



Panduan Developer

# AWS Blockchain Templates



# AWS Blockchain Templates: Panduan Developer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

---

# Table of Contents

.....	iv
Apa Itu AWS Blockchain Templates? .....	1
Cara Memulai .....	2
Saya mahir dengan dan blockchain AWS .....	2
Saya mahir AWS dan baru mengenal blockchain .....	3
Saya seorang pemula dengan AWS dan mahir dengan blockchain .....	3
Saya baru AWS dan blockchain .....	3
Layanan Terkait .....	3
Mengatur .....	5
Mendaftar ke AWS .....	5
Membuat Pengguna IAM .....	6
Membuat Pasangan Kunci .....	8
Memulai .....	10
Siapkan Prasyarat .....	11
Buat VPC dan Subnet .....	11
Buat Grup Keamanan .....	14
Membuat Peran IAM untuk Amazon ECS dan Profil Instance EC2 .....	17
Buat Host Bastion .....	22
Buat Jaringan Ethereum .....	24
Connect ke EthStats dan EthExplorer Menggunakan Bastion Host .....	27
Pembersihan Sumber Daya .....	30
Template dan Fitur AWS Blockchain .....	32
Template AWS Blockchain untuk Ethereum .....	32
Tautan ke Peluncuran .....	32
Opsi Ethereum .....	32
Prasyarat .....	36
Menghubungkan ke Sumber Daya Ethereum .....	45
Template AWS Blockchain untuk Kain Hyperledger .....	46
Tautan ke Peluncuran .....	47
Template AWS Blockchain untuk Komponen Kain Hyperledger .....	47
Prasyarat .....	48
Menghubungkan ke Sumber Daya Kain Hyperledger .....	50
Riwayat Dokumen .....	52
AWS Glosarium .....	53

AWS Blockchain Templates dihentikan pada 30 April 2019. Tidak ada pembaruan lebih lanjut untuk layanan ini atau dokumentasi pendukung ini akan dilakukan. Untuk pengalaman Blockchain Terkelola terbaik AWS, kami sarankan Anda menggunakan [Amazon Managed Blockchain \(AMB\)](#). Untuk mempelajari lebih lanjut tentang memulai dengan Amazon Managed Blockchain, lihat [lokakarya kami tentang Hyperledger Fabric](#), atau [blog kami tentang penerapan node Ethereum](#). Jika Anda memiliki pertanyaan tentang AMB atau memerlukan dukungan lebih lanjut, [hubungi Dukungan](#) atau tim AWS akun Anda.

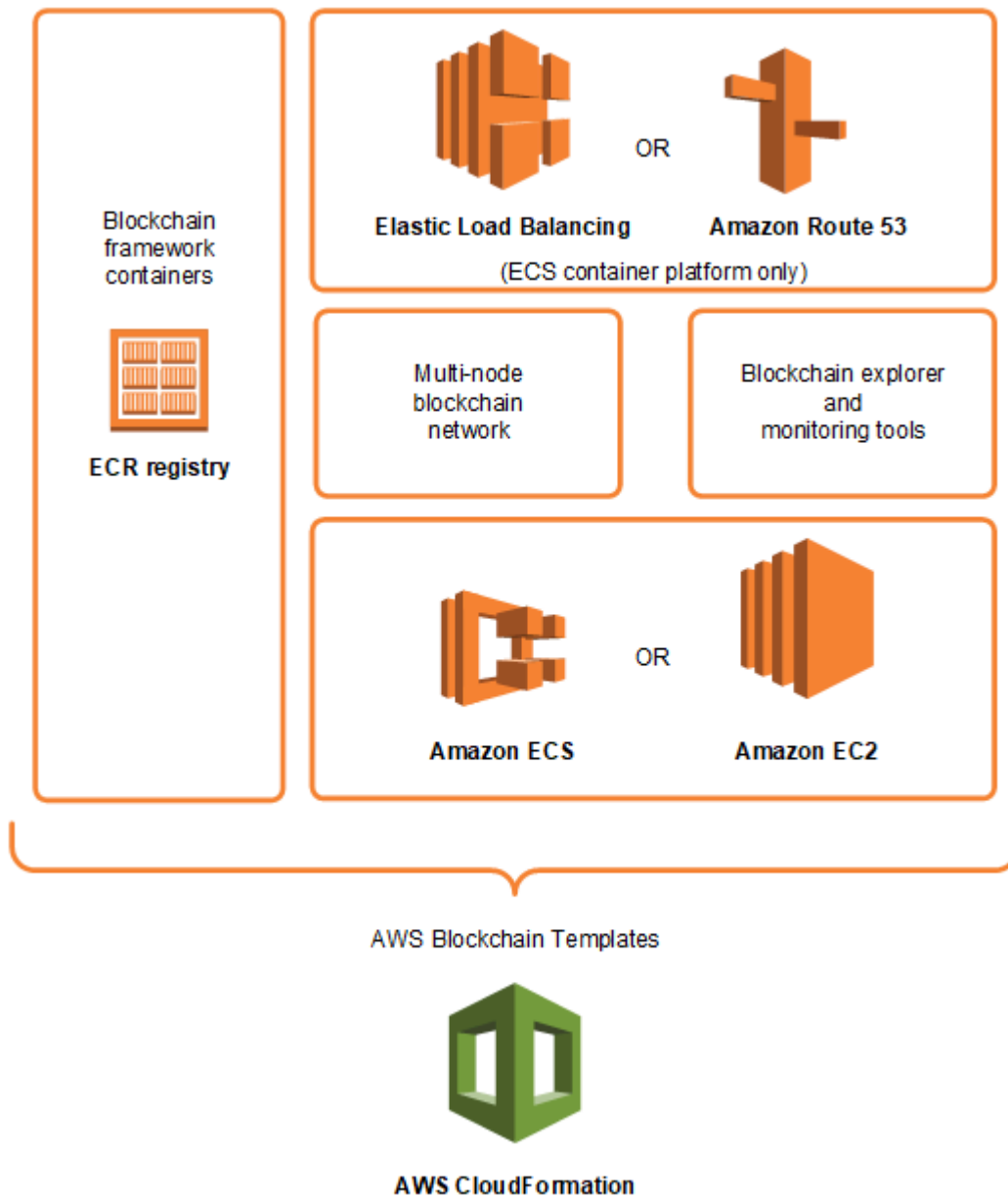
Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.

# Apa Itu AWS Blockchain Templates?

AWS Blockchain Templates membantu Anda membuat dan menyebarkan jaringan blockchain dengan cepat AWS menggunakan kerangka kerja blockchain yang berbeda. Blockchain adalah teknologi database terdesentralisasi yang mempertahankan serangkaian transaksi yang terus berkembang dan kontrak pintar yang diperkeras terhadap gangguan dan revisi menggunakan kriptografi.

Jaringan blockchain adalah peer-to-peer jaringan yang meningkatkan efisiensi dan kekekalan transaksi untuk proses bisnis seperti pembayaran internasional, manajemen rantai pasokan, pendaftaran tanah, pendanaan kerumunan, tata kelola, transaksi keuangan, dan banyak lagi. Hal ini memungkinkan orang dan organisasi yang mungkin tidak mengenal satu sama lain untuk mempercayai dan secara independen memverifikasi catatan transaksi.

Anda menggunakan AWS Blockchain Templates untuk mengonfigurasi dan meluncurkan CloudFormation tumpukan untuk membuat jaringan blockchain. AWS Sumber daya dan layanan yang Anda gunakan bergantung pada AWS Blockchain Template yang Anda pilih dan opsi yang Anda tentukan. Untuk informasi tentang template yang tersedia dan fitur-fiturnya, lihat [Template dan Fitur AWS Blockchain](#). Komponen mendasar dari jaringan blockchain yang AWS dibuat menggunakan AWS Blockchain Templates ditunjukkan pada diagram berikut.



## Cara Memulai

Tempat terbaik untuk memulai tergantung pada tingkat keahlian Anda dengan blockchain dan AWS terutama layanan yang terkait dengan AWS Blockchain Templates.

### Saya mahir dengan dan blockchain AWS

Mulailah dengan topik [Template dan Fitur AWS Blockchain](#) tentang kerangka kerja yang ingin Anda gunakan. Gunakan tautan untuk meluncurkan AWS Blockchain Template dan mengonfigurasi jaringan blockchain, atau unduh templat untuk memeriksanya sendiri.

## Saya mahir AWS dan baru mengenal blockchain

Mulailah dengan [Memulai dengan AWS Blockchain Templates](#) tutorial. Ini memandu Anda melalui pembuatan jaringan blockchain Ethereum pengantar dengan pengaturan default. Setelah selesai, lihat [Template dan Fitur AWS Blockchain](#) ikhtisar kerangka kerja dan tautan blockchain untuk mempelajari lebih lanjut tentang pilihan dan fitur konfigurasi.

## Saya seorang pemula dengan AWS dan mahir dengan blockchain

Mulailah dengan [Menyiapkan Template Blockchain AWS](#). Ini membantu Anda mengatur dasar-dasar AWS, seperti akun dan profil pengguna. Selanjutnya, jalankan [Memulai dengan AWS Blockchain Templates](#) tutorial. Tutorial ini memandu Anda melalui pembuatan jaringan blockchain Ethereum pengantar. Bahkan jika pada akhirnya Anda tidak akan menggunakan Ethereum, Anda mendapatkan pengalaman langsung dalam menyiapkan layanan terkait. Pengalaman ini berguna untuk semua kerangka kerja blockchain. Terakhir, lihat topik di [Template dan Fitur AWS Blockchain](#) bagian untuk kerangka kerja Anda.

## Saya baru AWS dan blockchain

Mulailah dengan [Menyiapkan Template Blockchain AWS](#). Ini membantu Anda mengatur dasar-dasar AWS, seperti akun dan profil pengguna. Kemudian jalankan melalui [Memulai dengan AWS Blockchain Templates](#) tutorial. Tutorial ini memandu Anda melalui pembuatan jaringan blockchain Ethereum pengantar. Luangkan waktu untuk menjelajahi tautan untuk mempelajari lebih lanjut tentang AWS layanan dan Ethereum.

## Layanan Terkait

Bergantung pada opsi yang Anda pilih, AWS Blockchain Templates dapat menggunakan AWS layanan berikut untuk menyebarkan blockchain:

- Amazon EC2 —Menyediakan kapasitas komputasi untuk jaringan blockchain Anda. Untuk informasi selengkapnya, lihat [Panduan EC2 Pengguna Amazon](#).
- Amazon ECS —Mengatur penyebaran kontainer di antara EC2 instance dalam kluster untuk jaringan blockchain Anda, jika Anda memilih untuk menggunakannya. Untuk informasi selengkapnya, silakan lihat [Panduan Developer Amazon Elastic Container Service](#).
- Amazon VPC —Menyediakan akses jaringan untuk sumber daya Ethereum yang Anda buat. Anda dapat menyesuaikan konfigurasi untuk aksesibilitas dan keamanan. Untuk informasi selengkapnya, lihat Panduan [Pengembang Amazon VPC](#).

- **Application Load Balancing** —Berfungsi sebagai titik kontak tunggal untuk akses ke antarmuka pengguna yang tersedia dan penemuan layanan internal saat menggunakan Amazon ECS sebagai platform kontainer. Untuk informasi selengkapnya, lihat [Apa itu Application Load Balancer?](#) dalam Panduan Pengguna untuk Penyeimbang Beban Aplikasi. .

# Menyiapkan Template Blockchain AWS

Sebelum memulai dengan AWS Blockchain Templates, selesaikan tugas-tugas berikut:

- [Mendaftar ke AWS](#)
- [Membuat Pengguna IAM](#)
- [Membuat Pasangan Kunci](#)

Ini adalah prasyarat mendasar untuk semua konfigurasi blockchain. Selain itu, jaringan blockchain yang Anda pilih mungkin memiliki prasyarat, yang bervariasi sesuai dengan lingkungan dan pilihan konfigurasi yang Anda inginkan. Untuk informasi lebih lanjut, lihat bagian yang relevan untuk template blockchain Anda di [Template dan Fitur AWS Blockchain](#).

Untuk step-by-step petunjuk menyiapkan prasyarat untuk jaringan Ethereum pribadi menggunakan kluster Amazon ECS, lihat. [Memulai dengan AWS Blockchain Templates](#)

## Mendaftar ke AWS

Ketika Anda mendaftar AWS, AWS akun Anda secara otomatis mendaftar untuk semua layanan. Anda hanya membayar biaya layanan yang Anda gunakan.

Jika Anda sudah memiliki AWS akun, lompat ke tugas berikutnya. Jika Anda belum memiliki akun AWS, gunakan prosedur berikut untuk membuatnya.

Untuk membuat AWS akun

1. Buka <https://portal.aws.amazon.com/billing/pendaftaran>.
2. Ikuti petunjuk online.

Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

Catat nomor AWS akun Anda. Anda membutuhkannya saat Anda membuat pengguna IAM di tugas berikutnya.

## Membuat Pengguna IAM

Layanan AWS mengharuskan Anda memberikan kredensial saat Anda mengaksesnya, sehingga layanan dapat menentukan apakah Anda memiliki izin untuk mengakses sumber dayanya. Konsol tersebut memerlukan kata sandi. Anda dapat membuat kunci akses untuk AWS akun Anda untuk mengakses antarmuka baris perintah atau API. Namun, kami tidak menyarankan Anda mengakses AWS menggunakan kredensial untuk AWS akun Anda; kami sarankan Anda menggunakan AWS Identity and Access Management (IAM) sebagai gantinya. Buat pengguna IAM, lalu tambahkan pengguna ke grup IAM dengan izin administratif atau berikan izin administratif kepada pengguna IAM yang Anda buat. Anda kemudian dapat mengakses AWS menggunakan URL khusus dan kredensial untuk pengguna IAM.

Jika Anda mendaftar AWS tetapi belum membuat pengguna IAM untuk diri Anda sendiri, Anda dapat membuatnya menggunakan konsol IAM. Jika Anda sudah memiliki pengguna IAM, Anda dapat melewati langkah ini.

Untuk membuat pengguna administrator, pilih salah satu opsi berikut.

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
Di Pusat Identitas IAM (Direkomendasikan)	Gunakan kredensial jangka pendek untuk mengakses. AWS Ini sejalan dengan praktik terbaik	Mengikuti petunjuk di <a href="#">Memulai</a> di Panduan AWS IAM Identity Center Pengguna.	Konfigurasi akses terprogram dengan <a href="#">Mengonfigurasi AWS CLI yang akan digunakan AWS IAM Identity Center</a> dalam AWS Command

Pilih salah satu cara untuk mengelola administrator Anda	Untuk	Oleh	Anda juga bisa
	keamanan. Untuk informasi tentang praktik terbaik, lihat <a href="#">Praktik terbaik keamanan di IAM</a> di Panduan Pengguna IAM.		Line Interface Panduan Pengguna.
Di IAM (Tidak direkomendasikan)	Gunakan kredensial jangka panjang untuk mengakses. AWS	Mengikuti petunjuk di <a href="#">Buat pengguna IAM untuk akses darurat</a> di Panduan Pengguna IAM.	Konfigurasi akses terprogram dengan <a href="#">Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM</a> .

Untuk masuk sebagai pengguna IAM baru ini, keluar dari, lalu gunakan URL berikut Konsol Manajemen AWS, di mana `your_aws_account_id` adalah nomor AWS akun Anda tanpa tanda hubung (misalnya, jika nomor akun Anda, ID akun Anda adalah): AWS 1234-5678-9012 AWS 123456789012

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

Masukkan nama pengguna dan kata sandi IAM yang baru saja Anda buat. Saat Anda masuk, bilah navigasi menampilkan "nama\_pengguna\_Anda @ id\_akun\_aws\_Anda".

Jika Anda tidak ingin URL untuk halaman login Anda berisi ID AWS akun Anda, Anda dapat membuat alias akun. Dari dasbor IAM, pilih Create Account Alias (Buat Alias Akun) dan masukkan alias, misalnya nama perusahaan Anda. Untuk masuk setelah membuat alias akun, gunakan URL berikut:

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

Untuk memverifikasi tautan masuk pengguna IAM untuk akun Anda, buka konsol IAM dan periksa di Tautan masuk pengguna IAM di dasbor.

Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS Identity and Access Management](#).


## Membuat Pasangan Kunci

AWS menggunakan kriptografi kunci publik untuk mengamankan informasi login untuk instance dalam jaringan blockchain. Anda menentukan nama key pair saat Anda menggunakan setiap AWS Blockchain Template. Anda kemudian dapat menggunakan key pair untuk mengakses instance secara langsung, misalnya, untuk masuk menggunakan SSH.

Jika Anda sudah memiliki key pair di Region yang tepat, Anda dapat melewati langkah ini. Jika Anda belum membuat key pair, Anda dapat membuatnya menggunakan EC2 konsol Amazon. Buat key pair di Wilayah yang sama yang Anda gunakan untuk meluncurkan jaringan Ethereum. Untuk informasi selengkapnya, lihat [Wilayah dan Zona Ketersediaan](#) di Panduan EC2 Pengguna Amazon.

Untuk membuat pasangan kunci

1. Buka EC2 konsol Amazon di <https://console.aws.amazon.com/ec2/>.
2. Di bilah navigasi, pilih Wilayah untuk pasangan kunci. Anda dapat memilih Wilayah mana pun yang tersedia untuk Anda, terlepas dari lokasi Anda, tetapi pasangan kunci khusus untuk Wilayah. Misalnya, jika Anda berencana untuk meluncurkan instance di wilayah AS Timur (Ohio), Anda harus membuat key pair untuk instance di Wilayah yang sama.
3. Di panel navigasi, pilih Key Pairs (Pasangan Kunci), Create Key Pair (Buat Pasangan Kunci).
4. Untuk nama pasangan Kunci, masukkan nama untuk key pair baru. Pilih nama yang mudah diingat, seperti nama pengguna IAM Anda, diikuti oleh `-key-pair`, ditambah nama wilayah. Misalnya, `me-key-pair-useast2`. Pilih Buat.
5. File kunci privat tersebut akan secara otomatis diunduh oleh peramban Anda. Nama file dasar adalah nama yang Anda tentukan sebagai nama key pair Anda, dan ekstensi nama file adalah `.pem`. Simpan file kunci privat di suatu tempat yang aman.

 Important

Ini adalah satu-satunya kesempatan Anda untuk menyimpan file kunci privat tersebut. Anda memberikan nama key pair Anda saat meluncurkan jaringan Ethereum.

Untuk informasi selengkapnya, lihat [Pasangan EC2 Kunci Amazon](#) di Panduan EC2 Pengguna Amazon. Untuk informasi selengkapnya tentang menghubungkan ke EC2 instance menggunakan key pair, lihat [Connect to Your Linux Instance](#) di Panduan EC2 Pengguna Amazon.

# Memulai dengan AWS Blockchain Templates

Tutorial ini menunjukkan cara menggunakan AWS Blockchain Template untuk Ethereum untuk membuat jaringan blockchain pribadi di AWS through CloudFormation. Jaringan yang Anda buat memiliki dua klien Ethereum dan satu penambang yang berjalan di EC2 instans Amazon di cluster Amazon ECS. Amazon ECS menjalankan layanan ini dalam wadah Docker yang ditarik dari Amazon ECR. Sebelum Anda memulai tutorial ini, akan sangat membantu untuk mengetahui tentang jaringan blockchain dan AWS layanan yang terlibat, tetapi tidak diperlukan.

Tutorial ini mengasumsikan bahwa Anda telah menyiapkan prasyarat umum yang tercakup dalam [Menyiapkan Template Blockchain AWS](#). Selain itu, Anda harus menyiapkan beberapa AWS sumber daya, seperti jaringan VPC Amazon dan izin khusus untuk peran IAM, sebelum Anda menggunakan templat.

Tutorial menunjukkan cara mengatur prasyarat tersebut. Kami membuat pilihan pengaturan, tetapi mereka tidak preskriptif. Selama Anda memenuhi prasyarat, Anda dapat membuat pilihan konfigurasi lain berdasarkan kebutuhan aplikasi dan lingkungan Anda. Untuk informasi tentang fitur dan prasyarat umum untuk setiap templat, dan untuk mengunduh templat atau meluncurkannya langsung, lihat [CloudFormation Template dan Fitur AWS Blockchain](#).

Sepanjang tutorial ini, contoh menggunakan Wilayah AS Barat (Oregon) (us-west-2), tetapi Anda dapat menggunakan wilayah apa pun yang mendukung AWS Blockchain Templates:

- Wilayah AS Barat (Oregon) (us-west-2)
- Wilayah AS Timur (Virginia N.) (us-east-1)
- Wilayah AS Timur (Ohio) (us-east-2)

## Note

Menjalankan template di Wilayah yang tidak tercantum di atas meluncurkan sumber daya di Wilayah AS Timur (Virginia N.) (us-east-1).

Template AWS Blockchain untuk Ethereum yang Anda konfigurasi menggunakan tutorial ini menciptakan sumber daya berikut:

- EC2 Instance On-Demand dari jenis dan nomor yang Anda tentukan. Tutorial menggunakan tipe instance t2.medium default.
- Application Load Balancer internal.

Mengikuti tutorial, langkah-langkah disediakan untuk membersihkan sumber daya yang Anda buat.

Topik

- [Siapkan Prasyarat](#)
- [Buat Jaringan Ethereum](#)
- [Connect ke EthStats dan EthExplorer Menggunakan Bastion Host](#)
- [Pembersihan Sumber Daya](#)

## Siapkan Prasyarat

Template AWS Blockchain untuk konfigurasi Ethereum yang Anda tentukan dalam tutorial ini mengharuskan Anda melakukan hal berikut:

- [Buat VPC dan Subnet](#)
- [Buat Grup Keamanan](#)
- [Membuat Peran IAM untuk Amazon ECS dan Profil Instance EC2](#)
- [Buat Host Bastion](#)

## Buat VPC dan Subnet

Template AWS Blockchain untuk Ethereum meluncurkan sumber daya ke jaringan virtual yang Anda tentukan menggunakan Amazon Virtual Private Cloud (Amazon VPC). Konfigurasi yang Anda tentukan dalam tutorial ini menciptakan Application Load Balancer, yang membutuhkan dua subnet publik di Availability Zone yang berbeda. Selain itu, subnet pribadi diperlukan untuk instance container, dan subnet harus berada di Availability Zone yang sama dengan Application Load Balancer. Anda pertama kali menggunakan VPC Wizard untuk membuat satu subnet publik dan subnet pribadi di Availability Zone yang sama. Anda kemudian membuat subnet publik kedua dalam VPC ini di Availability Zone yang berbeda.

Untuk informasi lebih lanjut, lihat [Apa itu Amazon VPC?](#) di Panduan Pengguna Amazon VPC.

Gunakan konsol Amazon VPC (<https://console.aws.amazon.com/vpc/>) untuk membuat alamat IP Elastis, VPC, dan subnet seperti yang dijelaskan di bawah ini.

Untuk membuat Alamat IP elastis

1. Buka konsol VPC Amazon di <https://console.aws.amazon.com/vpc/>
2. Pilih elastis IPs, alokasikan alamat baru, alokasikan.
3. Catat alamat IP Elastis yang Anda buat dan pilih Tutup.
4. Dalam daftar alamat IP elastis, temukan ID Alokasi untuk alamat IP elastis yang dibuat sebelumnya. Anda menggunakan ini saat Anda membuat VPC.

Untuk membuat VPC

1. Dari bilah navigasi, pilih Wilayah untuk VPC. VPCs khusus untuk Wilayah, jadi pilih Wilayah yang sama tempat Anda membuat key pair dan tempat Anda meluncurkan tumpukan Ethereum. Untuk informasi selengkapnya, lihat [Membuat Pasangan Kunci](#).
2. Di dasbor VPC, pilih Start VPC Wizard (Mulai Wizard VPC).
3. Pada Langkah 1: Pilih halaman Konfigurasi VPC, pilih VPC dengan Subnet Publik dan Pribadi, Pilih.
4. Pada Langkah 2: VPC dengan halaman Subnet Publik dan Pribadi, biarkan blok CIDR dan blok IPv4 CIDR ke nilai defaultnya IPv6 . Untuk nama VPC, masukkan nama yang ramah.
5. Untuk IPv4 CIDR subnet Public, tinggalkan nilai default. Untuk Availability Zone, pilih zona. Untuk nama subnet Publik, masukkan nama yang ramah.

Anda menentukan subnet ini sebagai salah satu dari dua subnet pertama untuk Application Load Balancer ketika Anda menggunakan template.

Perhatikan Availability Zone subnet ini karena Anda memilih Availability Zone yang sama untuk subnet pribadi, dan yang berbeda untuk subnet publik lainnya.

6. Untuk IPv4 CIDR subnet Private, tinggalkan nilai default. Untuk Availability Zone, pilih Availability Zone yang sama seperti pada langkah sebelumnya. Untuk nama subnet pribadi, masukkan nama yang ramah.
7. Untuk ID Alokasi IP Elastis, pilih alamat IP Elastis yang Anda buat sebelumnya.
8. Tinggalkan nilai default untuk pengaturan lain.
9. Pilih Buat VPC.

Contoh di bawah ini menunjukkan VPC EthereumNetworkVPC dengan subnet publik EthereumPubSub1 dan subnet pribadi 1. EthereumPvtSub Subnet publik menggunakan 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

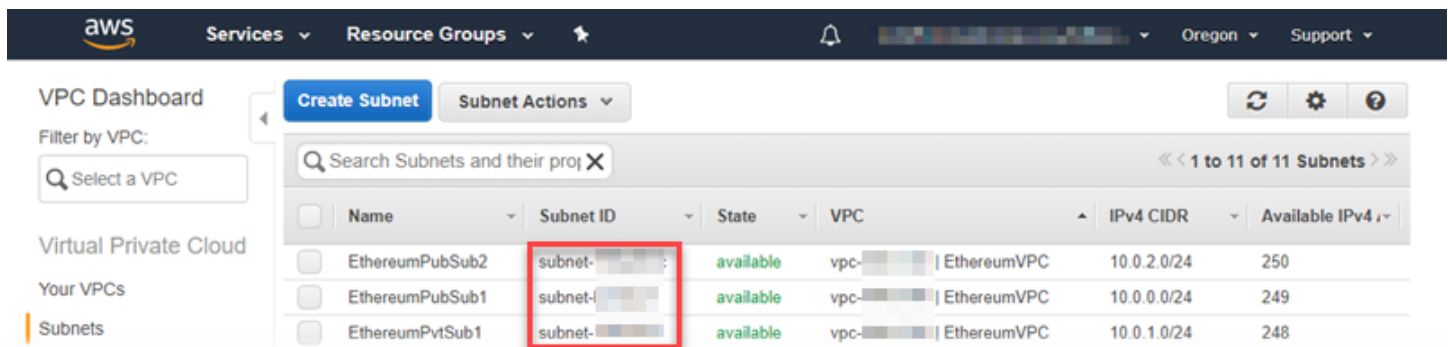
**Hardware tenancy:**\*  ▼

---

Untuk membuat subnet publik kedua di Availability Zone yang berbeda

1. Pilih Subnet dan kemudian pilih subnet publik yang Anda buat sebelumnya dari daftar. Pilih tab Route Table dan perhatikan ID tabel Route. Anda menentukan tabel rute yang sama untuk subnet publik kedua di bawah ini.
2. Pilih Buat Subnet
3. Untuk Tanda nama, masukkan nama untuk subnet. Anda menggunakan nama ini nanti ketika Anda membuat host bastion di jaringan ini.
4. Untuk VPC, pilih VPC yang Anda buat sebelumnya.
5. Untuk Availability Zone, pilih zona berbeda dari zona yang Anda pilih untuk subnet publik pertama.
6. Untuk blok IPv4 CIDR, masukkan 10.0.2.0/24.
7. Pilih Ya, Buat. Subnet ditambahkan ke daftar subnet.
8. Dengan subnet yang dipilih dari daftar, pilih Tindakan Subnet, Ubah pengaturan IP penetapan otomatis. Pilih Tetapkan otomatis IPs, Simpan, Tutup. Ini memungkinkan host bastion untuk mendapatkan alamat IP publik saat Anda membuatnya di subnet ini.
9. Pada tab Tabel Rute, pilih Edit. Untuk Ubah ke, pilih ID tabel rute yang Anda catat sebelumnya dan pilih Simpan.

Anda sekarang akan melihat tiga subnet untuk VPC yang Anda buat sebelumnya. Catat nama subnet dan IDs agar Anda dapat menentukannya menggunakan templat.



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

## Buat Grup Keamanan

Kelompok keamanan bertindak sebagai firewall, mengendalikan lalu lintas masuk dan keluar ke sumber daya. Saat Anda menggunakan template untuk membuat jaringan Ethererum di klaster Amazon ECS, Anda menentukan dua grup keamanan:

- Grup keamanan untuk EC2 instance yang mengontrol lalu lintas ke dan dari EC2 instance di klaster
- Grup keamanan untuk Application Load Balancer yang mengontrol lalu lintas antara Application Load Balancer EC2 , instance, dan host bastion. Anda mengaitkan grup keamanan ini dengan host benteng juga.

Setiap grup keamanan memiliki aturan yang memungkinkan komunikasi antara Application Load Balancer dan EC2 instans, serta aturan minimum lainnya. Ini mengharuskan kelompok keamanan saling merujuk. Untuk alasan ini, pertama-tama Anda membuat grup keamanan dan kemudian memperbaruinya dengan aturan yang sesuai.

Untuk membuat dua grup keamanan

1. Buka EC2 konsol Amazon di <https://console.aws.amazon.com/ec2/>.
2. Di panel navigasi, pilih Grup Keamanan, Buat Grup Keamanan.
3. Untuk nama grup Keamanan, masukkan nama untuk grup keamanan yang mudah diidentifikasi dan akan membedakannya dari yang lain, seperti Ethereum EC2 -SG atau EthereumAlb-SG. Anda menggunakan nama-nama ini nanti. Untuk Deskripsi, masukkan ringkasan singkat.
4. Untuk VPC, pilih VPC yang Anda buat sebelumnya.
5. Pilih Buat.
6. Ulangi langkah-langkah di atas untuk membuat grup keamanan lainnya.

Menambahkan aturan masuk ke grup keamanan untuk instance EC2

1. Pilih grup keamanan untuk EC2 instance yang Anda buat sebelumnya
2. Pada tab Ke dalam, pilih Edit.
3. Untuk Tipe, pilih Semua lalu lintas. Untuk Sumber, biarkan Kustom dipilih, lalu pilih grup keamanan yang sedang Anda edit dari daftar, misalnya, Ethereum EC2 -SG. Hal ini memungkinkan EC2 contoh dalam kelompok keamanan untuk berkomunikasi satu sama lain.
4. Pilih Tambahkan aturan.
5. Untuk Tipe, pilih Semua lalu lintas. Untuk Sumber, biarkan Kustom dipilih, lalu pilih grup keamanan untuk Application Load Balancer dari daftar, misalnya, EthereumAlb-SG. Hal ini memungkinkan EC2 instance dalam grup keamanan untuk berkomunikasi dengan Application Load Balancer.
6. Pilih Simpan.

## Menambahkan aturan masuk dan mengedit keluar untuk grup keamanan untuk Application Load Balancer

1. Pilih grup keamanan untuk Application Load Balancers yang Anda buat sebelumnya
2. Pada tab Inbound, pilih Edit lalu tambahkan aturan masuk berikut:
  - a. Untuk Tipe, pilih Semua lalu lintas. Untuk Sumber, biarkan Kustom dipilih, lalu pilih grup keamanan yang sedang Anda edit dari daftar, misalnya, EthereumAlb-SG. Hal ini memungkinkan Application Load Balancer untuk berkomunikasi dengan dirinya sendiri dan dengan host bastion.
  - b. Pilih Tambahkan aturan.
  - c. Untuk Tipe, pilih Semua lalu lintas. Untuk Sumber, biarkan Kustom dipilih, lalu pilih grup keamanan untuk EC2 instance dari daftar, misalnya, Ethereum EC2 -SG. Hal ini memungkinkan EC2 instance dalam grup keamanan untuk berkomunikasi dengan Application Load Balancer dan host bastion.
  - d. Pilih Add Rule (Tambahkan Aturan).
  - e. Untuk Tipe, pilih SSH. Untuk Sumber, pilih IP Saya, yang mendeteksi CIDR IP komputer Anda dan memasukkannya.

### Important

Aturan ini memungkinkan host bastion untuk menerima lalu lintas SSH dari komputer Anda, memungkinkan komputer Anda menggunakan host bastion untuk melihat antarmuka web dan terhubung ke EC2 instance di jaringan Ethereum. Untuk memungkinkan orang lain terhubung ke jaringan Ethereum, tambahkan mereka sebagai sumber ke aturan ini. Hanya izinkan lalu lintas masuk ke sumber tepercaya.

- f. Pilih Simpan.
3. Pada tab Keluar, pilih Edit dan hapus aturan yang dibuat secara otomatis untuk mengizinkan lalu lintas keluar ke semua alamat IP.
4. Pilih Tambahkan aturan.
5. Untuk Tipe, pilih Semua lalu lintas. Untuk Tujuan, biarkan Kustom dipilih, lalu pilih grup keamanan untuk EC2 instance dari daftar. Hal ini memungkinkan koneksi keluar dari Application Load Balancer dan host bastion EC2 ke instance di jaringan Ethereum.
6. Pilih Tambahkan aturan.

7. Untuk Tipe, pilih Semua lalu lintas. Untuk Tujuan, biarkan Kustom dipilih, lalu pilih grup keamanan yang sedang Anda edit dari daftar, misalnya, EthereumAlb-SG. Hal ini memungkinkan Application Load Balancer untuk berkomunikasi dengan dirinya sendiri dan dengan host bastion.
8. Pilih Simpan.

## Membuat Peran IAM untuk Amazon ECS dan Profil Instance EC2

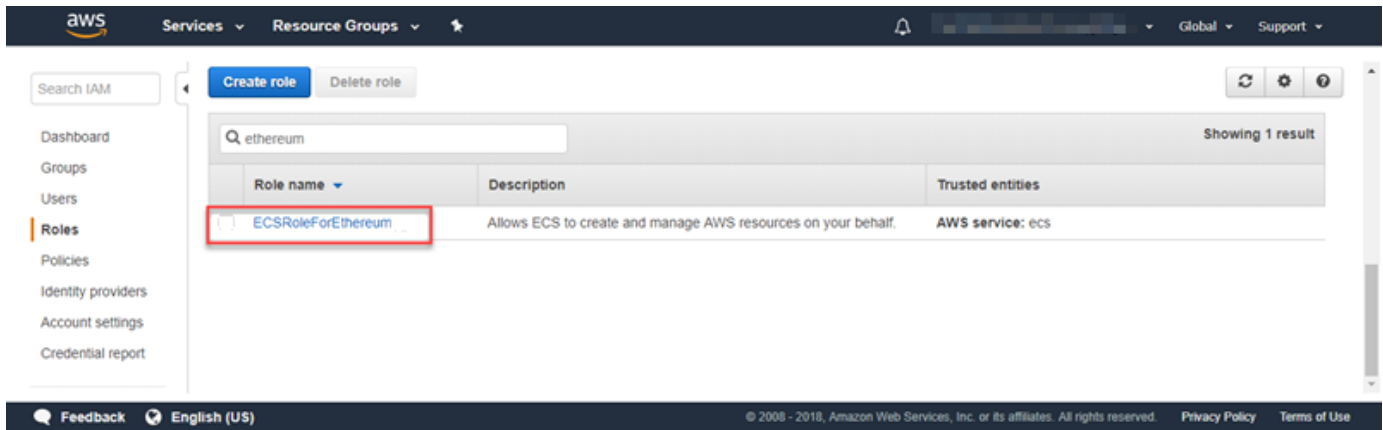
Saat Anda menggunakan template ini, Anda menentukan peran IAM untuk Amazon ECS dan profil EC2 instans. Kebijakan izin yang dilampirkan pada peran ini memungkinkan sumber daya AWS dan instance di klaster Anda berinteraksi dengan sumber daya AWS lainnya. Untuk informasi lebih lanjut, lihat [Peran IAM](#) dalam Panduan Pengguna IAM. Anda mengatur peran IAM untuk Amazon ECS dan profil EC2 instans menggunakan konsol IAM (). <https://console.aws.amazon.com/iam/>

Untuk membuat peran IAM untuk Amazon ECS

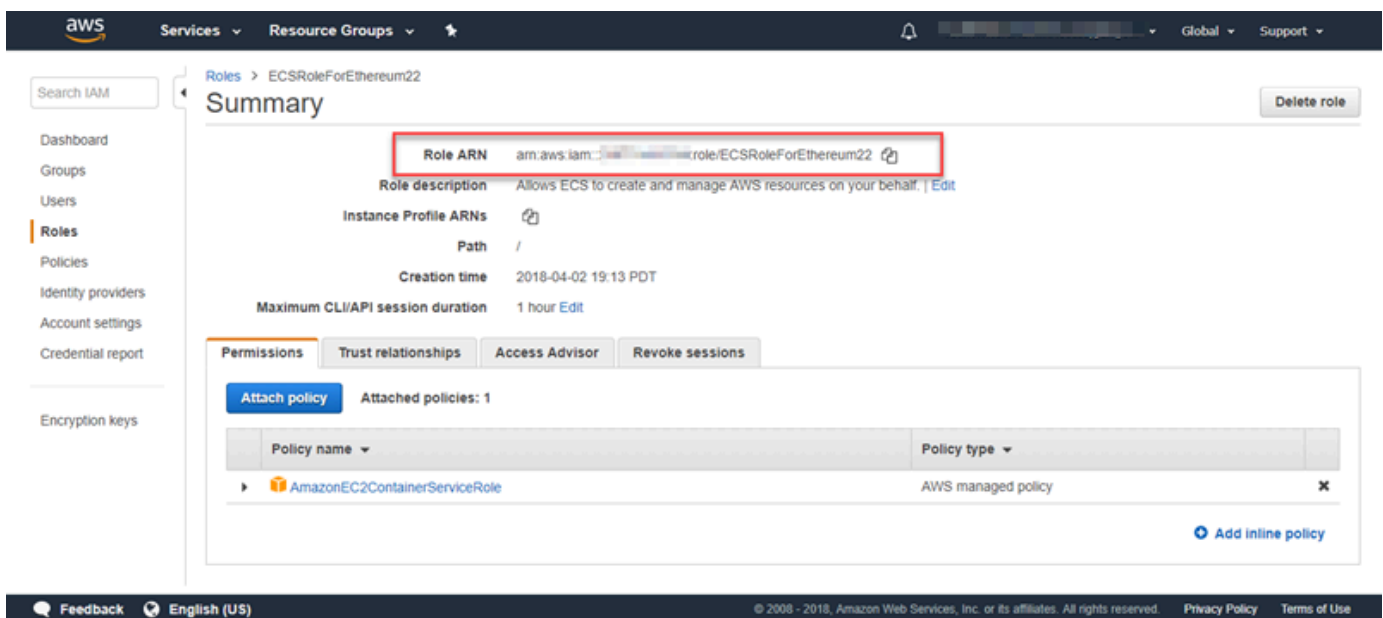
1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Dalam panel navigasi, pilih Roles (Peran), Create role (Buat Peran).
3. Di bawah Pilih jenis entitas tepercaya, pilih Layanan AWS.
4. Untuk Choose the service that will use this role (Pilih layanan yang akan menggunakan peran ini), pilih Elastic Container Service.
5. Di bawah Pilih kasus penggunaan Anda, pilih Layanan Kontainer Elastis, Berikutnya: Izin.

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. Untuk kebijakan Izin, biarkan kebijakan default (Amazon EC2 ContainerServiceRole) dipilih, dan pilih Next:Review.
7. Untuk nama Peran, masukkan nilai yang membantu Anda mengidentifikasi peran, seperti ECSRoleForEthereum. Untuk Deskripsi Peran, masukkan ringkasan singkat. Perhatikan nama peran untuk nanti.
8. Pilih Buat peran.
9. Pilih peran yang baru saja Anda buat dari daftar. Jika akun Anda memiliki banyak peran, Anda dapat mencari nama peran.



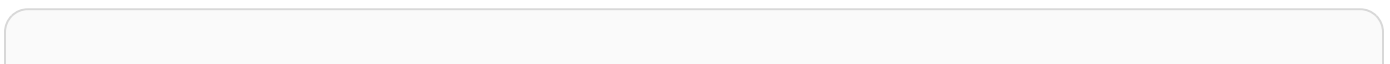
10. Salin nilai ARN Peran dan simpan sehingga Anda dapat menyalinnya lagi. Anda memerlukan ARN ini saat membuat jaringan Ethereum.



Profil EC2 instans yang Anda tentukan dalam templat diasumsikan oleh EC2 instance di jaringan Ethereum untuk berinteraksi dengan AWS layanan lain. Anda membuat kebijakan izin untuk peran, membuat peran (yang secara otomatis membuat profil instance dengan nama yang sama), lalu melampirkan kebijakan izin ke peran tersebut.

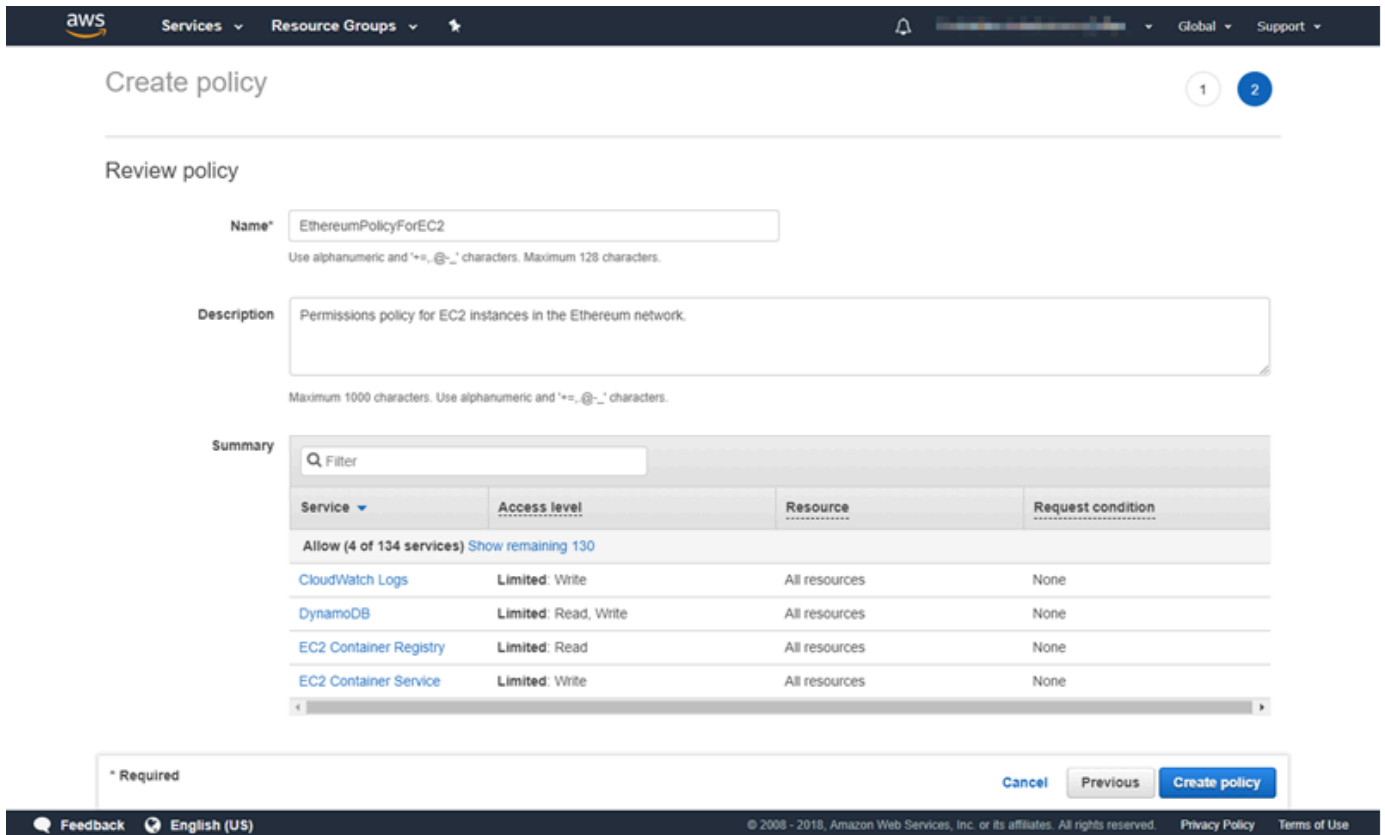
Untuk membuat profil EC2 instance

1. Di panel navigasi, pilih Kebijakan, Buat kebijakan.
2. Pilih JSON dan ganti pernyataan kebijakan default dengan kebijakan JSON berikut:

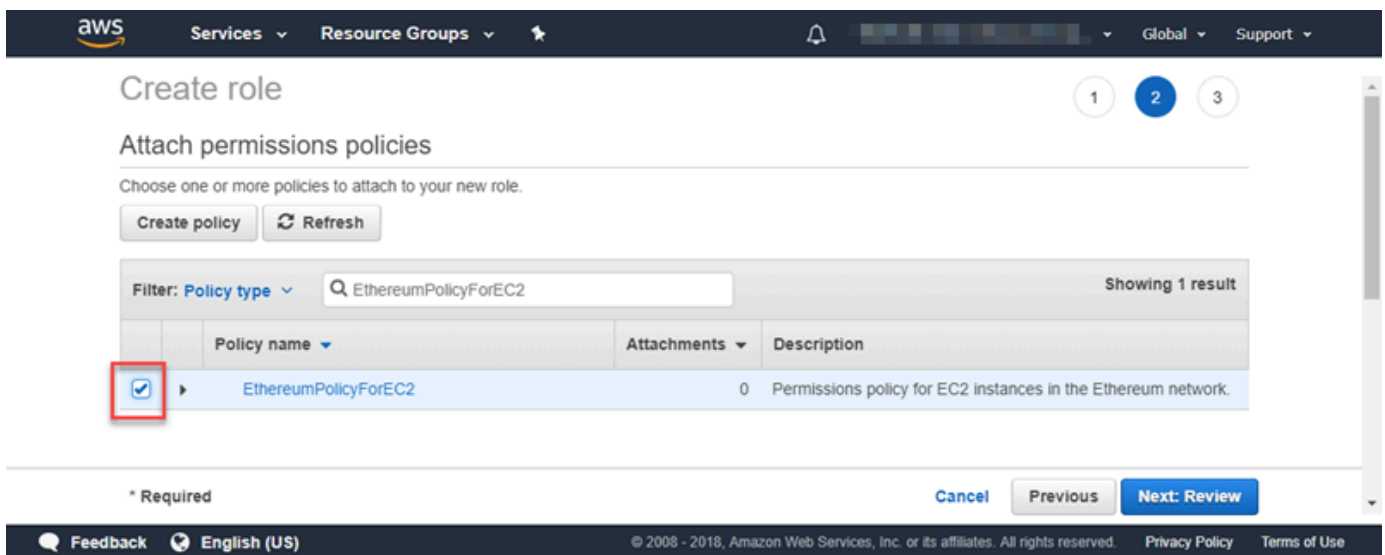


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:DeregisterContainerInstance",
        "ecs:DiscoverPollEndpoint",
        "ecs:Poll",
        "ecs:RegisterContainerInstance",
        "ecs:StartTelemetrySession",
        "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": "*"
    }
  ]
}
```

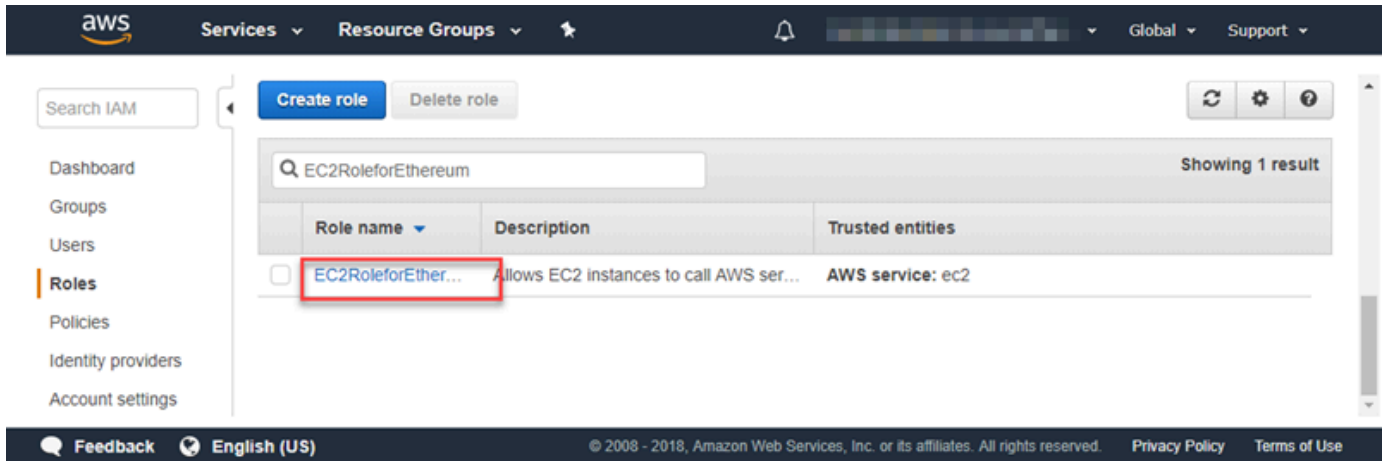
3. Pilih Tinjau kebijakan.
4. Untuk Nama, masukkan nilai yang membantu Anda mengidentifikasi kebijakan izin ini, misalnya EthereumPolicyForEC2. Untuk Deskripsi, masukkan ringkasan singkat. Pilih Buat kebijakan.



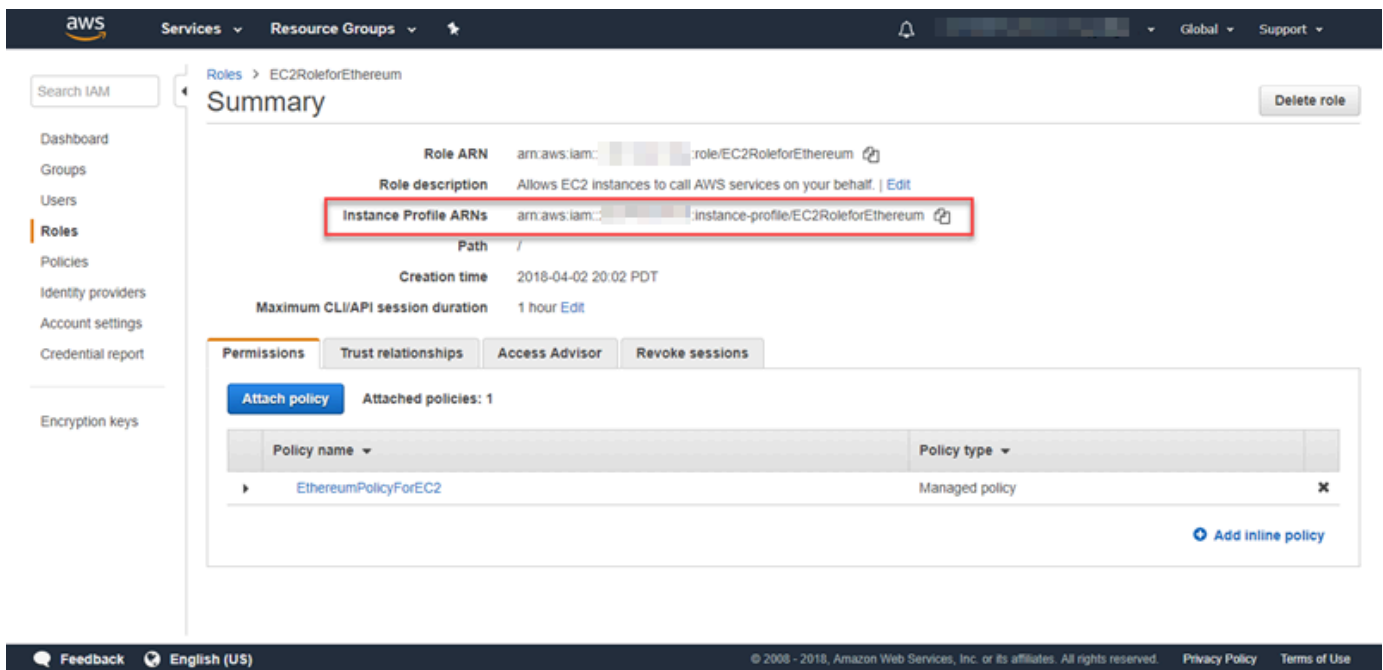
5. Pilih Peran, Buat peran.
6. Pilih EC2, Berikutnya: Izin.
7. Di bidang Pencarian, masukkan nama kebijakan izin yang Anda buat sebelumnya, misalnya EthereumPolicyForEC2.
8. Pilih tanda centang untuk kebijakan yang Anda buat sebelumnya, dan pilih Berikutnya: Tinjau.



9. Untuk nama Peran, masukkan nilai yang membantu Anda mengidentifikasi peran, misalnya EC2RoleForEthereum. Untuk deskripsi peran, masukkan ringkasan singkat. Pilih Buat peran.
10. Pilih peran yang baru saja Anda buat dari daftar. Jika akun Anda memiliki banyak peran, Anda dapat memasukkan nama peran di kolom Pencarian.



11. Salin nilai ARN Profil Instance dan simpan sehingga Anda dapat menyalinnya lagi. Anda memerlukan ARN ini saat membuat jaringan Ethereum.



## Buat Host Bastion

Dalam tutorial ini, Anda membuat host benteng. Ini adalah EC2 contoh yang Anda gunakan untuk terhubung ke antarmuka web dan instance di jaringan Ethereum Anda. Tujuan utamanya

adalah untuk meneruskan lalu lintas SSH dari klien tepercaya di luar VPC sehingga mereka dapat mengakses sumber daya jaringan Ethereum.

Anda mengatur host bastion karena Application Load Balancer yang dibuat template bersifat internal, artinya hanya merutekan alamat IP internal. Tuan rumah benteng:

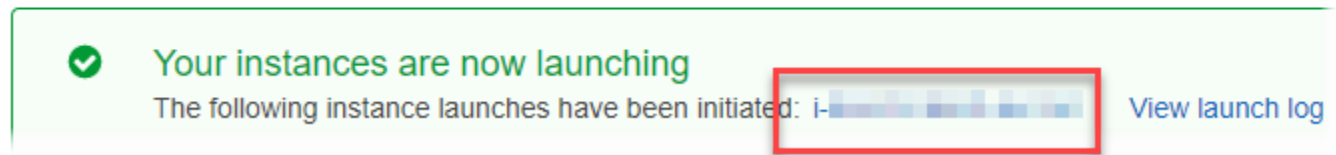
- Memiliki alamat IP internal yang dikenali Application Load Balancer karena Anda meluncurkannya di subnet publik kedua yang Anda buat sebelumnya.
- Memiliki alamat IP publik yang diberikan subnet, yang dapat diakses oleh sumber tepercaya di luar VPC.
- Terkait dengan grup keamanan untuk Application Load Balancer yang Anda buat sebelumnya, yang memiliki aturan masuk yang memungkinkan lalu lintas SSH (port 22) dari klien tepercaya.

Untuk dapat mengakses jaringan Ethereum, klien tepercaya perlu diatur untuk terhubung melalui host benteng. Untuk informasi selengkapnya, lihat [Connect ke EthStats dan EthExplorer Menggunakan Bastion Host](#). Tuan rumah benteng adalah salah satu pendekatan. Anda dapat menggunakan pendekatan apa pun yang menyediakan akses dari klien tepercaya ke sumber daya pribadi dalam VPC.

Untuk membuat host benteng

1. Ikuti lima langkah pertama untuk [Meluncurkan Instance](#) di Panduan EC2 Pengguna Amazon.
2. Pilih Edit Detail Instance. Untuk Jaringan, pilih VPC yang Anda buat sebelumnya, untuk Subnet pilih subnet publik kedua yang Anda buat sebelumnya. Biarkan semua pengaturan lainnya ke defaultnya.
3. Konfirmasikan perubahan saat diminta, lalu pilih Tinjau dan Luncurkan.
4. Pilih Edit Grup Keamanan. Untuk Tetapkan grup keamanan, pilih Pilih grup keamanan yang sudah ada.
5. Dari daftar grup keamanan, pilih grup keamanan untuk Application Load Balancer yang Anda buat sebelumnya, lalu pilih Tinjau dan Luncurkan.
6. Pilih Luncurkan.
7. Perhatikan ID instance. Anda membutuhkannya nanti ketika Anda [Connect ke EthStats dan EthExplorer Menggunakan Bastion Host](#).

## Launch Status



## Buat Jaringan Ethereum

Jaringan Ethereum yang Anda tentukan menggunakan templat dalam topik ini meluncurkan CloudFormation tumpukan yang membuat kluster EC2 instans Amazon ECS untuk jaringan Ethereum. Template bergantung pada sumber daya yang Anda buat sebelumnya. [Siapkan Prasyarat](#)

Saat Anda meluncurkan CloudFormation tumpukan menggunakan template, itu membuat tumpukan bersarang untuk beberapa tugas. Setelah selesai, Anda dapat terhubung ke sumber daya yang dilayani oleh Application Load Balancer jaringan melalui host bastion untuk memverifikasi bahwa jaringan Ethereum Anda berjalan dan dapat diakses.

Untuk membuat jaringan Ethereum menggunakan AWS Blockchain Template untuk Ethereum

1. Lihat [Memulai dengan AWS Blockchain Templates](#), dan buka Template AWS Blockchain terbaru untuk Ethereum di CloudFormation konsol menggunakan tautan cepat untuk Wilayah AWS Anda.
2. Masukkan nilai sesuai dengan pedoman berikut:
  - Untuk nama Stack, masukkan nama yang mudah untuk Anda identifikasi. Nama ini digunakan dalam nama sumber daya yang dibuat oleh tumpukan.
  - Di bawah Parameter Jaringan Ethereum dan Parameter Jaringan Ethereum Pribadi, tinggalkan pengaturan default.

### ⚠ Warning

Gunakan akun default dan frase mnemonik terkait hanya untuk tujuan pengujian. Jangan mengirim Eter asli menggunakan kumpulan akun default karena siapa pun yang memiliki akses ke frase mnemonik dapat mengakses atau mencuri Eter dari akun. Sebagai gantinya, tentukan akun khusus untuk tujuan produksi. Frase mnemonik

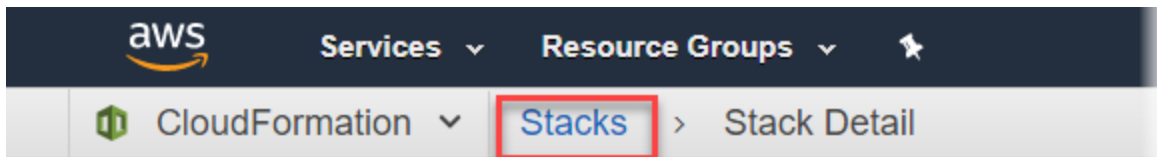
yang terkait dengan akun default adalah. outdoor father modify clever trophy abandon vital feel portion grit evolve twist

- Di bawah Konfigurasi Platform, tinggalkan pengaturan default, yang membuat kluster EC2 instans Amazon ECS. Alternatifnya, docker-local membuat jaringan Ethereum menggunakan satu instance. EC2
- Di bawah EC2 konfigurasi, pilih opsi sesuai dengan pedoman berikut:
  - Untuk EC2 Key Pair, pilih key pair. Untuk informasi tentang membuat key pair, lihat [Membuat Pasangan Kunci](#).
  - Untuk Grup EC2 Keamanan, pilih grup keamanan yang Anda buat sebelumnya [Buat Grup Keamanan](#).
  - Untuk EC2 Instance Profile ARN, masukkan ARN dari profil instance yang Anda buat sebelumnya. [Membuat Peran IAM untuk Amazon ECS dan Profil Instance EC2](#)
- Di bawah konfigurasi jaringan VPC, pilih opsi sesuai dengan pedoman berikut:
  - Untuk ID VPC, pilih VPC yang Anda buat sebelumnya. [Buat VPC dan Subnet](#)
  - Untuk Ethereum Network Subnet IDs, pilih subnet pribadi tunggal yang Anda buat sebelumnya dalam prosedur. [To create the VPC](#)
- Di bawah konfigurasi cluster ECS, biarkan default. Ini menciptakan cluster ECS dari tiga EC2 contoh.
- Di bawah konfigurasi Application Load Balancer (khusus ECS), pilih opsi sesuai dengan pedoman berikut:
  - Untuk Application Load Balancer Subnet IDs, pilih dua subnet publik dari [list of subnets](#) yang Anda sebutkan sebelumnya.
  - Untuk Grup Keamanan Application Load Balancer, pilih grup keamanan untuk Application Load Balancer yang Anda buat sebelumnya. [Buat Grup Keamanan](#)
  - Untuk Peran IAM, masukkan ARN dari peran ECS yang Anda buat sebelumnya. [Membuat Peran IAM untuk Amazon ECS dan Profil Instance EC2](#)
- Di bawah EthStats, pilih opsi sesuai dengan pedoman berikut:
  - Untuk Deploy EthStats, tinggalkan pengaturan default, yang benar.
  - Untuk EthStats Connection Secret, ketikkan nilai arbitrer yang setidaknya enam karakter.
- Di bawah EthExplorer, tinggalkan pengaturan default untuk Deploy EthExplorer, yang benar.
- Di bawah Parameter lain, tinggalkan nilai default untuk Awalan URL Nested Template S3 dan

3. Biarkan semua pengaturan lain ke defaultnya, pilih kotak centang pengakuan, dan pilih Buat.

Halaman Stack Detail untuk tumpukan root yang CloudFormation diluncurkan muncul.

4. Untuk memantau kemajuan tumpukan root dan tumpukan bersarang, pilih Stacks.



## MyFirstEthereumStack

Stack name: MyFirstEthereumStack

5. Ketika semua tumpukan menampilkan CREATE\_COMPLETE untuk Status, Anda dapat terhubung ke antarmuka pengguna Ethereum untuk memverifikasi bahwa jaringan berjalan dan dapat diakses. Saat Anda menggunakan platform kontainer ECS, URLs untuk menghubungkan ke EthStats EthExplorer, dan EthJson RPC melalui Application Load Balancer tersedia di tab Output dari tumpukan root.

### Important

Anda tidak akan dapat terhubung langsung ke ini URLs atau SSH secara langsung sampai Anda mengatur koneksi proxy melalui host bastion di komputer klien Anda.

Untuk informasi selengkapnya, lihat [Connect ke EthStats dan EthExplorer Menggunakan Bastion Host](#).

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 'MyFirstEthereumStack' selected and highlighted in blue. Below the table, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. The 'Outputs' tab is active, showing a table with columns 'Key', 'Value', 'Description', and 'Export Name'. Three outputs are listed: 'EthStatsURL', 'EthExplorerURL', and 'EthJsonRPCURL'. The 'Value' for 'EthStatsURL' is highlighted with a red box.

Stack Name	Created Time	Status	Description
MyFirstEthereumStack-Ether... NESTED	2018-04-12 13:26:46 UTC-0700	CREATE_COMPLETE	This template creates an AutoScalingGroup of EC2 I...
MyFirstEthereumStack-Ether... NESTED	2018-04-12 13:26:38 UTC-0700	CREATE_COMPLETE	This template creates the ECS cluster and Ethereu...
MyFirstEthereumStack-Ether... NESTED	2018-04-12 13:25:59 UTC-0700	CREATE_COMPLETE	This template deploys an Ethereum cluster on an ex...
<input checked="" type="checkbox"/> 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 ke EthStats dan EthExplorer Menggunakan Bastion Host

Untuk terhubung ke sumber daya Ethereum dalam tutorial ini, Anda mengatur penerusan port SSH (SSH tunneling) melalui host bastion. Petunjuk berikut menunjukkan cara melakukan ini sehingga Anda dapat terhubung ke EthStats dan EthExplorer URLs menggunakan browser. Dalam petunjuk di bawah ini, Anda terlebih dahulu mengatur proxy SOCKS pada port lokal. Anda kemudian menggunakan ekstensi browser, [FoxyProxy](#), untuk menggunakan port yang diteruskan ini untuk jaringan Ethereum Anda. URLs

Jika Anda menggunakan Mac OS atau Linux, gunakan klien SSH untuk mengatur koneksi proxy SOCKS ke host bastion. Jika Anda adalah pengguna Windows, gunakan Putty. Sebelum Anda terhubung, konfirmasi bahwa komputer klien yang Anda gunakan ditentukan sebagai sumber yang diizinkan untuk lalu lintas SSH masuk dalam grup keamanan untuk Application Load Balancer yang Anda atur sebelumnya.

Untuk terhubung ke host bastion dengan penerusan port SSH menggunakan SSH

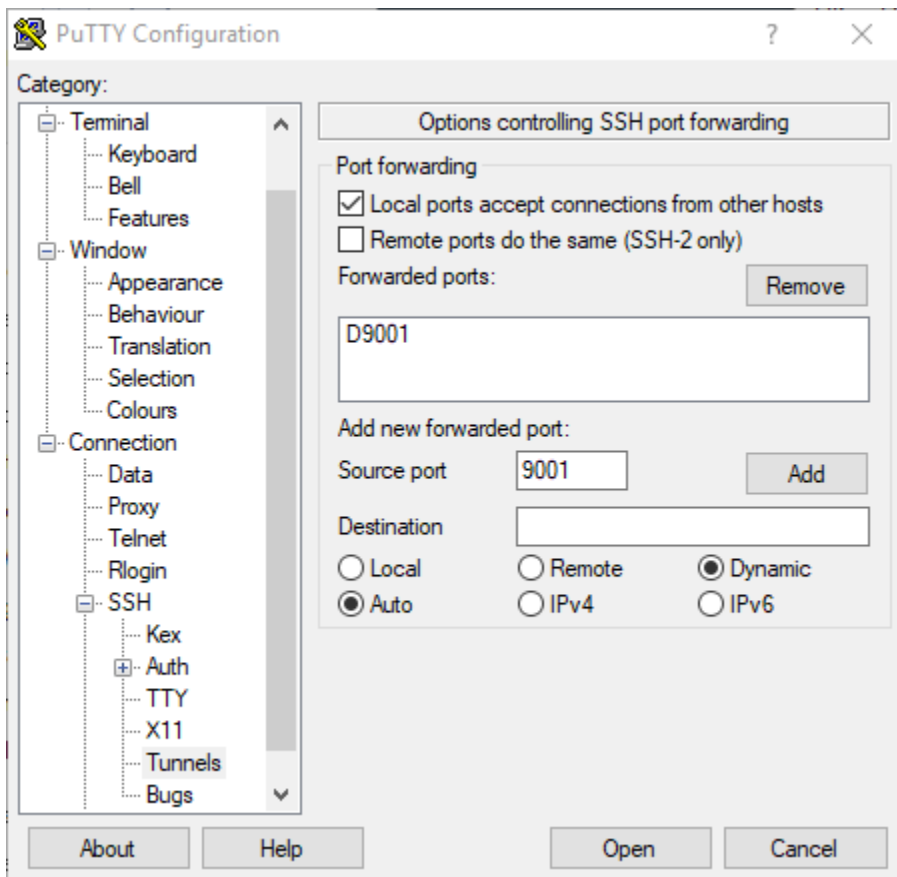
- Ikuti prosedur di [Menghubungkan ke Instans Linux Anda Menggunakan SSH](#) di Panduan EC2 Pengguna Amazon. Untuk langkah 4 dari prosedur [Connecting to Your Linux Instance](#), tambahkan `-D 9001` ke perintah SSH, tentukan key pair yang sama yang Anda tentukan di AWS Blockchain Template untuk konfigurasi Ethereum, dan tentukan nama DNS dari host bastion.

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

Untuk terhubung ke host bastion dengan penerusan port SSH menggunakan PuTTY (Windows)

- Ikuti prosedur dalam [Menghubungkan ke Instans Linux Anda dari Windows Menggunakan PuTTY](#) di Panduan EC2 Pengguna Amazon melalui langkah 7 dari prosedur [Memulai Sesi PuTTY](#), menggunakan key pair yang sama dengan yang Anda tentukan dalam konfigurasi AWS Blockchain Template for Ethereum.
- Di PuTTY, di bawah Kategori, pilih Koneksi, SSH, Terowongan.
- Untuk penerusan Port, pilih Port lokal menerima koneksi dari host lain.
- Di bawah Tambahkan port baru yang diteruskan:
  - Untuk port Sumber, masukkan 9001. Ini adalah port yang tidak terpakai sewenang-wenang yang kami pilih, dan Anda dapat memilih yang berbeda jika perlu.
  - Biarkan Tujuan kosong.
  - Pilih Dinamis.
  - Pilih Tambahkan.

Untuk port yang diteruskan, D9001 akan muncul seperti yang ditunjukkan di bawah ini.



5. Pilih Buka dan kemudian autentikasi ke host bastion seperti yang dipersyaratkan oleh konfigurasi kunci Anda. Biarkan koneksi terbuka.

Dengan koneksi PuTTY terbuka, Anda sekarang mengonfigurasi sistem atau ekstensi browser untuk menggunakan port yang diteruskan untuk jaringan Ethereum Anda. URLs Instruksi berikut didasarkan pada penggunaan FoxyProxy Standar untuk meneruskan koneksi berdasarkan pola URL EthStats dan EthExplorer dan port 9001, yang Anda tetapkan sebelumnya sebagai port yang diteruskan, tetapi Anda dapat menggunakan metode apa pun yang Anda inginkan.

Untuk mengkonfigurasi FoxyProxy untuk menggunakan terowongan SSH untuk jaringan Ethereum URLs

Prosedur ini ditulis berdasarkan Chrome. Jika Anda menggunakan browser lain, terjemahkan pengaturan dan urutan ke versi FoxyProxy untuk browser itu.

1. Unduh dan instal ekstensi browser FoxyProxy Standar, lalu buka Opsi sesuai dengan instruksi untuk browser Anda.
2. Pilih Tambahkan Proksi Baru.

3. Pada tab Umum, pastikan bahwa proxy Diaktifkan dan masukkan Nama Proxy dan Catatan Proxy yang membantu Anda mengidentifikasi konfigurasi proxy ini.
4. Pada tab Detail Proxy, pilih Konfigurasi Proxy Manual. Untuk Host atau Alamat IP (atau Server atau Alamat IP dalam beberapa versi), masukkan localhost. Untuk Port, masukkan 9001. Pilih SOCKS Proxy? .
5. Pada tab URL Pattern, pilih Add New Pattern.
6. Untuk nama Pola, masukkan nama yang mudah diidentifikasi, dan untuk Pola URL, masukkan pola yang cocok dengan semua sumber daya Ethereum yang URLs Anda buat dengan templat, misalnya `http://internal - MyUser -LoadB-*`. Untuk informasi tentang melihat URLs, lihat [Ethereum URLs](#).
7. Tinggalkan pilihan default untuk pengaturan lain dan pilih Simpan.

Anda sekarang dapat terhubung ke Ethereum URLs, yang tersedia di CloudFormation konsol menggunakan tab Output dari tumpukan root yang Anda buat dengan template.

## Pembersihan Sumber Daya

CloudFormation membuatnya mudah untuk membersihkan sumber daya yang dibuat tumpukan. Saat Anda menghapus tumpukan, semua sumber daya yang dibuat tumpukan akan dihapus.

Untuk menghapus sumber daya yang dibuat template

- Buka CloudFormation konsol, pilih tumpukan root yang Anda buat sebelumnya, pilih Tindakan, Hapus.

Status tumpukan root yang Anda buat sebelumnya dan tumpukan bersarang terkait diperbarui ke DELETE\_IN\_PROGRESS.

Anda dapat memilih untuk menghapus prasyarat yang Anda buat untuk jaringan Ethereum.

Hapus VPC

- Buka konsol Amazon VPC, pilih VPC yang Anda buat sebelumnya dan kemudian pilih Tindakan, Hapus VPC. Ini juga menghapus subnet, grup keamanan, dan gateway NAT yang terkait dengan VPC.

## Hapus peran IAM dan profil EC2 instans

- Buka konsol IAM dan pilih Peran. Pilih peran untuk ECS dan peran EC2 yang Anda buat sebelumnya dan pilih Hapus.

## Mengakhiri EC2 instance untuk host bastion

- Buka EC2 dasbor Amazon, pilih Menjalankan instance, pilih EC2 instance yang Anda buat untuk host bastion, pilih Actions, Instance State, Terminate.

# Template dan Fitur AWS Blockchain

Bagian ini menyediakan tautan bagi Anda untuk segera mulai membuat jaringan blockchain, serta informasi tentang opsi konfigurasi dan prasyarat untuk mengatur jaringan. AWS

Templat berikut ini tersedia:

- [Templat AWS Blockchain untuk Ethereum](#)
- [Template AWS Blockchain untuk Kain Hyperledger](#)

AWS Blockchain Templates tersedia di Wilayah berikut:

- Wilayah AS Barat (Oregon) (us-west-2)
- Wilayah AS Timur (Virginia N.) (us-east-1)
- Wilayah AS Timur (Ohio) (us-east-2)

## Note

Menjalankan template di Wilayah yang tidak tercantum di atas meluncurkan sumber daya di Wilayah AS Timur (Virginia N.) (us-east-1).

## Menggunakan Template AWS Blockchain untuk Ethereum

Ethereum adalah kerangka kerja blockchain yang menjalankan kontrak cerdas menggunakan Solidity, bahasa khusus Ethereum. Homestead adalah rilis Ethereum terbaru. Untuk informasi selengkapnya, lihat Dokumentasi [Ethereum Homestead dan dokumentasi Solidity](#).

## Tautan ke Peluncuran

Lihat [Memulai dengan AWS Blockchain Templates](#) untuk tautan yang akan diluncurkan CloudFormation di Wilayah tertentu menggunakan templat Ethereum.

## Opsi Ethereum

Saat Anda mengonfigurasi jaringan Ethereum menggunakan templat, Anda membuat pilihan yang menentukan persyaratan berikut:

- [Memilih Platform Kontainer](#)
- [Memilih Jaringan Ethereum Pribadi atau Publik](#)
- [Mengubah Akun Default dan Frasa Mnemonik](#)

## Memilih Platform Kontainer

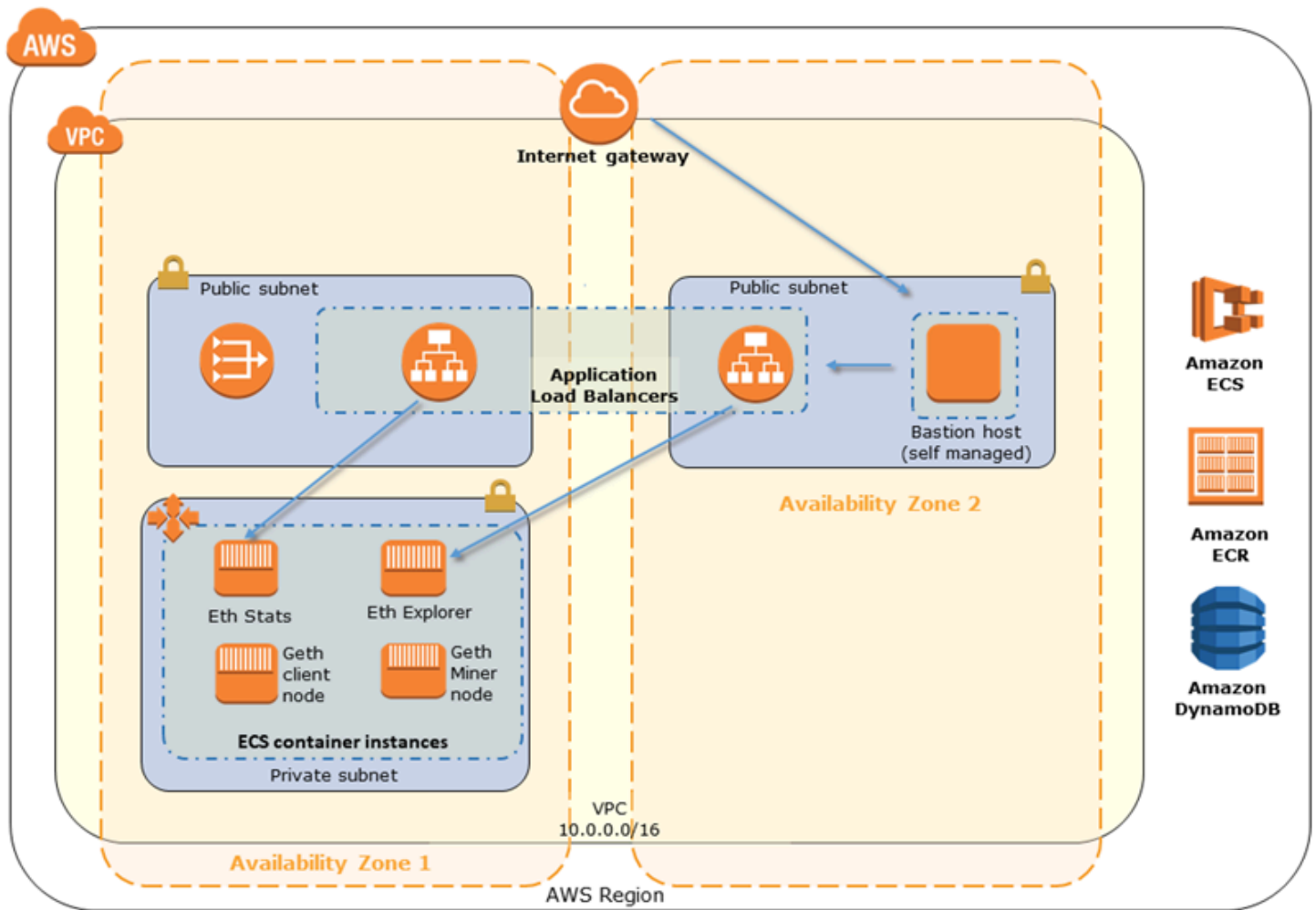
AWS Blockchain Templates menggunakan kontainer Docker yang disimpan di Amazon ECR untuk menyebarkan perangkat lunak blockchain. Template AWS Blockchain untuk Ethereum menawarkan dua pilihan untuk Platform Kontainer:

- `ecs` —Menentukan bahwa Ethereum berjalan pada cluster Amazon ECS dari instans Amazon. EC2
- `docker-local` —Menentukan bahwa Ethereum berjalan pada satu instance. EC2

## Menggunakan Platform Amazon ECS Container

Dengan Amazon ECS, Anda membuat jaringan Ethereum di kluster ECS yang terdiri dari beberapa EC2 instans, dengan Application Load Balancer dan sumber daya terkait. Untuk informasi selengkapnya tentang penggunaan konfigurasi Amazon ECS, lihat [Memulai dengan AWS Blockchain Templates](#) tutorialnya.

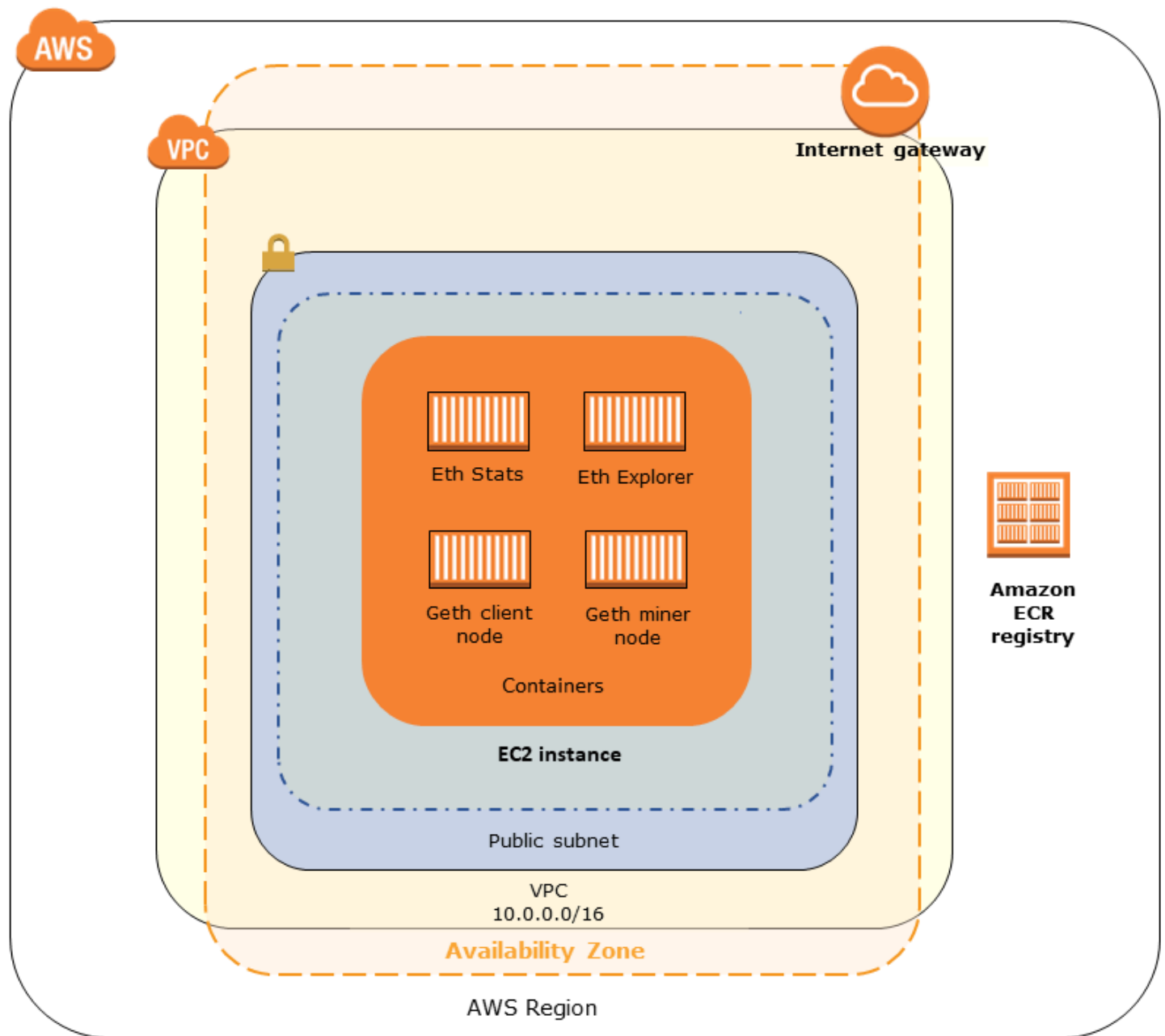
Diagram berikut menggambarkan jaringan Ethereum yang dibuat menggunakan template dengan opsi platform kontainer ECS:



### Menggunakan Platform Docker-Local

Atau, Anda dapat meluncurkan kontainer Ethereum dalam satu EC2 instans Amazon. Semua kontainer berjalan pada satu EC2 instance. Ini adalah pengaturan yang disederhanakan.

Diagram berikut menggambarkan jaringan Ethereum yang dibuat menggunakan template dengan opsi platform wadah docker-local:



## Memilih Jaringan Ethereum Pribadi atau Publik

Memilih nilai ID Jaringan Ethereum selain 1—4 menciptakan node Ethereum pribadi yang berjalan dalam jaringan yang Anda tentukan, menggunakan parameter jaringan pribadi yang Anda tentukan.

Ketika Anda memilih ID Jaringan Ethereum dari 1-4, node Ethereum yang Anda buat digabungkan ke jaringan Ethereum publik. Anda dapat mengabaikan pengaturan jaringan pribadi dan defaultnya. Jika Anda memilih untuk bergabung dengan node Ethereum ke jaringan Ethereum publik, pastikan bahwa layanan yang sesuai di jaringan Anda dapat diakses oleh internet.

## Mengubah Akun Default dan Frasa Mnemonik

Frasa mnemonik adalah kumpulan kata acak yang dapat Anda gunakan untuk menghasilkan dompet Ethereum (yaitu, pasangan private/public kunci) untuk akun terkait di jaringan apa pun. Frasa mnemonik dapat digunakan untuk mengakses Ether untuk akun terkait. Kami membuat mnemonic default yang terkait dengan akun default yang digunakan template Ethereum.

### Warning

Gunakan akun default dan frase mnemonik terkait hanya untuk tujuan pengujian. Jangan mengirim Eter asli menggunakan kumpulan akun default karena siapa pun yang memiliki akses ke frasa mnemonik dapat mengakses atau mencuri Eter dari akun. Sebagai gantinya, tentukan akun khusus untuk tujuan produksi. Frasa mnemonik yang terkait dengan akun default adalah `outdoor father modify clever trophy abandon vital feel portion grit evolve twist`

## Prasyarat

Saat Anda mengatur jaringan Ethereum menggunakan AWS Blockchain Template untuk Ethereum, persyaratan minimum yang tercantum di bawah ini harus dipenuhi. Template membutuhkan AWS komponen yang terdaftar untuk masing-masing kategori berikut:

### Topik

- [Prasyarat untuk Mengakses Sumber Daya Ethereum](#)
- [Prasyarat IAM](#)
- [Prasyarat Grup Keamanan](#)
- [Prasyarat VPC](#)
- [Contoh Izin IAM untuk Profil EC2 Instans dan Peran ECS](#)

## Prasyarat untuk Mengakses Sumber Daya Ethereum

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
Amazon EC2 key pair yang dapat Anda gunakan untuk	✓	✓

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
mengakses EC2 instans. Kuncinya harus ada di Wilayah yang sama dengan cluster ECS dan sumber daya lainnya.		
Komponen yang menghadap ke internet, seperti host benteng atau penyeimbang beban yang menghadap ke internet, dengan alamat internal tempat lalu lintas diizinkan masuk ke Application Load Balancer. Ini diperlukan dengan platform ECS karena template membuat penyeimbang beban internal untuk alasan keamanan. Ini diperlukan dengan platform docker-local saat EC2 instance berada di subnet pribadi, yang kami rekomendasikan. Untuk informasi tentang mengonfigurasi host bastion, lihat. <a href="#">Buat Host Bastion</a>	✓	✓ (dengan subnet pribadi)

## Prasyarat IAM

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
Prinsipal IAM (pengguna atau grup) yang memiliki izin untuk bekerja dengan semua layanan terkait.	✓	✓

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
Profil EC2 instans Amazon dengan izin yang sesuai untuk EC2 instans untuk berinteraksi dengan layanan lain. Untuk informasi selengkapnya, lihat <a href="#">To create an EC2 instance profile</a> .	✓	✓
Peran IAM dengan izin untuk Amazon ECS untuk berinteraksi dengan layanan lain. Untuk informasi selengkapnya, lihat <a href="#">Membuat Peran dan Izin ECS</a> .	✓	

## Prasyarat Grup Keamanan

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
Grup keamanan untuk EC2 instans, dengan persyaratan berikut:	✓	✓
<ul style="list-style-type: none"> <li>Aturan keluar yang memungkinkan lalu lintas ke 0.0.0.0/0 (default).</li> </ul>	✓	✓
<ul style="list-style-type: none"> <li>Aturan masuk yang memungkinkan semua lalu lintas dari dirinya sendiri (grup keamanan yang sama).</li> </ul>	✓	✓
<ul style="list-style-type: none"> <li>Aturan masuk yang memungkinkan semua lalu lintas dari grup keamanan</li> </ul>	✓	

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
untuk Application Load Balancer.		
<ul style="list-style-type: none"><li>• Aturan masuk yang memungkinkan HTTP (port 80), EthStats (disajikan pada port 8080), JSON RPC melalui HTTP (port 8545), dan SSH (port 22) dari sumber eksternal terpercaya , seperti IP CIDR komputer klien Anda.</li></ul>		✓

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
<p>Grup keamanan untuk Application Load Balancer, dengan persyaratan sebagai berikut:</p> <ul style="list-style-type: none"><li>• Aturan masuk yang memungkinkan semua lalu lintas dari dirinya sendiri (grup keamanan yang sama).</li><li>• Aturan masuk yang memungkinkan semua lalu lintas dari grup keamanan untuk EC2 instance.</li><li>• Aturan keluar yang memungkinkan semua lalu lintas hanya ke grup keamanan untuk EC2 instance. Untuk informasi selengkapnya, lihat <a href="#">Buat Grup Keamanan</a>.</li><li>• Jika mengaitkan grup keamanan yang sama ini dengan host bastion, aturan masuk yang memungkinkan lalu lintas SSH (port 22) dari sumber tepercaya.</li><li>• Jika host bastion atau komponen lain yang menghadap ke internet berada dalam grup keamanan yang berbeda, aturan masuk yang</li></ul>	✓	

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
memungkinkan lalu lintas dari komponen tersebut.		

## Prasyarat VPC

Prasyarat	Untuk Platform ECS	Untuk Docker-Local
Alamat IP Elastis, yang digunakan untuk mengakses layanan Ethereum.	✓	✓
Sebuah subnet untuk menjalankan EC2 instance. Kami sangat merekomendasikan subnet pribadi.	✓	✓
Dua subnet yang dapat diakses publik. Setiap subnet harus berada di Availability Zone yang berbeda satu sama lain, dengan satu di Availability Zone yang sama dengan subnet untuk EC2 instance.	✓	

## Contoh Izin IAM untuk Profil EC2 Instans dan Peran ECS

Anda menentukan EC2 contoh profil ARN sebagai salah satu parameter saat Anda menggunakan template. Jika Anda menggunakan platform kontainer ECS, Anda juga menentukan peran ECS ARN. Kebijakan izin yang dilampirkan pada peran ini memungkinkan sumber daya dan instans AWS di kluster Anda berinteraksi dengan sumber daya AWS lainnya. Untuk informasi lebih lanjut, lihat [Peran IAM](#) dalam Panduan Pengguna IAM. Gunakan pernyataan kebijakan dan prosedur di bawah ini sebagai titik awal untuk membuat izin.

## Contoh Kebijakan Izin untuk Profil EC2 Instance

Kebijakan izin berikut menunjukkan tindakan yang diizinkan untuk profil EC2 instans saat Anda memilih platform kontainer ECS. Pernyataan kebijakan yang sama dapat digunakan dalam platform container docker-local, dengan kunci ecs konteks dihapus untuk membatasi akses.

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

## Membuat Peran dan Izin ECS

Untuk izin yang dilampirkan pada peran ECS, kami sarankan Anda memulai dengan kebijakan EC2 ContainerServiceRole izin Amazon. Gunakan prosedur berikut untuk membuat peran dan melampirkan kebijakan izin ini. Gunakan konsol IAM untuk melihat sebagian besar up-to-date izin dalam kebijakan ini.

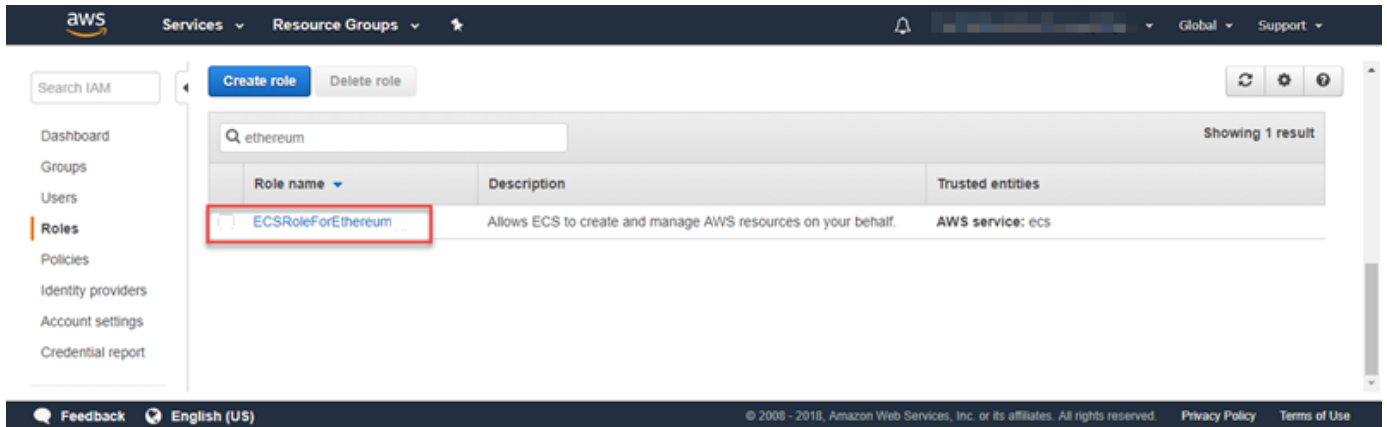
Untuk membuat peran IAM untuk Amazon ECS

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Dalam panel navigasi, pilih Roles (Peran), Create role (Buat Peran).
3. Di bawah Pilih jenis entitas tepercaya, pilih Layanan AWS.
4. Untuk Choose the service that will use this role (Pilih layanan yang akan menggunakan peran ini), pilih Elastic Container Service.
5. Di bawah Pilih kasus penggunaan Anda, pilih Layanan Kontainer Elastis, Berikutnya:Izin.

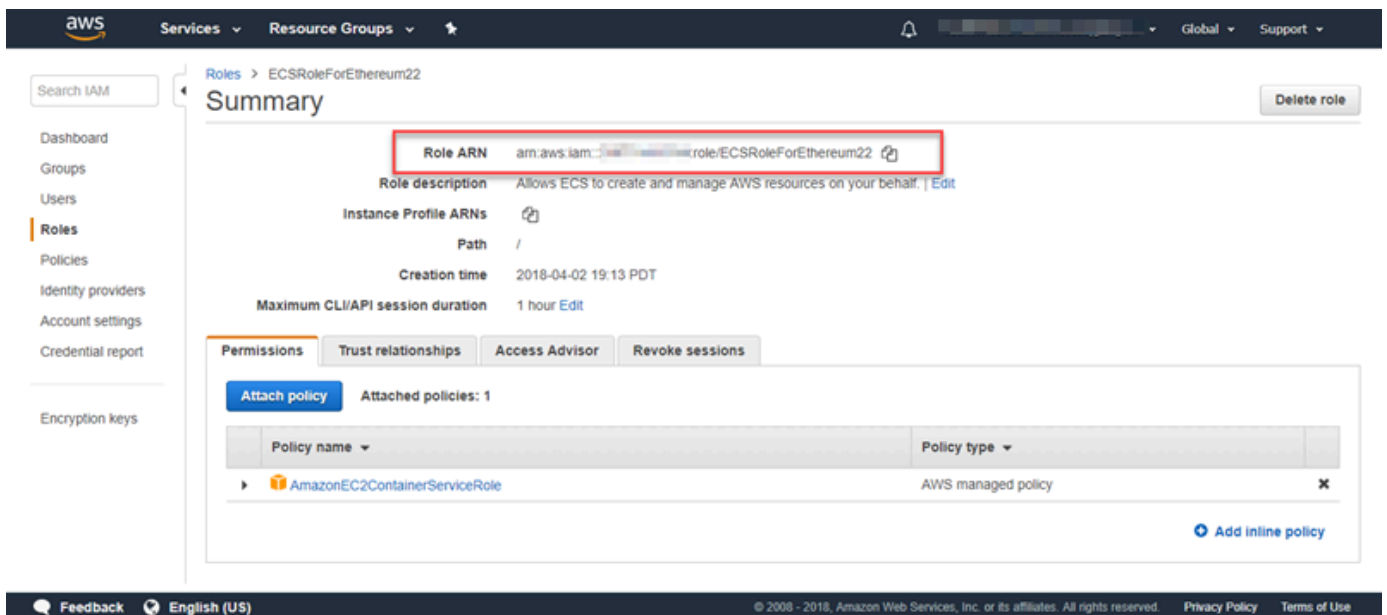
The screenshot shows the AWS IAM console 'Create role' wizard. The 'Select type of trusted entity' step has 'AWS service' selected. The 'Choose the service that will use this role' step shows a grid of services, with 'Elastic Container Service' highlighted. The 'Select your use case' step shows 'Elastic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

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

6. Untuk kebijakan Izin, biarkan kebijakan default (Amazon EC2 ContainerServiceRole) dipilih, dan pilih Next:Review.
7. Untuk nama Peran, masukkan nilai yang membantu Anda mengidentifikasi peran, seperti ECSRoleForEthereum. Untuk Deskripsi Peran, masukkan ringkasan singkat. Perhatikan nama peran untuk nanti.
8. Pilih Buat peran.
9. Pilih peran yang baru saja Anda buat dari daftar. Jika akun Anda memiliki banyak peran, Anda dapat mencari nama peran.



10. Salin nilai ARN Peran dan simpan sehingga Anda dapat menyalinnya lagi. Anda memerlukan ARN ini saat membuat jaringan Ethereum.



## Menghubungkan ke Sumber Daya Ethereum

Setelah tumpukan root yang Anda buat dengan template menunjukkan CREATE\_COMPLETE, Anda dapat terhubung ke sumber daya Ethereum menggunakan konsol. CloudFormation Cara Anda terhubung tergantung pada platform kontainer yang Anda pilih, ECS atau docker-local:

- ECS —Tab Output dari tumpukan root menyediakan tautan ke layanan yang berjalan pada Application Load Balancer. Ini URLs tidak dapat diakses secara langsung karena alasan keamanan. Untuk terhubung, Anda dapat mengatur dan menggunakan host bastion untuk koneksi proxy ke mereka. Untuk informasi selengkapnya, lihat [Koneksi Proxy Menggunakan Host Bastion](#) di bawah ini.
- docker-local —Anda terhubung menggunakan alamat IP EC2 instans hosting layanan Ethereum seperti yang tercantum di bawah ini. Gunakan EC2 konsol untuk menemukan contoh yang dibuat template. *ec2-IP-address*
  - EthStats—Gunakan `http://ec2-IP-address`
  - EthExplorer—Gunakan `http://:8080 ec2-IP-address`
  - EthJsonRpc—Gunakan `http://:8545 ec2-IP-address`

Jika Anda menentukan subnet publik untuk Ethereum Network Subnet ID (Daftar Subnet VPC untuk digunakan dalam template), Anda dapat terhubung secara langsung. Klien Anda harus menjadi sumber terpercaya lalu lintas masuk untuk SSH (port 22), serta port yang terdaftar. Ini ditentukan oleh Grup EC2 Keamanan yang Anda tentukan menggunakan Template AWS Blockchain untuk Ethereum.

Jika Anda menentukan subnet pribadi, Anda dapat mengatur dan menggunakan host bastion untuk koneksi proxy ke alamat ini. Untuk informasi selengkapnya, lihat [Koneksi Proxy Menggunakan Host Bastion](#) di bawah ini.

### Koneksi Proxy Menggunakan Host Bastion

Dengan beberapa konfigurasi, layanan Ethereum mungkin tidak tersedia untuk umum. Dalam kasus tersebut, Anda dapat terhubung ke sumber daya Ethereum melalui host benteng. Untuk informasi selengkapnya tentang host bastion, lihat [Linux Bastion Host Architecture](#) di Panduan Mulai Cepat Linux Bastion Host.

Host benteng adalah sebuah EC2 instance. Pastikan persyaratan berikut terpenuhi:

- EC2 Instance untuk host bastion berada dalam subnet publik dengan Auto-assign Public IP diaktifkan dan yang memiliki gateway internet.
- Host bastion memiliki key pair yang memungkinkan koneksi ssh.
- Host bastion dikaitkan dengan grup keamanan yang memungkinkan lalu lintas SSH masuk dari klien yang terhubung.
- Grup keamanan yang ditugaskan ke host Ethereum (misalnya, Application Load Balancer jika ECS adalah platform container, atau EC2 instance host jika docker-local adalah platform container) memungkinkan lalu lintas masuk pada semua port dari sumber dalam VPC.

Dengan pengaturan host bastion, pastikan bahwa klien yang terhubung menggunakan host bastion sebagai proxy. Contoh berikut menunjukkan pengaturan koneksi proxy menggunakan Mac OS. Ganti *BastionIP* dengan alamat IP EC2 instance host bastion dan *MySshKey.pem* dengan file key pair yang Anda salin ke host bastion.

Pada baris perintah, ketik yang berikut ini:

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

Ini mengatur penerusan port untuk port 9001 pada mesin lokal ke host benteng.

Selanjutnya, konfigurasi browser atau sistem Anda untuk menggunakan proxy SOCKS untuk `localhost:9001`. Misalnya, menggunakan Mac OS, pilih System Preferences, Network, Advanced, pilih SOCKS proxy, dan ketik localhost: 9001.

Menggunakan FoxyProxy Standar dengan Chrome, pilih Alat Lainnya, Ekstensi. Di bawah FoxyProxy Standar, pilih Detail, Opsi ekstensi, Tambahkan Proksi Baru. Pilih Konfigurasi Proxy Manual. Untuk Host atau IP Address ketik localhost dan untuk tipe Port 9001. Pilih proxy SOCKS? , Simpan.

Anda sekarang harus dapat terhubung ke alamat host Ethereum yang tercantum dalam output template.

## Menggunakan Template AWS Blockchain untuk Hyperledger Fabric

Hyperledger Fabric adalah kerangka kerja blockchain yang menjalankan kontrak pintar yang disebut chaincode, yang ditulis dalam Go. Anda dapat membuat jaringan pribadi dengan Hyperledger Fabric, membatasi rekan-rekan yang dapat terhubung dan berpartisipasi dalam jaringan. Untuk informasi lebih lanjut tentang Hyperledger Fabric, lihat dokumentasi [Hyperledger Fabric](#). [Untuk informasi](#)

[selengkapnya tentang chaincode, lihat topik Chaincode for Developers di dokumentasi Hyperledger Fabric.](#)

Template AWS Blockchain untuk Hyperledger Fabric hanya mendukung platform kontainer docker-lokal, yang berarti kontainer Hyperledger Fabric digunakan pada satu instans EC2.

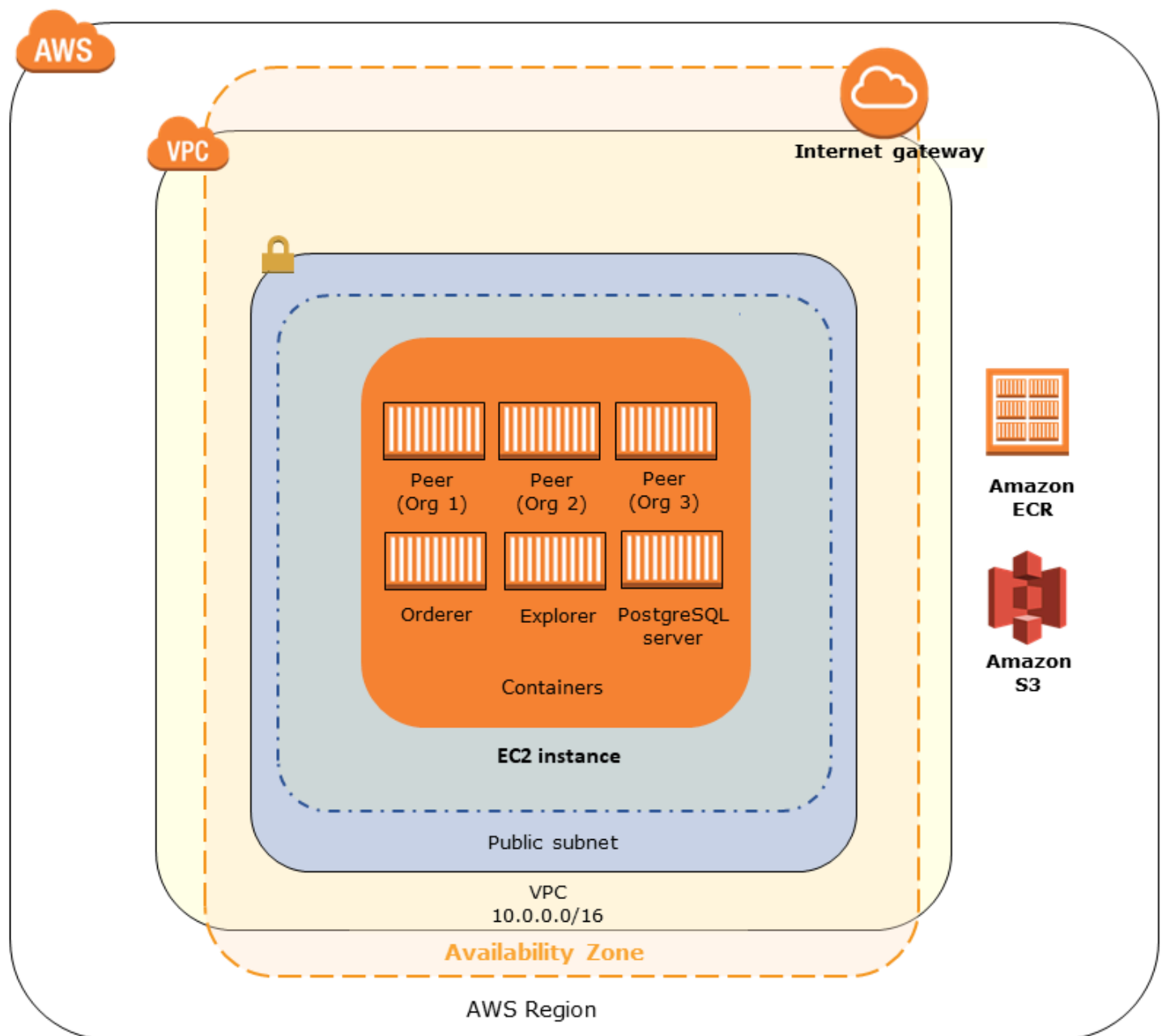
## Tautan ke Peluncuran

Lihat [Memulai dengan AWS Blockchain Templates](#) untuk tautan yang akan diluncurkan CloudFormation di Wilayah tertentu menggunakan templat Hyperledger Fabric.

## Template AWS Blockchain untuk Komponen Kain Hyperledger

Template AWS Blockchain untuk Hyperledger Fabric membuat instans EC2 dengan Docker, dan meluncurkan jaringan Hyperledger Fabric menggunakan container pada instance itu. Jaringan ini mencakup satu layanan pesanan dan tiga organisasi, masing-masing dengan satu layanan sejawat. Template ini juga meluncurkan wadah Hyperledger Explorer, yang memungkinkan Anda menelusuri data blockchain. Kontainer server PostgreSQL diluncurkan untuk mendukung Hyperledger Explorer.

Diagram berikut menggambarkan jaringan Hyperledger Fabric yang dibuat menggunakan template:



## Prasyarat

Sebelum Anda meluncurkan jaringan Hyperledger Fabric menggunakan template, pastikan bahwa persyaratan berikut terpenuhi:

- Prinsip IAM (pengguna atau grup) yang Anda gunakan harus memiliki izin untuk bekerja dengan semua layanan terkait.
- Anda harus memiliki akses ke key pair yang dapat Anda gunakan untuk mengakses instans EC2 (misalnya, menggunakan SSH). Kuncinya harus ada di wilayah yang sama dengan instance.

- Anda harus memiliki profil instans EC2 dengan kebijakan izin terlampir yang memungkinkan akses ke Amazon S3 dan ke Amazon Elastic Container Registry (Amazon ECR) Registry ECR) untuk menarik kontainer. Untuk contoh kebijakan izin, lihat [Contoh Izin IAM untuk Profil Instans EC2](#).
- Anda harus memiliki jaringan VPC Amazon dengan subnet publik, atau subnet pribadi dengan NAT Gateway dan alamat IP Elastis sehingga Amazon S3,, CloudFormation dan Amazon ECR dapat diakses.
- Anda harus memiliki grup keamanan EC2 dengan aturan masuk yang memungkinkan lalu lintas SSH (port 22) dari alamat IP yang perlu terhubung ke instance menggunakan SSH, dan hal yang sama untuk klien yang perlu terhubung ke Hyperledger Explorer (port 8080).

## Contoh Izin IAM untuk Profil Instans EC2

Anda menentukan profil instans EC2 ARN sebagai salah satu parameter saat Anda menggunakan AWS Blockchain Template for Hyperledger Fabric. Gunakan pernyataan kebijakan berikut sebagai titik awal untuk kebijakan izin yang dilampirkan pada peran EC2 dan profil instans tersebut.

### 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": "*"
    }
  ]
}
```

```
}
```

## Menghubungkan ke Sumber Daya Kain Hyperledger

Setelah tumpukan root yang Anda buat dengan template menunjukkan CREATE\_COMPLETE, Anda dapat terhubung ke sumber daya Hyperledger Fabric pada instance EC2. Jika Anda menentukan subnet publik, Anda dapat terhubung ke instans EC2 seperti halnya instans EC2 lainnya. Untuk informasi selengkapnya, lihat [Menghubungkan ke Instans Linux Anda Menggunakan SSH](#) di Panduan Pengguna Amazon EC2.

Jika Anda menentukan subnet pribadi, Anda dapat mengatur dan menggunakan host bastion untuk koneksi proxy ke sumber daya Hyperledger Fabric. Untuk informasi selengkapnya, lihat [Koneksi Proxy Menggunakan Host Bastion](#) di bawah ini.

### Note

Anda mungkin memperhatikan bahwa template mengalokasikan alamat IP publik ke instans EC2 yang menghosting layanan Hyperledger Fabric; namun, alamat IP ini tidak dapat diakses publik karena kebijakan perutean di subnet pribadi yang Anda tentukan tidak mengizinkan lalu lintas antara alamat IP ini dan sumber publik.

## Koneksi Proxy Menggunakan Host Bastion

Dengan beberapa konfigurasi, layanan Hyperledger Fabric mungkin tidak tersedia untuk umum. Dalam kasus tersebut, Anda dapat terhubung ke sumber daya Hyperledger Fabric melalui host benteng. Untuk informasi selengkapnya tentang host bastion, lihat [Linux Bastion Host Architecture](#) di Panduan Mulai Cepat Linux Bastion Host.

Host bastion adalah instance EC2. Pastikan persyaratan berikut terpenuhi:

- Instans EC2 untuk host bastion berada dalam subnet publik dengan Auto-assign Public IP diaktifkan dan yang memiliki gateway internet.
- Host bastion memiliki key pair yang memungkinkan koneksi ssh.
- Host bastion dikaitkan dengan grup keamanan yang memungkinkan lalu lintas SSH masuk dari klien yang terhubung.

- Grup keamanan yang ditugaskan ke host Hyperledger Fabric (misalnya, Application Load Balancer jika ECS adalah platform kontainer, atau instance host EC2 jika docker-local adalah platform kontainer) memungkinkan lalu lintas masuk pada semua port dari sumber dalam VPC.

Dengan pengaturan host bastion, pastikan bahwa klien yang terhubung menggunakan host bastion sebagai proxy. Contoh berikut menunjukkan pengaturan koneksi proxy menggunakan Mac OS. Ganti *BastionIP* dengan alamat IP instans bastion host EC2 dan *MySshKey.pem* dengan file key pair yang Anda salin ke host bastion.

Pada baris perintah, ketik yang berikut ini:

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

Ini mengatur penerusan port untuk port 9001 pada mesin lokal ke host benteng.

Selanjutnya, konfigurasi browser atau sistem Anda untuk menggunakan proxy SOCKS untuk `localhost:9001`. Misalnya, menggunakan Mac OS, pilih System Preferences, Network, Advanced, pilih SOCKS proxy, dan ketik `localhost:9001`.

Menggunakan FoxyProxy Standar dengan Chrome, pilih Alat Lainnya, Ekstensi. Di bawah FoxyProxy Standar, pilih Detail, Opsi ekstensi, Tambahkan Proksi Baru. Pilih Konfigurasi Proxy Manual. Untuk Host atau IP Address ketik `localhost` dan untuk tipe Port `9001`. Pilih proxy SOCKS? , Simpan.

Anda sekarang harus dapat terhubung ke alamat host Hyperledger Fabric yang tercantum dalam output template.

# Riwayat Dokumen

Tabel berikut menjelaskan perubahan dokumentasi untuk panduan ini.

Pembaruan dokumentasi terbaru: Mei 1, 2019

Ubah	Deskripsi	Date
Penghentian Template Blockchain AWS.	AWS Blockchain Templates dihentikan pada 30 April 2019. Tidak ada pembaruan lebih lanjut untuk layanan ini atau dokumentasi pendukung ini akan dilakukan. Untuk pengalaman Blockchain Terkelola terbaik AWS, kami sarankan Anda menggunakan <a href="#">Amazon Managed Blockchain (AMB)</a> .	1 Mei 2019
Pembaruan host Bastion.	Tutorial memulai yang dimodifikasi dan persyaratan prasyarat Ethereum untuk penambahan host bastion, yang memungkinkan akses ke sumber daya web yang disajikan melalui penyeimbangan beban internal saat menggunakan platform ECS dan instance saat menggunakan docker-local. EC2	3 Mei 2018
Panduan yang dibuat.	Panduan pengembang baru untuk mendukung rilis awal AWS Blockchain Templates.	19 April 2018

# AWS Glosarium

Untuk AWS terminologi terbaru, lihat [AWS glosarium di Referensi](#).Glosarium AWS