



アーキテクチャ図

AWS App Runner でのサーバーレスウェブホスティング



AWS App Runner でのサーバーレスウェブホスティング: アーキテクチャ図

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

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

でのサーバーレスウェブホスティング AWS App Runner i

でのサーバーレスウェブホスティング AWS App Runner 1

でのサーバーレスウェブホスティング AWS App Runner 3

詳細情報 4

図の履歴 4

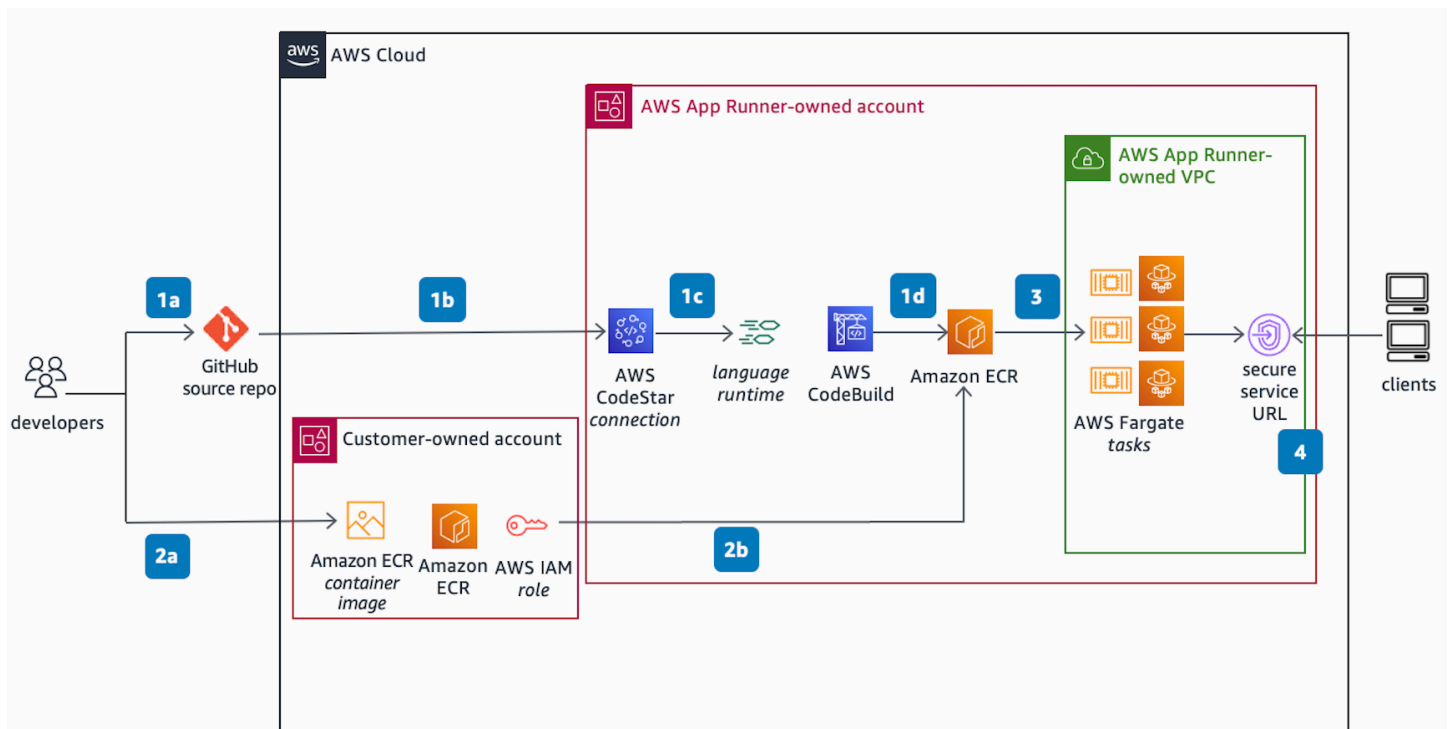
..... vi

でのサーバーレスウェブホスティング AWS App Runner

公開日: 2023 年 5 月 3 日 ([図の履歴](#))

このリファレンスアーキテクチャでは、インフラストラクチャをプロビジョニングまたは管理することなく、サーバーレスでコンテナ化されたウェブアプリケーションを実行する方法に関する 2 つのシナリオについて詳しく説明します。

でのサーバーレスウェブホスティング AWS App Runner



デベロッパーエクスペリエンス: AWS App Runnerを使用して、ソースコードとソースイメージの 2 種類のサービスソースに基づいてウェブサービスを作成および管理します。

1. ソースコードを使用したシナリオ 1:

- GitHub ソースリポジトリにコードをコミットします。
- を使用して GitHub のソースコードへのアクセスを許可しますAWS CodeStar。

c. ウェブサービスを構築および実行するための手順を提供し、でサポートされているマネージドランタイム環境を指定しますAWS App Runner。AWS CodeBuildその後、はコードとその依存関係を自動パッケージ化し、新しいコンテナイメージを構築します。

d. その後、コンテナイメージはプッシュされ、AWS App Runner所有アカウント内でホストされている Amazon Elastic Container Registry (Amazon ECR) リポジトリに保存されます。

2. ソースイメージを使用するシナリオ 2:

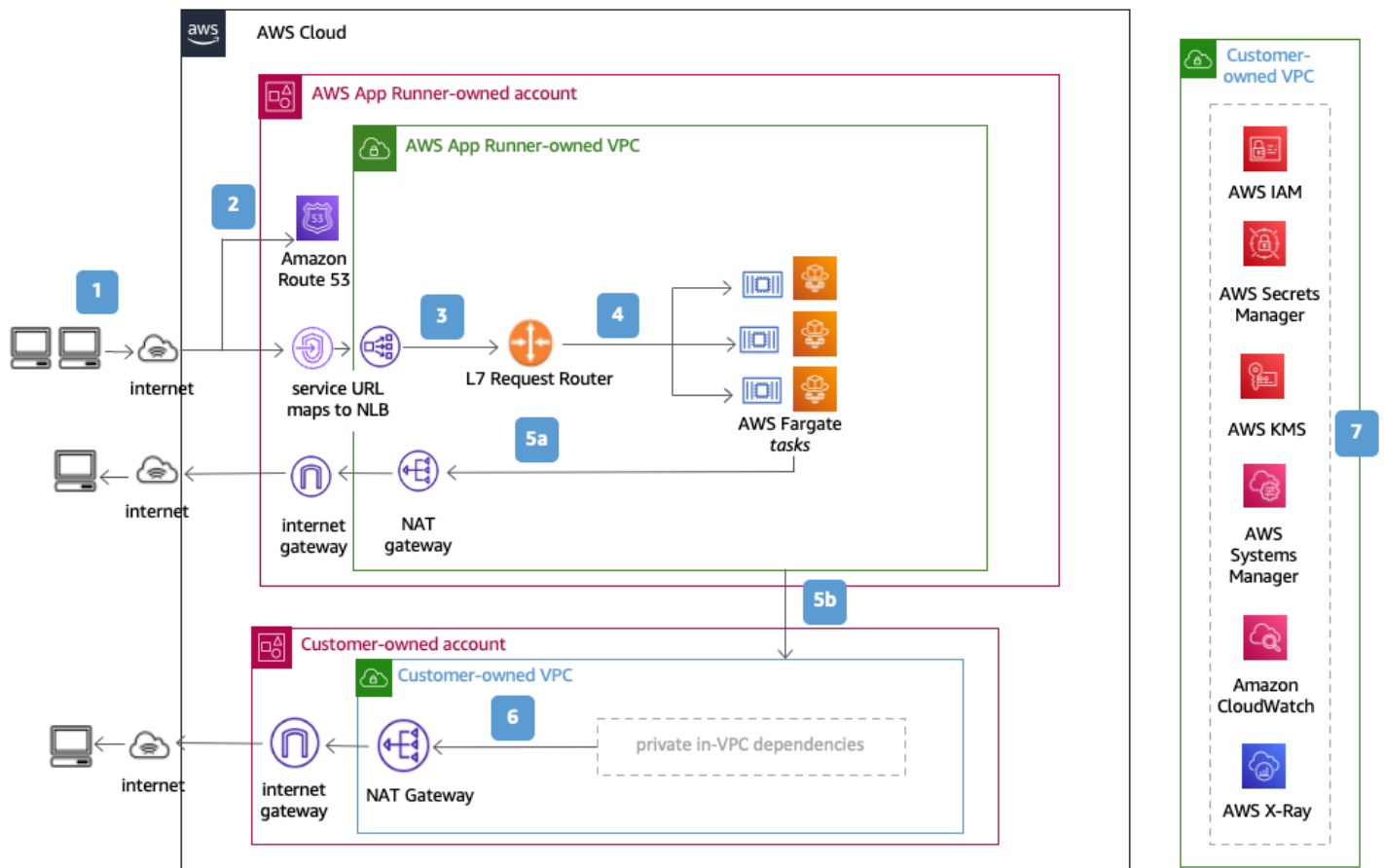
a. または、カスタマー所有アカウント内でホストされているカスタマー管理の Amazon ECR リポジトリに保存されている構築済みイメージから始めます。

b. AWS IAM アイデンティティセンター ロールを使用してコンテナイメージへのアクセスを許可することで、イメージは AWS App Runnerマネージド Amazon ECR リポジトリにプッシュされて保存されます。

App Runner サービスの作成:

- その後、Amazon ECR に保存されているコンテナイメージがプルされ、AWS App Runnerサービス VPC と呼ばれる AWS App Runner所有の Amazon Virtual Private Cloud 内のサーバーレスAWS Fargateタスクとしてデプロイされます。AWS Fargate タスクは、プライベートサブネットで高可用性を実現するために複数のアベイラビリティゾーンで実行されている AWS Fargateクラスターにデプロイされ、AWS Application Auto Scaling で設定されます。
- 安全なサービス URL は、デプロイが成功した後に作成/公開されます。Secure Service URL は、公開されている AWS Network Load Balancer にマッピングされ、によってデフォルトのドメイン名が割り当てられますAWS App Runner。お客様は、所有するカスタムドメイン名を関連付け/マッピングすることもできますが、パブリックに解決可能で、Amazon Route 53 Public Hosted Zone または任意の DNS プロバイダーに登録できる必要があります。

でのサーバーレスウェブホスティング AWS App Runner



AWS App Runner は、パブリックエンドポイントまたはサービス URL を使用してウェブアプリケーションを公開するための安全で一貫性のあるソリューションです。

インバウンドトラフィックパス:

1. クライアントはサービス URL へのリクエストを開始します。
2. サービス URL は、Amazon Route 53 を使用して公開されている Network Load Balancer に解決されます。 AWS NLB は自動的にプロビジョニングされ、AWS App Runnerサービスによって所有されます。
3. リクエストは AWS Network Load Balancer (AWS NLB) を介して AWS App Runner所有の VPC に入り、L7 リクエストルーターにリダイレクトされます。
4. L7 リクエストルーターは、受信リクエストを特定のウェブサービスインスタンスまたはAWS Fargateタスクにルーティングします。

アウトバウンドトラフィックパス、シナリオ 1 (ウェブサービスでは、顧客/プライベート VPC 内のダウンストリーム依存関係にアクセスする必要はありません)。

5. a. アウトバウンド (またはリターン) トラフィックは、AWS マネージド NAT ゲートウェイ、および AWS App Runner 所有 VPC 内にプロビジョニングされたインターネットゲートウェイ、またはデフォルトのネットワークモードを介してインターネットにルーティングされます。

アウトバウンドトラフィックパス、シナリオ 2 (ウェブサービスでは、顧客/プライベート VPC 内のダウンストリーム依存関係にアクセスする必要があります)。

- b. アウトバウンドトラフィックは顧客/プライベート所有の VPC に転送され、プライベートリソース/依存関係はエンドユーザー/開発者によって手動でプロビジョニングされます。トラフィックは、VPC ルーティングテーブルごとに送信先エンドポイントに転送されます。
6. インターネットへのトラフィックは、カスタマー管理の NAT ゲートウェイと、カスタマー/プライベート所有の VPC 内にプロビジョニングされたインターネットゲートウェイ、または VPC Egress ネットワークモードを介してルーティングされます。
7. AWS Identity and Access Management (AWS IAM)、AWS Key Management Service (AWS KMS)、AWS Secrets Manager、および AWS Systems Manager Parameter Store は、ロールベースのアクセスを確保し、機密データを安全に保存します。Amazon CloudWatch とはオブザーバビリティ AWS X-Ray を維持します。

詳細情報

詳細については、「」を参照してください。

- [AWS アーキテクチャアイコン](#)
- [AWS アーキテクチャセンター](#)
- [AWS Well-Architected](#)

図の履歴

このリファレンスアーキテクチャ図の更新について通知を受け取るには、RSS フィードをサブスクライブします。

変更

説明

日付

[初版発行](#)

リファレンスアーキテクチャ 2021 年 5 月 3 日
図が最初に公開されました。

 Note

RSS 更新をサブスクライブするには、使用しているブラウザで RSS プラグインが有効になっている必要があります。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。