufgeführt. Verwenden Sie die EC2 Konsole, um die *ec2-IP-address* Instanz zu finden, die die Vorlage erstellt hat.
 - `EthStats`— Verwenden Sie `http://ec2-IP-address`
 - `EthExplorer`— Verwenden Sie `http://:8080 ec2-IP-address`
 - `EthJsonRpc`—Verwenden Sie `http://:8545 ec2-IP-address`

Wenn Sie ein öffentliches Subnetz für die Ethereum Network Subnet ID (Ethereum-Netzwerk-Subnetz-ID) (Liste der in der Vorlage zu verwendenden VPC-Subnetze) angegeben haben, können Sie direkt eine Verbindung herstellen. Ihr Client muss eine vertrauenswürdige Quelle für eingehenden Datenverkehr für SSH (Port 22) sowie die aufgeführten Ports sein. Dies wird von der EC2 Sicherheitsgruppe bestimmt, die Sie mithilfe der AWS-Blockchain-Template für Ethereum angegeben haben.

Wenn Sie ein privates Subnetz angegeben haben, können Sie einen Bastion-Host für Proxy-Verbindungen diesen Adressen einrichten und verwenden. Weitere Informationen hierzu finden Sie unter [Proxy-Verbindungen unter Verwendung eines Bastion-Hosts](#) weiter unten.

Proxy-Verbindungen unter Verwendung eines Bastion-Hosts

Bei einigen Konfigurationen sind Ethereum-Dienste möglicherweise nicht öffentlich verfügbar. In diesen Fällen können Sie über einen Bastion-Host eine Verbindung zu Ethereum-Ressourcen herstellen. Weitere Informationen über Bastion-Hosts finden Sie unter [Linux-Bastion-Host-Architektur](#) im Linux Bastion-Host Quick Start-Leitfaden.

Der Bastion-Host ist eine Instanz EC2 . Stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Die EC2 Instanz für den Bastion-Host befindet sich in einem öffentlichen Subnetz, in dem `Auto-Assign Public IP` aktiviert ist und das über ein Internet-Gateway verfügt.
- Der Bastion-Host hat das `key pair`, das SSH-Verbindungen ermöglicht.
- Der Bastion-Host ist einer Sicherheitsgruppe zugeordnet, die eingehenden SSH-Verkehr von den Clients, die eine Verbindung herstellen, zulässt.
- Die den Ethereum-Hosts zugewiesene Sicherheitsgruppe (z. B. der Application Load Balancer, wenn ECS die Containerplattform ist, oder die EC2 Host-Instance, wenn Docker-Local die Container-Plattform ist) erlaubt eingehenden Datenverkehr auf allen Ports von Quellen innerhalb der VPC.

Stellen Sie bei der Einrichtung eines Bastion-Hosts sicher, dass die Clients, die eine Verbindung herstellen, den Bastion-Host als Proxy verwenden. Das folgende Beispiel zeigt die Einrichtung einer Proxyverbindung unter Mac OS. *BastionIP* Ersetzen Sie durch die IP-Adresse der EC2 Bastion-Host-Instance und *MySshKey.pem* durch die Schlüsselpaardatei, die Sie auf den Bastion-Host kopiert haben.

Geben Sie in der Befehlszeile Folgendes ein:

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

Dadurch wird die Portweiterleitung für Port 9001 auf dem lokalen Computer zum Bastion-Host eingerichtet.

Als Nächstes konfigurieren Sie Ihren Browser oder Ihr System für `localhost:9001` die Verwendung des SOCKS-Proxys. Beispielsweise wählen Sie unter Mac OS System Preferences (Systemeinstellungen), Network (Netzwerk), Advanced (Erweitert), SOCKS proxy (SOCKS-Proxy) und geben dann `localhost:9001` ein.

Wenn Sie FoxyProxy Standard mit Chrome verwenden, wählen Sie Weitere Tools, Erweiterungen aus. Wählen Sie unter FoxyProxy Standard die Optionen Details, Erweiterungsoptionen und Neuen Proxy hinzufügen aus. Wählen Sie Manual Proxy Configuration (Manuelle Proxy-Konfiguration) aus. Geben Sie für Host or IP Address (Host- oder IP-Adresse) `localhost` und für Port 9001 ein. Wählen Sie SOCKS proxy?, Save (Speichern) aus.

Sie sollten jetzt in der Lage sein, eine Verbindung zu den Ethereum-Hostadressen herzustellen, die in der Vorlagenausgabe aufgeführt sind.

Verwenden der AWS-Blockchainvorlage für Hyperledger Fabric

Hyperledger Fabric ist ein Blockchain-Framework, das intelligente Verträge ausführt, die als Chaincode bezeichnet werden und in Go geschrieben sind. Sie können mit Hyperledger Fabric ein privates Netzwerk erstellen und so die Anzahl der Peers einschränken, die sich mit dem Netzwerk verbinden und daran teilnehmen können. [Weitere Informationen zu Hyperledger Fabric finden Sie in der Hyperledger Fabric-Dokumentation.](#) [Weitere Informationen zu Chaincode finden Sie im Thema Chaincode für Entwickler in der Hyperledger Fabric-Dokumentation.](#)

Das AWS Blockchain Template for Hyperledger Fabric unterstützt nur eine Docker-Local-Container-Plattform, was bedeutet, dass die Hyperledger Fabric-Container auf einer einzigen EC2-Instance bereitgestellt werden.

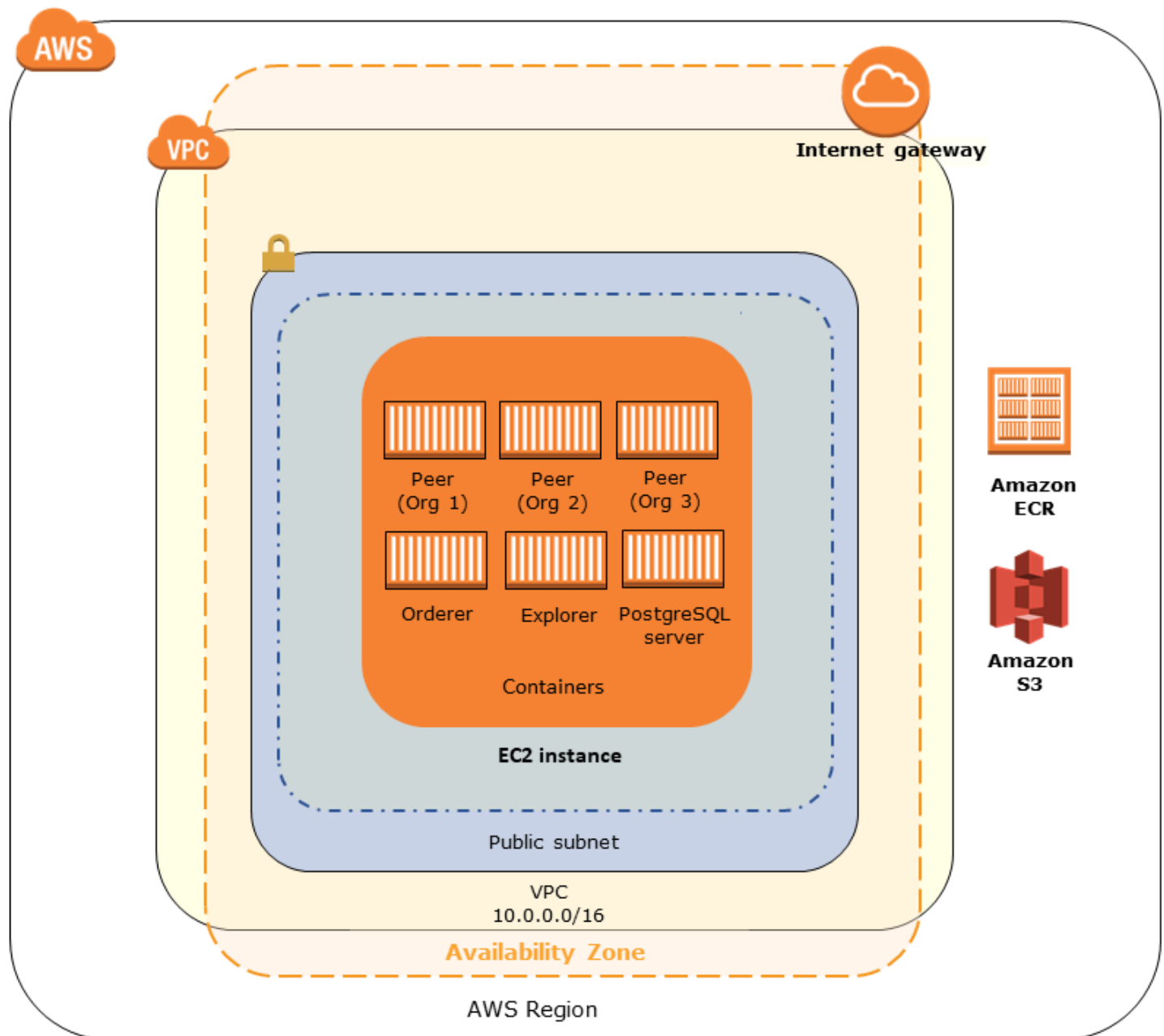
Links für den Start

Links zum Start CloudFormation in bestimmten Regionen mithilfe der Hyperledger [Fabric-Vorlagen](#) finden Sie unter [Erste Schritte mit AWS-Blockchain-Templates](#).

AWS-Blockchain-Template für Hyperledger Fabric-Komponenten

Die AWS Blockchain Template for Hyperledger Fabric erstellt eine EC2-Instance mit Docker und startet ein Hyperledger Fabric-Netzwerk mithilfe von Containern auf dieser Instance. Das Netzwerk umfasst einen Bestellservice und drei Organisationen mit jeweils einem Peer-Service. Über die Vorlage wird auch ein Hyperledger-Explorer-Container gestartet, mit dem Sie Blockchain-Daten durchsuchen können. Ein PostgreSQL-Server-Container wird gestartet, um den Hyperledger Explorer zu unterstützen.

Das folgende Diagramm zeigt ein Hyperledger Fabric-Netzwerk, das mit der Vorlage erstellt wurde:



Voraussetzungen

Bevor Sie ein Hyperledger Fabric-Netzwerk mithilfe einer Vorlage starten, stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Das von Ihnen verwendete IAM-Prinzip (Benutzer oder Gruppe) muss über die erforderlichen Berechtigungen verfügen, um mit allen zugehörigen Diensten arbeiten zu können.
- Sie müssen Zugriff auf ein Schlüsselpaar haben, mit dem Sie auf EC2-Instances zugreifen können (z. B. über SSH). Der Schlüssel muss sich in derselben Region wie die Instance befinden.

- Sie benötigen ein EC2-Instance-Profil mit einer angehängten Berechtigungsrichtlinie, die den Zugriff auf Amazon S3 und Amazon Elastic Container Registry (Amazon ECR) ermöglicht, um Container abzurufen. Ein Beispiel für eine Berechtigungsrichtlinie finden Sie unter [Beispiel für IAM-Berechtigungen für das EC2-Instance-Profil](#).
- Sie benötigen ein Amazon VPC-Netzwerk mit einem öffentlichen Subnetz oder ein privates Subnetz mit einem NAT-Gateway und einer Elastic IP-Adresse CloudFormation, damit auf Amazon S3 und Amazon ECR zugegriffen werden kann.
- Sie müssen über eine EC2-Sicherheitsgruppe mit Regeln für eingehenden Datenverkehr verfügen, die den SSH-Verkehr (Port 22) von den IP-Adressen zulassen, die eine Verbindung mit der Instance über SSH herstellen müssen, und dieselbe für Clients, die eine Verbindung zum Hyperledger Explorer (Port 8080) herstellen müssen.

Beispiel für IAM-Berechtigungen für das EC2-Instance-Profil

Sie geben einen ARN für ein EC2-Instance-Profil als einen der Parameter an, wenn Sie die AWS Blockchain Template for Hyperledger Fabric verwenden. Verwenden Sie die folgende Richtlinienanweisung als Basis für die Berechtigungsrichtlinie, die dieser EC2-Rolle und diesem Instance-Profil zugewiesen ist.

JSON

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

Verbindung zu Hyperledger Fabric-Ressourcen herstellen

Nachdem der Root-Stack, den Sie mit der Vorlage erstellen, CREATE_COMPLETE anzeigt, können Sie eine Verbindung zu Hyperledger Fabric-Ressourcen auf der EC2-Instance herstellen. Wenn Sie ein öffentliches Subnetz angegeben haben, können Sie eine Verbindung zur EC2-Instance wie bei jeder anderen EC2-Instance herstellen. Weitere Informationen finden Sie unter [Herstellen einer Verbindung zu Ihrer Linux-Instance mithilfe von SSH](#) im Amazon EC2 EC2-Benutzerhandbuch.

Wenn Sie ein privates Subnetz angegeben haben, können Sie einen Bastion-Host einrichten und verwenden, um Verbindungen zu Hyperledger Fabric-Ressourcen per Proxy herzustellen. Weitere Informationen hierzu finden Sie unter [Proxy-Verbindungen unter Verwendung eines Bastion-Hosts](#) weiter unten.

Note

Möglicherweise stellen Sie fest, dass die Vorlage der EC2-Instance, die Hyperledger Fabric-Dienste hostet, eine öffentliche IP-Adresse zuweist. Diese IP-Adresse ist jedoch nicht öffentlich zugänglich, da die Routing-Richtlinien in dem von Ihnen angegebenen privaten Subnetz keinen Verkehr zwischen dieser IP-Adresse und öffentlichen Quellen zulassen.

Proxyverbindungen, die einen Bastion-Host verwenden

Bei einigen Konfigurationen sind Hyperledger Fabric-Dienste möglicherweise nicht öffentlich verfügbar. In diesen Fällen können Sie über einen Bastion-Host eine Verbindung zu Hyperledger Fabric-Ressourcen herstellen. Weitere Informationen über Bastion-Hosts finden Sie unter [Linux-Bastion-Host-Architektur](#) im Linux Bastion-Host Quick Start-Leitfaden.

Der Bastion-Host ist eine EC2-Instance. Stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Die EC2-Instance für den Bastion-Host befindet sich in einem öffentlichen Subnetz, in dem Auto-Assign Public IP aktiviert ist und das über ein Internet-Gateway verfügt.

- Der Bastion-Host hat das key pair, das SSH-Verbindungen ermöglicht.
- Der Bastion-Host ist einer Sicherheitsgruppe zugeordnet, die eingehenden SSH-Verkehr von den Clients, die eine Verbindung herstellen, zulässt.
- Die den Hyperledger Fabric-Hosts zugewiesene Sicherheitsgruppe (z. B. der Application Load Balancer, wenn ECS die Containerplattform ist, oder die Host-EC2-Instance, wenn Docker-Local die Container-Plattform ist) lässt eingehenden Datenverkehr auf allen Ports von Quellen innerhalb der VPC zu.

Stellen Sie bei der Einrichtung eines Bastion-Hosts sicher, dass die Clients, die eine Verbindung herstellen, den Bastion-Host als Proxy verwenden. Das folgende Beispiel zeigt die Einrichtung einer Proxyverbindung unter Mac OS. *BastionIP* Ersetzen Sie durch die IP-Adresse der Bastion-Host-EC2-Instance und *MySshKey.pem* durch die Schlüsselpaardatei, die Sie auf den Bastion-Host kopiert haben.

Geben Sie in der Befehlszeile Folgendes ein:

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

Dadurch wird die Portweiterleitung für Port 9001 auf dem lokalen Computer zum Bastion-Host eingerichtet.

Als Nächstes konfigurieren Sie Ihren Browser oder Ihr System für localhost:9001 die Verwendung des SOCKS-Proxys. Beispielsweise wählen Sie unter Mac OS System Preferences (Systemeinstellungen), Network (Netzwerk), Advanced (Erweitert), SOCKS proxy (SOCKS-Proxy) und geben dann localhost:9001 ein.

Wenn Sie FoxyProxy Standard mit Chrome verwenden, wählen Sie Weitere Tools, Erweiterungen aus. Wählen Sie unter FoxyProxy Standard die Optionen Details, Erweiterungsoptionen und Neuen Proxy hinzufügen aus. Wählen Sie Manual Proxy Configuration (Manuelle Proxy-Konfiguration) aus. Geben Sie für Host or IP Address (Host- oder IP-Adresse) localhost und für Port 9001 ein. Wählen Sie SOCKS proxy?, Save (Speichern) aus.

Sie sollten jetzt in der Lage sein, eine Verbindung zu den Hyperledger Fabric-Hostadressen herzustellen, die in der Vorlagenausgabe aufgeführt sind.

Dokumentverlauf

In der folgenden Tabelle sind die Dokumentationsänderungen für dieses Handbuch beschrieben.

Letzte Aktualisierung der Dokumentation: 1. Mai 2019

Änderungen	Beschreibung	Date
Einstellung der AWS-Blockchain-Templates.	AWS Blockchain Templates wurde am 30. April 2019 eingestellt. Es werden keine weiteren Aktualisierungen dieses Services oder dieser unterstützenden Dokumentation vorgenommen. Für ein optimales Managed Blockchain-Erlebnis empfehlen wir Ihnen AWS, Amazon Managed Blockchain (AMB) zu verwenden.	1. Mai 2019
Aktualisierungen für den Bastion Host.	Geändertes Tutorial für die ersten Schritte und Ethereum-Voraussetzungen für das Hinzufügen eines Bastion-Hosts, der den Zugriff auf Webressourcen ermöglicht, die über den internen Load Balancer bereitgestellt werden, wenn die ECS-Plattform verwendet wird, und die EC2 Instanz, wenn Docker-Local verwendet wird.	3. Mai 2018
Handbuch erstellt.	Neues Entwicklerhandbuch zur Unterstützung der ersten	19. April 2018

Änderungen	Beschreibung	Date
	Version von AWS Blockchain Templates.	

AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.