



AWS IPv6 ハンズオン Advance

ハンズオンテキスト

Document Version	1.2	Effective Date	2022.03.31
Creation Date	2022.03.31	Document Author	Nobuaki Kikuchi
Last Revision Date	2022.11.18	Revised By	Nobuaki Kikuchi

目次

1. イントロダクション	4
1.1. 目的	4
1.2. 対象者	4
1.3. 準備物	4
2. はじめに	5
2.1. 操作環境の確認	5
3. ハイブリット VPC の作成	7
3.1. Client VPC 環境	7
3.2. 踏み台用 VPC 環境の作成	7
3.3. Linux Client へのログイン	14
3.4. DNS64/NAT64 の動作確認	17
4. 公開ウェブサーバー環境として接続先環境を作成	22
4.1. 接続先 VPC 環境	22
4.2. VPC 内のネットワーク環境作成	22
4.3. Web サーバの配置	28
4.4. Elastic Load Balancer (ALB) の設定	36
4.5. ALB への接続確認	49
5. DNS で IPv6 を FQDN に登録する	52
5.1. Route 53 でプライベートホストゾーンの登録	53
5.2. Linux OS から名前解決の確認	59
6. ALB の前に NLB を配置	61
6.1. NLB を作成	62
6.2. NLB への接続確認	71
7. ハンズオン環境のクリーニング	73
7.1. クリーニング対象リソース	73

8. ドキュメント情報.....	74
8.1. 更新履歴	74

1. イントロダクション

1.1. 目的

本セッションはIPv6 対応環境を AWS で構築し、実際にアクセスしていただくハンズオン形式のセミナーです。参加者の PC から AWS 上に起動した Linux クライアントにリモート接続し、そこを起点として、AWS サービスで構成する Web サーバ、DNS、インターネット上の他のサイトへアクセスいただきます。このセミナーを通し、AWS 上で構築した IPv6 環境をご理解いただける内容となっています。

1.2. 対象者

このテキストは技術者向けのハンズオンとして利用されることを想定しています。すでに初級レベルのハンズオンを体験済みで、ご自身のスキルで AWS サービスのマネジメントコンソールを利用し、EC2 インスタンスなどの起動、VPC におけるルーティングの設定作業ができる方が対象です。また、講師の説明を参考にお手元のパソコンからセッションマネージャーを利用し Linux ホストにログイン、VIM でテキストファイルを編集できる方を前提としています。

1.3. 準備物

パソコン：

Windows OS、Mac OS にて、デュアルディスプレイ環境を強く推奨いたします

ブラウザ環境：

Google Chrome (最新バージョン)

Mozilla Firefox (最新バージョン)

インターネット接続環境：

1 Mbps 以上の回線を利用できる事

2. はじめに

2.1. 操作環境の確認

受講者はお手元のパソコンから、AWS が Web ページ上で提供するマネジメントコンソールへログインし、必要なリソースを作成します。


Web ブラウザから以下の URL へアクセスし、右上の「コンソールにサインイン」からサインイン画面へ進み、あらかじめ用意した AWS アカウントを利用してマネジメントコンソールにログインできることを確認してください。

<https://aws.amazon.com/jp/console/>

主催者から、別途、AWS アカウントのログイン方法が指定されている場合、そちらの指示に従ってください。

The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with the AWS logo, links for 'お問い合わせ', 'サポート', '日本語', and 'アカウント'. A red box highlights the 'コンソールにサインイン' button. Below the navigation bar, there's a section titled 'AWS マネジメントコンソール' with a sub-header 'AWS クラウドにアクセスして管理するために必要なものすべてを 1 つのウェブインターフェイスに集結'. A large orange button says 'もう一度ログインする'. Below this is a section titled 'AWS の詳細を見る' with four cards:

- モバイルアプリとウェブアプリを迅速に構築**
わずか数行のコードで、認証と AWS Amplify とのデータ同期を追加
[詳細 >](#)
- Spot Blueprints の概要**
Kubernetes および Apache Spark などのフレームワーク向けテンプレートジェネレータ
[詳細 >](#)
- GraphQL を使用して API を最新化する**
AWS AppSync は、アプリケーションのパフォーマンスと開発者の生産性を向上させるフルマネージドの GraphQL サービスです
[詳細 >](#)
- AWS 機械学習ブログ**
機械学習のすべてに関する最新ニュースと更新情報をお読みください
[詳細 >](#)



IAM ユーザーとしてサインイン

アカウント ID (12 桁) またはアカウントエイリアス

ユーザー名:

パスワード:

☐ このアカウントを記憶する

[サインイン](#)

ルートユーザーの E メールを使用したサインイン
パスワードをお忘れですか?



Amazon Lightsail

Lightsail is the easiest way to get started on AWS

[Learn more »](#)

日本語

利用規約 プライバシーポリシー © 1996-2022, Amazon Web Services, Inc. or its affiliates.

サービス

サービス、特徴、ブログ、およびドキュメントを検索 [Alt+S]

東京

id

n@

コンソールのホーム 情報

最近アクセスしたサービス 情報

EC2

Cloud9

AWS Single Sign-On

VPC

Resource Access Manager

Route 53

CloudWatch

Direct Connect

AWS Organizations

AWS Cost Explorer

IAM

Systems Manager

Amazon Grafana

CloudFormation

AWS へようこそ

[AWS の開始方法](#)

AWS を最大限に活用するために基礎を学び、有益な情報を見つけましょう。

[トレーニングと認定](#)

AWS のエキスパートから学び、スキルと知識を深めましょう。

[AWS の最新情報](#)

新しい AWS のサービス、機能、およびリージョンについてご覧ください。

AWS Health 情報

未解決の問題

0

過去 7 日間

スケジュールされている変更

0

近日および過去 7 日間

その他の通知

0

過去 7 日間

コストと使用状況 情報

コストと使用状況なし

これは、AWS Cost Manager を設定していないか、許可が付与されていないことが原因である可能性があります。

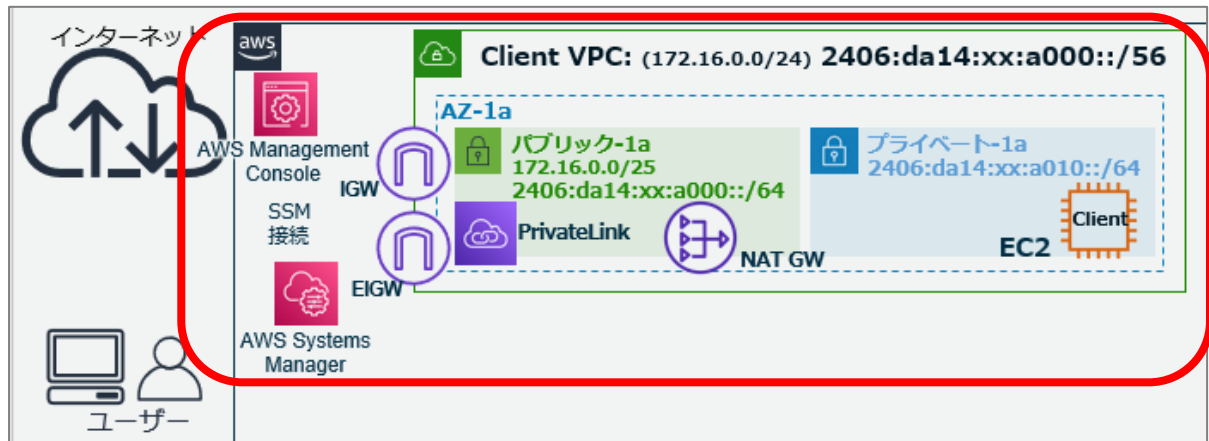
[AWS コスト管理に移動](#)

3. ハイブリッド VPC の作成

3.1. Client VPC 環境

本ハンズオン環境の全体像は以下の構成となっています。

初めに、Client VPC 環境を CloudFormation テンプレートを利用して作成します。



3.2. 踏み台用 VPC 環境の作成

CloudFormation により Client VPC 環境を作成します。

- ① AWS マネジメントコンソールにログインし、画面右上のリージョンセクターから東京リージョンを選択します。



- ② [サービス]-[管理とガバナンス]-[CloudFormation]をクリックします。（Tips：画面上の検索機能より、短縮名の“cfn”と検索すると、素早く見つかります）CloudFormation のダッシュボードが表示されたら、[スタックの作成]をクリックします。



- ③ “スタックの作成”の画面で、“テンプレートソース”から[テンプレートファイルのアップロード]にチェックを入れ、[ファイルの選択]をクリックします。

The screenshot shows the AWS CloudFormation console's 'Create Stack' wizard. On the left, a sidebar lists the steps: Step 1: Specify template (selected), Step 2: Specify stack details, Step 3: Configure stack options, and Step 4: Review. The main area is titled 'スタックの作成' (Create Stack). Under '前提条件 - テンプレートの準備' (Prerequisites - Prepare template), there are three radio buttons: 'テンプレートの準備完了' (Template preparation complete) which is selected, 'サンプルテンプレートを使用' (Use sample template), and 'デザイナーでテンプレートを作成' (Create template in Designer). Below this, the 'テンプレートの指定' (Specify template) section explains that templates are JSON or YAML files. Under 'テンプレートソース' (Template source), there are two radio buttons: 'Amazon S3 URL' and 'テンプレートファイルのアップロード' (Upload template file), with the latter being selected and highlighted by a red box. Below this, the 'テンプレートファイルのアップロード' (Upload template file) section shows a 'ファイルの選択' (Select file) button, which is also highlighted by a red box. To the right of this button, it says 'ファイルが選択されていません' (No file is selected). Below the button, it says 'JSON または YAML 形式のファイル' (File in JSON or YAML format). At the bottom right, there are 'キャンセル' (Cancel) and '次へ' (Next) buttons.

- ④ テンプレートファイルをアップロードする画面が表示されます。ハンズオン主催者から予め配布されているテンプレートファイルから“IPv6-HandsOn-Client.yaml”を指定します。（お使いのブラウザにより、表示が異なります。）

- ⑤ [ファイルの選択]欄の右に、選択したテンプレートファイル名が追記されている事を確認し、右下の[次へ]をクリックします。

CloudFormation > スタック > スタックの作成

ステップ 1
テンプレートの指定

スタックの作成

前提条件 - テンプレートの準備

テンプレートの準備
各スタックはテンプレートに基づきます。テンプレートとは、スタックを含む AWS リソースに関する設定情報を含む JSON または YAML ファイルです。

☒ テンプレートの準備完了 ☐ サンプルテンプレートを使用 ☐ デザイナーでテンプレートを作成

テンプレートの指定
テンプレートは、スタックのリソースおよびプロパティを表す JSON または YAML ファイルです。

テンプレートソース
テンプレートを選択すると、保存先となる Amazon S3 URL が生成されます。

☐ Amazon S3 URL ☒ テンプレートファイルのアップロード

テンプレートファイルのアップロード

ファイルの選択 **IPv6-HandsOn-Client.yaml**

JSON または YAML 形式のファイル

S3 URL: `https://s3-ap-northeast-1.amazonaws.com/cf-templates-1li6cdhhcjcjv-ap-northeast-1/202208787W-IPv6-HandsOn-Client.yaml` デザイナーで表示

キャンセル **次へ**

備考：テンプレートファイル名が表示されない場合には、ブラウザのリロード、マネジメントコンソールへの再ログインをお試しください。

- ⑥ "スタックの詳細を指定"の画面で、"スタックの名前"に"ipv6-handson-client"と入力し、[次へ]をクリックします。

スタックの詳細を指定

スタックの名前

スタックの名前

ipv6-handson-client

スタック名では、大文字および小文字 (A-Z~a-z)、数字 (0-9)、ダッシュ (-) を使用することができます。

パラメータ
パラメータは、テンプレートで定義されます。また、パラメータを使用すると、スタックを作成または更新する際にカスタム値を入力できます。

LinuxLatestAmi

/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

キャンセル 戻る **次へ**

- ⑦ “スタックオプションの設定”の画面で[次へ]をクリックします。

スタックオプションの設定

タグ

スタックのリソースに適用するタグ (キーと値のペア) を指定できます。スタックごとに一意のタグを 50 個まで追加できます。[詳細はこちら](#)

削除

タグの追加

アクセス許可

CloudFormation を使用して、スタックのリソースを作成、変更、削除する方法を明示的に定義する IAM ロールを選択します。ロールを選択しない場合、CloudFormation はユーザーの認証情報に基づき、アクセス許可を使用します。[詳細はこちら](#)

IAM ロール - オプション

スタックで実行されるすべてのオペレーションで使用する CloudFormation の IAM ロールを選択します。

iamRo... ▼

▼

削除

スタックの失敗オプション

プロビジョニング失敗時の動作

スタックの失敗のロールバック動作を指定します。[詳細はこちら](#)

☒ すべてのスタックリソースをロールバックする

スタックを最新の安定した状態にロールバックします。

☐ 正常にプロビジョニングされたリソースの保持

正常にプロビジョニングされたリソースの状態を保持し、失敗したリソースを最後の既知の安定した状態にロールバックします。最後の安定した状態が既知ではないリソースは、次のスタックオペレーション時に削除されます。

詳細オプション

通知オプションやスタックポリシーなど、スタックのオプションを追加設定することができます。[詳細はこちら](#)

▶ スタックポリシー

スタックの更新中の意図しない更新から保護するリソースを定義します。

▶ ロールバック設定

スタックの作成時および更新時にモニタリングする CloudFormation のアラームを指定します。オペレーションでアラームのしきい値を超過した場合、CloudFormation では値がロールバックされます。[詳細はこちら](#)

▶ 通知オプション

▶ スタックの作成オプション

キャンセル

戻る

次へ

aws

11

- ⑧ “レビュー ipv6-handson”画面で設定内容を確認します。最下部までスクロールし、[AWS CloudFormation によって IAM リソースがカスタム名で作成される場合があることを承認します。]のチェックボックスにチェックを入れ、右下の[スタックの作成]をクリックします。

レビュー ipv6-handson-client

ステップ 1: テンプレートの指定 編集

テンプレート

テンプレート URL
https://s3-ap-northeast-1.amazonaws.com/cf-templates-n5lyt1hdr3cz-ap-northeast-1/2022088KU8-IPv6-HandsOn-Client.yaml

スタックの説明
AWS IPv6 Hands-On Advance

予想コスト
[🔗](#)

(中略)

スタックの作成オプション

タイムアウト
-

削除保護
無効

▶ クイック作成リンク

機能

ⓘ The following resource(s) require capabilities: [AWS::IAM::Role]

このテンプレートには、Identity and Access Management (IAM) リソースが含まれています。これらのリソースを個別に作成し、それぞれに最小限必要な権限を与えるかどうか確認してください。さらに、カスタム名が付けられているか確認してください。カスタム名が、ご利用の AWS アカウント内で一意のものであることを確認してください。詳細は[こちら](#) [🔗](#)

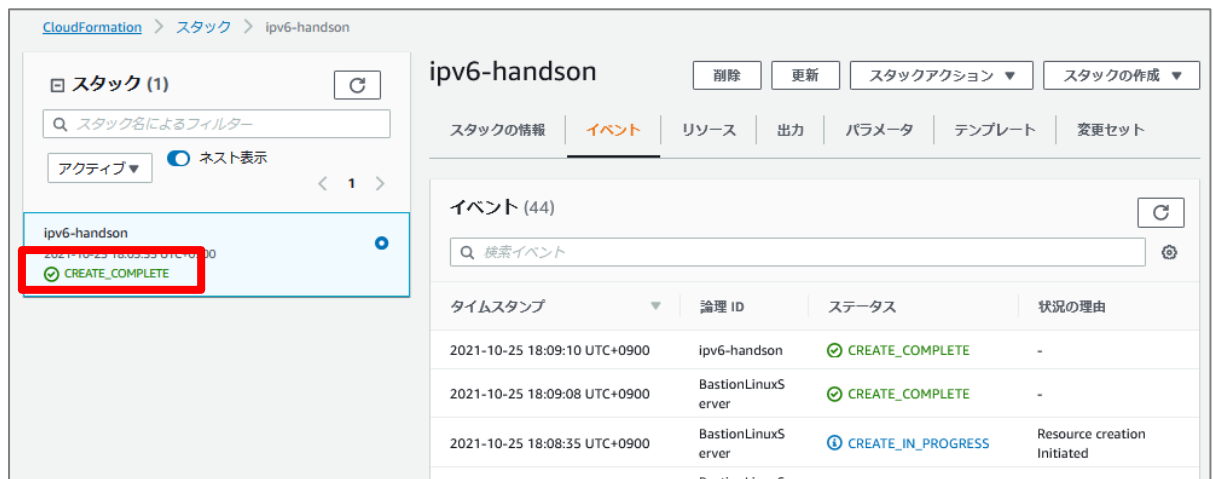
☒ AWS CloudFormation によって IAM リソースがカスタム名で作成される場合があることを承認します。

キャンセル 戻る 変更セットの作成 スタックの作成

- ⑨ 作成したスタックのステータスが **"CREATE_IN_PROGRESS"** の状態で表示されます。 **"CREATE_COMPLETE"** になるまで待ちます。



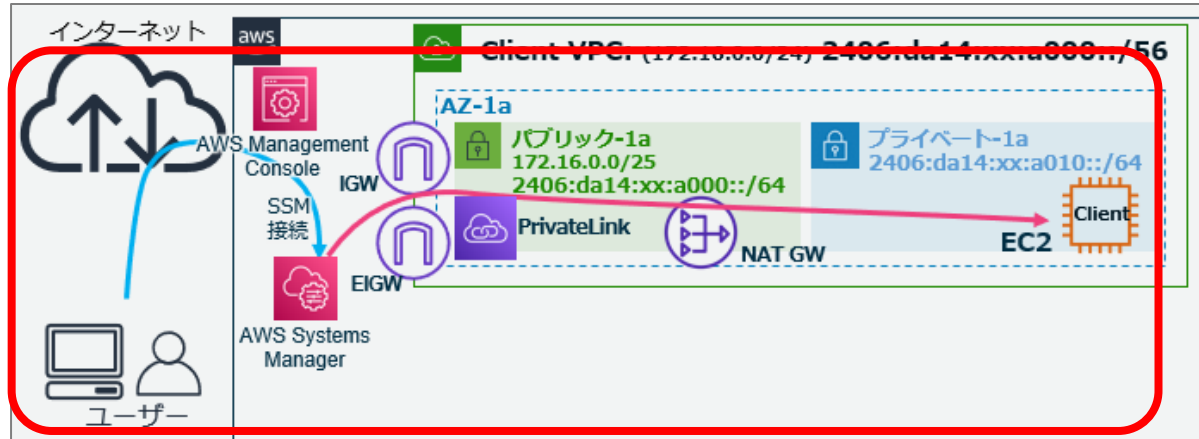
↓



この状態で、Client VPC 環境が作成されました。

3.3. Linux Client へのログイン

先ほど作成した Client VPC 環境には、すでに Linux OS が起動されています。この項目では、受講者の PC から、起動した Linux OS に SSM で接続します。



マネジメントコンソールの操作は、CloudFormation で Client VPC 環境のスタックが作成された画面から再開します。

- ① "リソース"タブをクリックし、論理 ID 列に"**BastionLinuxServer**"と表示される行を確認します。同じ行の"物理 ID"列にある EC2 インスタンスのリソース ID : "**i-xxxxxxx**"をクリックします。

論理 ID	物理 ID	タイプ
AttachGateway	ipv6-Attac-1GG0PXGKCWVK	AWS::EC2::VPCGatewayAttachment
BastionLinuxServer	i-0a7b5bce6e49709e6	AWS::EC2::Instance
ClientPrivateSubnet	subnet-096e70c3c69a119f3	AWS::EC2::Subnet

- ② EC2 ダッシュボードに移動し、起動した EC2 インスタンスが表示された状態になります。左側のラジオボタンにチェックを入れると、画面下に詳細情報が表示されます。左下の"インスタンス

スの状態が**実行中**となるまでお待ちください。

The screenshot shows the AWS Management Console interface for an instance. The instance name 'ipv6-handson-BastionLinuxServer' is highlighted with a red box. Below the instance list, the details for 'インスタンス: i-0a7b5bce6e49709e6 (ipv6-handson-BastionLinuxServer)' are displayed. The 'インスタンスの状態' (Instance State) is highlighted with a red box and shows '実行中' (Running) with a green checkmark icon.

Name	インスタンス ID	インスタンス...	インスタンス...	ステータス
<input checked="" type="checkbox"/> ipv6-handson-BastionLinuxServer	i-0a7b5bce6e49709e6	実行中	t3.micro	初期

インスタンス: i-0a7b5bce6e49709e6 (ipv6-handson-BastionLinuxServer)

上記からインスタンスを選択

詳細 | セキュリティ | ネットワーキング | ストレージ | ステータスチェック | モニタリング | タグ

▼ インスタンス概要 情報

インスタンス ID i-0a7b5bce6e49709e6 (ipv6-handson-BastionLinuxServer)	パブリック IPv4 アドレス -
IPv6 アドレス 2406:da14:3d:3901:1d8c:5a76:de77:487b	インスタンスの状態 実行中
ホスト名のタイプ リソース名: i-0a7b5bce6e49709e6.ap-northeast-1.compute.internal	プライベートリソース DNS 名 i-0a7b5bce6e49709e6.ap-northeast-1.compute.internal

③ インスタンスの状態が**実行中**となっていることを確認したのち、画面上の**[接続]**を選択します。

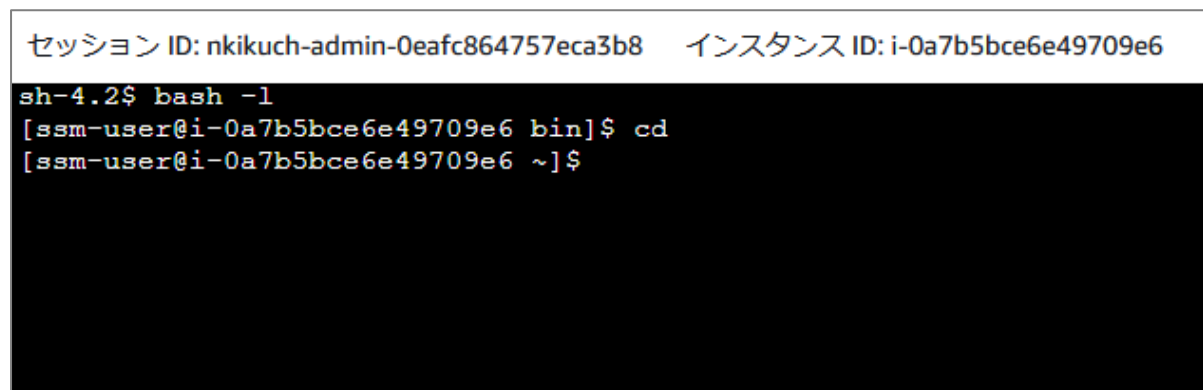
The screenshot shows the AWS Management Console interface for the same instance. The '接続' (Connect) button is highlighted with a red box. The instance status is still '実行中' (Running).

Name	インスタンス ID	インスタンス...	インスタンス...	ステータスチェック
<input checked="" type="checkbox"/> ipv6-handson-BastionLinuxServer	i-0a7b5bce6e49709e6	実行中	t3.micro	2/2 のチェックに合格しま

- ④ “インスタンスに接続”画面で、“セッションマネージャー”タブが選択されている（アンダーラインがついて、オレンジ色にハイライトされている状態）ことを確認し、右下の[接続]を選択します。



- ⑤ 以下の画面のように、黒いターミナル画面が表示されたら、接続成功です。インスタンス ID を確認するため、コマンド：“bash -l”（最後はハイフン、小文字のエル）を実行し、コマンド：“cd”でホームディレクトリに移動します。



- ⑥ コマンド：“ip addr show dev eth0”を実行して、Linux ホストに IPv6 の両方のアドレスが付与されていることを確認します。ここで表示された IPv6 アドレス(コマンド実行結果の“inet6 2406:…”に続く部分)は、後の確認で参照しますので、テキストエディタなどにメモしてください。なお、“inet 169.254.…”に IPv4 アドレスが付与されていますが、これは管理用に付与さ

れたリンクローカルアドレスとなり、実際の通信では利用されません。

ip address show dev eth0

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ ip address show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:fa:bd:44:a0:cd brd ff:ff:ff:ff:ff:ff
    inet 169.254.44.44/32 scope global dynamic eth0
        valid_lft 2863sec preferred_lft 2863sec
    inet6 2406:da14:3d:3901:1d8c:5a76:de77:487b/128 scope global dynamic
        valid_lft 425sec preferred_lft 115sec
    inet6 fe80::4fa:bdf:fe44:a0cd/64 scope link
        valid_lft forever preferred_lft forever
[ssm-user@i-0a7b5bce6e49709e6 ~]$
```

3.4. DNS64/NAT64 の動作確認

CloudFormation テンプレートを用いて作成した環境では、すでに DNS64/NAT64 を利用できる状態になっています。Linux へログインした状況でその動作を確認します。

- ① DNS64 が有効時の挙動を確認するため、コマンド：“dig checkip.amazonaws.com. aaaa +short”を実行します。以下の様に、“64:ff9b”が付与された IPv6 アドレスが返答されます。

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ dig checkip.amazonaws.com. aaaa +short
checkip.check-ip.aws.a2z.com.
checkip.eu-west-1.prod.check-ip.aws.a2z.com.
64:ff9b::22fc:6c05
64:ff9b::34d3:1585
64:ff9b::3648:1589
64:ff9b::22f8:c178
64:ff9b::34d4:a7e9
64:ff9b::364c:b498
64:ff9b::22f6:5d25
64:ff9b::3410:17d5
[ssm-user@i-0a7b5bce6e49709e6 ~]$
```

- ② DNS64 を無効にして挙動を確認します。セッションマネージャーで利用しているブラウザのタブとは別のタブで VPC ダッシュボードへアクセスし、ナビゲーションペインから[サブネット] > [ipv6-handson-ClientPrivateSubnet]を選択します。ここで、下部ペインを確認し、

対象サブネットに IPv4 がアサインされていないこと、および、DNS64 が有効化されていることを確認します。

subnets-0ef83f27d3fedd952 / ipv6-handson-client-ClientPrivateSubnet

詳細

<p>サブネット ID</p> <p>subnet-0ef83f27d3fedd952</p> <p>利用可能な IPv4 アドレス</p> <p>0</p> <p>ネットワークボーダーグループ</p> <p>ap-northeast-1</p> <p>デフォルトのサブネット</p> <p>いいえ</p> <p>お客様が所有する IPv4 プール</p> <p>-</p> <p>IPv6 のみ</p> <p>はい</p> <p>DNS64</p> <p>有効</p>	<p>サブネット ARN</p> <p>arn:aws:ec2:ap-northeast-1:500218855301:subnet/subnet-0ef83f27d3fedd952</p> <p>IPv6 CIDR</p> <p>2406:da14:5d0a:301::/64</p> <p>VPC</p> <p>vpc-0c7f4a245ceff8dea ipv6-handson-client-VPC</p> <p>パブリック IPv4 アドレスを自動割り当て</p> <p>いいえ</p> <p>Outpost ID</p> <p>-</p> <p>ホスト名のタイプ</p> <p>リソース名</p> <p>所有者</p> <p>500218855301</p>	<p>状態</p> <p>Available</p> <p>アベイラビリティゾーン</p> <p>ap-northeast-1a</p> <p>ルートテーブル</p> <p>rtb-044a1ffed74db8db ipv6-handson-client-PrivateRoute</p> <p>IPv6 アドレスを自動割り当て</p> <p>はい</p> <p>IPv4 CIDR の予約</p> <p>-</p> <p>リソース名 DNS A レコード</p> <p>無効</p>	<p>IPv4 CIDR</p> <p>-</p> <p>アベイラビリティゾーン ID</p> <p>apne1-az4</p> <p>ネットワーク ACL</p> <p>acl-0b7ee1b278d75d20e</p> <p>お客様が所有する IPv4 アドレスを自動割り当て</p> <p>いいえ</p> <p>IPv6 CIDR の予約</p> <p>-</p> <p>リソース名 DNS AAAA レコード</p> <p>有効</p>
---	---	--	--

③ [アクションメニュー]> [サブネットの設定を編集]を選択します。

サブネット (1/5) 情報

Q サブネットをフィルター

	Name	サブネット ID	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	ipv6-handson-ClientPublicSubnet	subnet-014f4ec492e3eee8c	172.16.0.0/25	2406:da14:5d0a:301::/64
<input checked="" type="checkbox"/>	ipv6-handson-ClientPrivateSubnet	subnet-096e70c3c69a119f3	-	2406:da14:5d0a:301::/64

アクション

- 詳細を表示
- フローログを作成
- サブネットの設定を編集
- IPv6 CIDR の編集
- ネットワーク ACL の関連付けの編集

④ [DNS64 を有効化]のチェックボックスを外し、[保存]を選択します。

DNS64 の設定

DNS64 を有効化して、Amazon VPC の IPv6 専用のサービスが IPv4 専用のサービスおよびネットワークと通信できるようにします。

☐ DNS64 を有効化 情報

キャンセル 保存

- ⑤ 再度、コマンド : "dig checkip.amazonaws.com. aaaa +short"を実行すると、IPv6 アドレスが入手できない状態になります。(IPv4 アドレスは解決可能です。)

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ dig checkip.amazonaws.com. aaaa +short
checkip.check-ip.aws.a2z.com.
checkip.eu-west-1.prod.check-ip.aws.a2z.com.
[ssm-user@i-0a7b5bce6e49709e6 ~]$
```

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ dig checkip.amazonaws.com. a +short
checkip.check-ip.aws.a2z.com.
checkip.eu-west-1.prod.check-ip.aws.a2z.com.
54.76.180.152
34.246.216.84
52.211.21.133
34.246.93.37
34.247.150.55
18.202.70.8
52.16.23.213
54.194.163.232
[ssm-user@i-0a7b5bce6e49709e6 ~]$
```

- ⑥ 改めて、DNS64 を有効化します。ナビゲーションペインから[サブネット]> [ipv6-handson-ClientPrivateSubnet]を選択> アクションメニュー> [サブネットの設定を編集]を選択します。

サブネット (1/5) 情報

サブネットをフィルター

	Name	サブネット ID	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	ipv6-handson-ClientPublicSubnet	subnet-014f4ec492e3eee8c	172.16.0.0/25	2400::0/25
<input checked="" type="checkbox"/>	ipv6-handson-ClientPrivateSubnet	subnet-096e70c3c69a119f3	-	2400::0/25

アクション

- 詳細を表示
- フローログを作成
- サブネットの設定を編集
- IPv6 CIDR の編集
- ネットワーク ACL の関連付けの編集

- ⑦ [DNS64 を有効化]のチェックボックスを選択し、[保存]を選択します。

DNS64 の設定

DNS64 を有効化して、Amazon VPC の IPv6 専用のサービスが IPv4 専用のサービスおよびネットワークと通信できるようにします。

☒ DNS64 を有効化 [情報](#)

キャンセル **保存**

- ⑧ 再度、コマンド : "dig checkip.amazonaws.com. aaaa +short"を実行し、IPv6 アドレスが入手できている事を確認します。

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ dig checkip.amazonaws.com. aaaa +short
checkip.check-ip.aws.a2z.com.
checkip.eu-west-1.prod.check-ip.aws.a2z.com.
64:ff9b::364c:b498
64:ff9b::3410:17d5
64:ff9b::3648:1589
64:ff9b::22f8:c178
64:ff9b::22f0:60ab
64:ff9b::34d3:1585
64:ff9b::22f6:d854
64:ff9b::22f7:9637
[ssm-user@i-0a7b5bce6e49709e6 ~]$
```

- ⑨ curl コマンドを利用して、IPv6 プロトコルで、外部のサイトへアクセスできることを確認します。（"-v6"は、リクエストの詳細表示と IPv6 を利用するオプションです。）

curl -v6 http://checkip.amazonaws.com/

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ curl -v6 http://checkip.amazonaws.com/
* Trying 64:ff9b::22fc:6c05:80...
* Connected to checkip.amazonaws.com (64:ff9b::22fc:6c05) port 80 (#0)
> GET / HTTP/1.1
> Host: checkip.amazonaws.com
> User-Agent: curl/7.79.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Mon, 28 Mar 2022 08:07:22 GMT
< Server: lighttpd/1.4.53
< Content-Length: 14
< Connection: keep-alive
<
13.230.58.255    ###<- NAT Gateway に関連付けられた EIP
* Connection #0 to host checkip.amazonaws.com left intact
[ssm-user@i-0a7b5bce6e49709e6 ~]$
```

- ⑩ 他の公開 Web サイトにも接続して、動作を確認します。

curl -v6 https://www.v6pc.jp/

```
[ssm-user@i-0a7b5bce6e49709e6 ~]$ curl -v https://www.v6pc.jp/
* Trying 2400:6700:ff00::36f8:67d4:443...
* Connected to www.v6pc.jp (2400:6700:ff00::36f8:67d4) port 443 (#0)
<<中略>>
> GET / HTTP/1.1
> Host: www.v6pc.jp
> User-Agent: curl/7.79.1
<<中略>>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="refresh" content="1;URL=./jp/index.phtml">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>IPv6 Promotion Council</title>
</head>
<body>
1 秒後に自動転送します。<br>
転送されない方はこちらをクリックして下さい。<br>
<a href="http://www.v6pc.jp/jp/">クリック</a>
</body>
</html>
* Connection #0 to host www.v6pc.jp left intact
sh-4.2$ curl -v6 https://www.v6pc.jp/sh-4.2$
```

- ⑪ 参考：その他の IPv6 接続を確認できるサイト

<https://www.kame.net/>

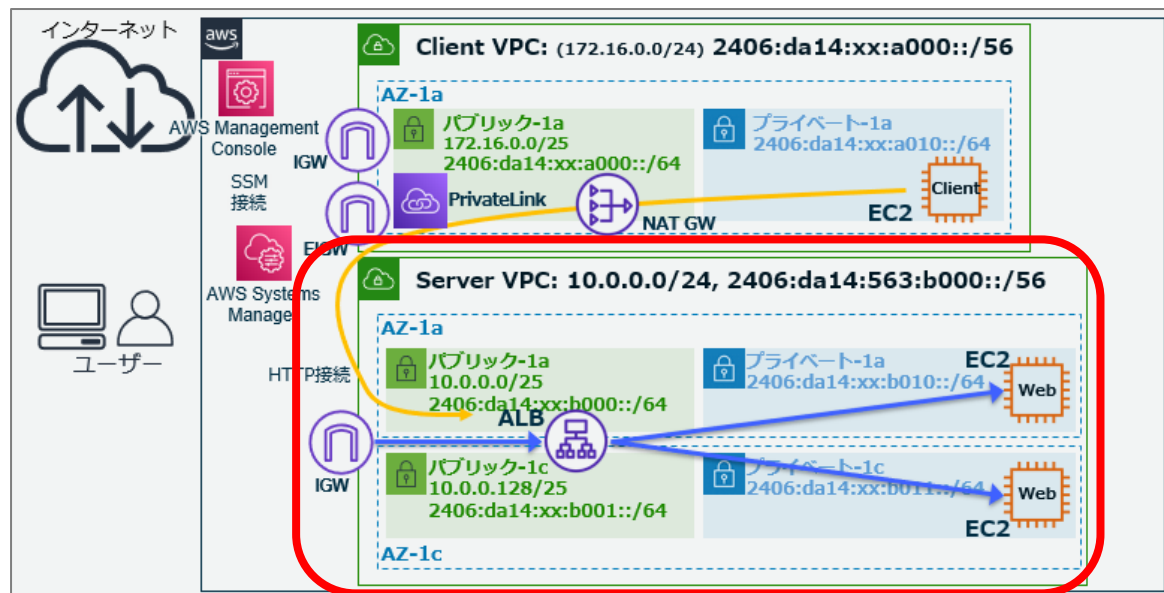
<https://www.kddi.com/>

ここまでで接続元環境のセットアップ、動作確認は完了です。Linux Client への SSM 接続環境は、後ほど利用します。SSM 接続で利用したブラウザのタブは、閉じずに保つことができますが、しばらくすると接続がタイムアウトされることがあります。その場合、ブラウザのリロード、もしくは SSM 接続で利用したブラウザのタブを閉じ、他のタブで再度[接続]をクリックして再接続してください。

4. 公開ウェブサーバー環境として接続先環境を作成

4.1. 接続先 VPC 環境

続いて、接続先 VPC 環境を作成します。こちらの VPC は受講者が手動で作成します。



4.2. VPC 内のネットワーク環境作成

VPC、サブネット、インターネットゲートウェイを作成し、各リソースがインターネットからアクセスできる環境を整えます。

- ① 画面右上のリージョンセクターにおいて、東京リージョンが選択されていることを確認します。
[サービス] > [ネットワーキングとコンテンツ配信] > [VPC]を選択し、VPC ダッシュボードを表示します。（検索機能から“VPC”を検索する方法でも可能）



② [VPC を作成]をクリックします。



③ VPC 作成画面では、ハイブリッド VPC とハイブリッドサブネットを作成します。現在のところ VPC ウィザードでは、IPv6 のみのサブネット作成に対応していません。後に記載した通りにパラメータを指定し、最後に[VPC を作成]を選択します。



設定項目	パラメータ	要修正
作成するリソース	VPC など	
名前タグの自動生成	ipv6-handson-server	レ
IPv4 CIDR ブロック	10.0.0.0/24	レ
IPv6 CIDR ブロック	Amazon 提供の IPv6 CIDR ブロック	レ
テナンシー	デフォルト	
アベイラビリティーゾーン (AZ)	2	
AZ のカスタマイズ	第 1 アベイラビリティーゾーン: ap-northeast-1a	
	第 2 アベイラビリティーゾーン: ap-northeast-1c	
パブリックサブネットの数	2	
プライベートサブネットの数	2	
サブネットの CIDR ブロックをカスタマイズ	ap-northeast-1a のパブリックサブネット CIDR ブロック 10.0.0.0/28	
	ap-northeast-1c のパブリックサブネット CIDR ブロック 10.0.0.16/28	
	ap-northeast-1a のプライベートサブネット CIDR ブロック 10.0.0.128/28	
	ap-northeast-1c のプライベートサブネット CIDR ブロック 10.0.0.144/28	
NAT ゲートウェイ (\$)	なし	
Egress Only インターネットゲートウェイ	はい	レ
VPC エンドポイント情報	S3 ゲートウェイ	
DNS オプション	DNS ホスト名を有効化 (チェックを入れる)	
	DNS 解決を有効化 (チェックを入れる)	

Egress Only インターネットゲートウェイは、後に Web サーバへ nginx をインストールする際に利用するものです。

- ④ 各リソースが正常に作成された旨が表示されます。[VPC を表示]を選択し、IPv4/IPv6 のデュアルスタック VPC が作成されていることを確認します。NAT ゲートウェイがアクティブ化する際には、少し時間が必要です。

VPC ワークフローの作成

成功

▼ 詳細

- 成功 VPC を作成: [vpc-0de64268832fc42f0](#)
- 成功 VPC IPv6 CIDR ブロックが関連付けられるのを待ちます
- 成功 DNS ホスト名を有効化
- 成功 DNS 解決を有効化
- 成功 VPC 作成の確認: [vpc-0de64268832fc42f0](#)
- 成功 S3 エンドポイントを作成: [vpce-0c8675e1b8137dbbb](#)
- 成功 サブネットを作成: [subnet-019bec2854896b745](#)
- 成功 サブネットを作成: [subnet-0a562a2cc223991f0](#)
- 成功 サブネットを作成: [subnet-092f112843dc424eb](#)
- 成功 サブネットを作成: [subnet-0bbe7b5f4e1fcccfe](#)
- 成功 インターネットゲートウェイの作成: [igw-097f9a96cea24e29b](#)
- 成功 インターネットゲートウェイを VPC にアタッチ
- 成功 ルートテーブルを作成: [rtb-09177ac8c113b8afc](#)
- 成功 ルートを作成
- 成功 ルートを作成
- 成功 ルートテーブルを関連付ける
- 成功 ルートテーブルを関連付ける
- 成功 Elastic IP を割り当て: [eipalloc-06bf6447f8675eac5](#)
- 成功 NAT ゲートウェイを作成: [nat-0e8dd6788661c7879](#)
- 成功 NAT ゲートウェイがアクティブ化されるのを待ちます
- 成功 ルートテーブルを作成: [rtb-034126c1b7e080e4a](#)
- 成功 ルートを作成
- 成功 ルートテーブルを関連付ける
- 成功 ルートテーブルを作成: [rtb-03aaed115a51d2e64](#)
- 成功 ルートを作成
- 成功 ルートテーブルを関連付ける
- 成功 ルートテーブルの作成を検証中
- 成功 S3 エンドポイントとプライベートサブネットのルートテーブルを関連付ける: [vpce-0c8675e1b8137dbbb](#)

VPC を表示

VPC > お使いの VPC > vpc-0ec991b3f0a4d96db

vpc-0ec991b3f0a4d96db / ipv6-handson-server-vpc アクション ▼

詳細 情報

VPC ID vpc-0ec991b3f0a4d96db	状態 ✔ Available	DNS ホスト名 有効	DNS 解決 有効
テナンシー Default	DHCP オプションセット dopt-05db3fdaf10a75584	メインルートテーブル rtb-0492a235a9c37698c	メインネットワーク ACL acl-0ff89b12a135c57cf
デフォルト VPC いいえ	IPv4 CIDR 10.0.0.0/24	IPv6 プール Amazon ✔ Associated	IPv6 CIDR (ネットワーク ボーダーグループ) 2406:da14:68a:d600::/56 ✔ Associated
Route 53 リゾルバー DNS ファイアウォールルールレ グループ -	所有者 ID 258341223113		

CIDR | フローログ | タグ

CIDRs 情報

Address type ▲	CIDR	Network Border ...	プール	ステータス
IPv4	10.0.0.0/24	-	-	✔ Associated
IPv6	2406:da14:68a:d600::/56	ap-northeast-1	Amazon	✔ Associated

- ⑤ ナビゲーションペインから[サブネット]を選択します。以下のように 4 つのサブネットが作成されていることを確認します。

New VPC Experience
Tell us what you think

VPC ダッシュボード
EC2 Global View new
VPC でフィルタリング:
Q VPC の選択

VIRTUAL PRIVATE
CLOUD
VPC
サブネット
ルートテーブル

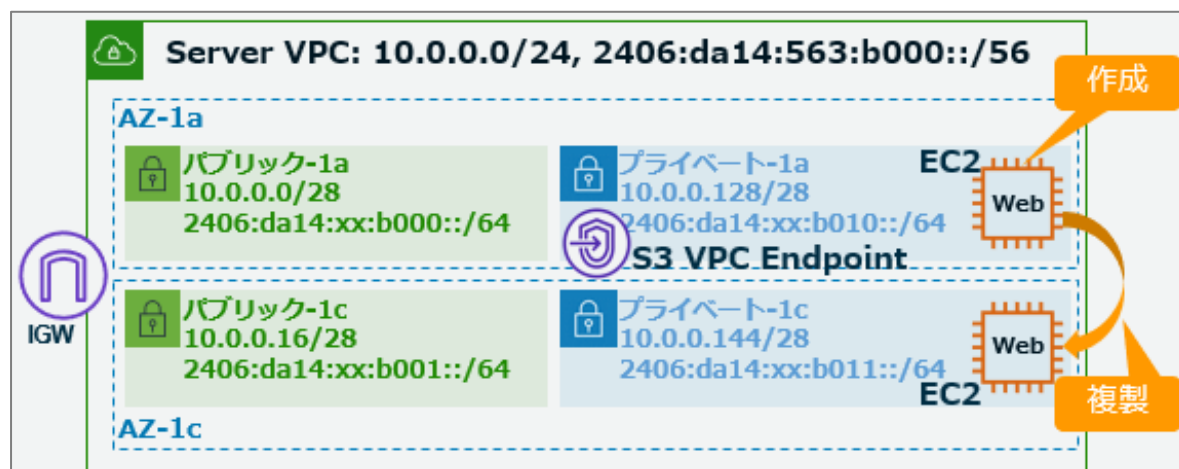
サブネット (4) 情報

Q サブネットをフィルター
search: ipv6-handson-server X フィルターをクリア

	Name	サブネット ID	IPv4 CIDR	IPv6 CIDR	利用可能な IPv4 アドレス
<input type="checkbox"/>	ipv6-handson-server-subnet-public1-ap-northeast-1a	subnet-0058163584257900b	10.0.0.0/28	2406:da14:68a:d600::/64	10
<input type="checkbox"/>	ipv6-handson-server-subnet-public2-ap-northeast-1c	subnet-016f7c1349dc3c670	10.0.0.16/28	2406:da14:68a:d601::/64	11
<input type="checkbox"/>	ipv6-handson-server-subnet-private1-ap-northeast-1a	subnet-0e0be7276974a6a6b	10.0.0.128/28	2406:da14:68a:d602::/64	11
<input type="checkbox"/>	ipv6-handson-server-subnet-private2-ap-northeast-1c	subnet-085a21f6ceb2bdf04	10.0.0.144/28	2406:da14:68a:d603::/64	11

4.3. Web サーバの配置

先ほど作成したプライベートサブネットに、Web サーバを作成、その後、他のサブネットに複製します。



- ① 画面左上の[サービス] > [コンピューティング]>“EC2”をクリックし、EC2 ダッシュボードに移動します。



- ② 画面中央から[インスタンスを起動] > [インスタンスの起動]をクリックします。

The screenshot shows the AWS Management Console interface. On the left sidebar, the 'EC2 ダッシュボード' (EC2 Dashboard) is highlighted with a red box. The main content area is divided into several sections. At the top, there's a 'リソース' (Resources) section showing a table of resources. Below that, there's a section titled 'インスタンスを起動' (Launch Instance) with a red box around the 'インスタンスを起動' button. To the right of this, there's a 'サービス状態' (Service Status) section showing the status of the AWS services.

リソース	数
インスタンス (実行中)	1
Elastic IP	4
インスタンス	1
キーペア	1
スナップショット	1
セキュリティグループ	4
プレースメントグループ	0
ボリューム	1
ロードバランサー	0
専用ホスト	0

サービス状態

リージョン	ステータス
アジアパシフィック (東京)	このサービスは正常に動作しています

- ③ 名前とタグの [名前]に"ipv6-handson-server-1a"を入力します。

The screenshot shows the 'インスタンスを起動' (Launch Instance) page in the AWS Management Console. The breadcrumb navigation at the top reads 'EC2 > インスタンス > インスタンスを起動'. The main heading is 'インスタンスを起動' with a '情報' (Information) link. Below the heading, there's a paragraph explaining that you can create an Amazon EC2 instance. The '名前とタグ' (Name and Tags) section is highlighted, and within it, the '名前' (Name) field is highlighted with a red box and contains the text 'ipv6-handson-server-1a'. There is also a link 'さらにタグを追加' (Add more tags) next to the name field.

- ④ “アプリケーションおよび OS イメージ (Amazon マシンイメージ)”では“Amazon Linux”が選択されていることを確認し、“インスタンスタイプ”で、“t3.micro”へ変更します。(デフォルトの t2 シリーズでは、IPv6 のみの IP アドレス形態に対応していないので注意)

▼ アプリケーションおよび OS イメージ (Amazon マシンイメージ) 情報

AMI は、インスタンスの起動に必要なソフトウェア設定 (オペレーティングシステム、アプリケーションサーバー、アプリケーション) を含むテンプレートです。お探しのものが以下に表示されない場合は、AMI を検索または参照してください。

最新

クイックスタート

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE

SUSE

その他の AMI を閲覧する

AWS、Marketplace、コミュニティからの AMI を含む

Amazon マシンイメージ (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-0de5311b2a443fb89 (64 ビット (x86)) / ami-082dd9d89994d3690 (64 ビット (Arm))

仮想化: hvm ENA 有効: true ルートデバイスタイプ: ebs

無料利用枠の対象

説明

Amazon Linux 2 Kernel 5.10 AMI 2.0.20221004.0 x86_64 HVM gp2

アーキテクチャ

AMI ID

64 ビット (x86)

ami-0de5311b2a443fb89

検証済みプロバイダー

▼ インスタンスタイプ 情報

インスタンスタイプ

t3.micro

Family: t3 2 vCPU 1 GiB メモリ

オンデマンド Linux 料金: 0.0136 USD 1 時間あたり

オンデマンド Windows 料金: 0.0228 USD 1 時間あたり

Compare instance types

- ⑤ “キーペア (ログイン)”では“キーペアなしで続行 (推奨されません)”を選択します。

▼ キーペア (ログイン) 情報

キーペアを使用してインスタンスに安全に接続できます。インスタンスを起動する前に、選択したキーペアにアクセスできることを確認してください。

キーペア名 - 必須

キーペアなしで続行 (推奨されません)

デフォルト値 ▼

新しいキーペアの作成

- ⑥ “ネットワーク設定”では右上の[編集]を押し、詳細な設定項目を表示します。



▼ ネットワーク設定 情報

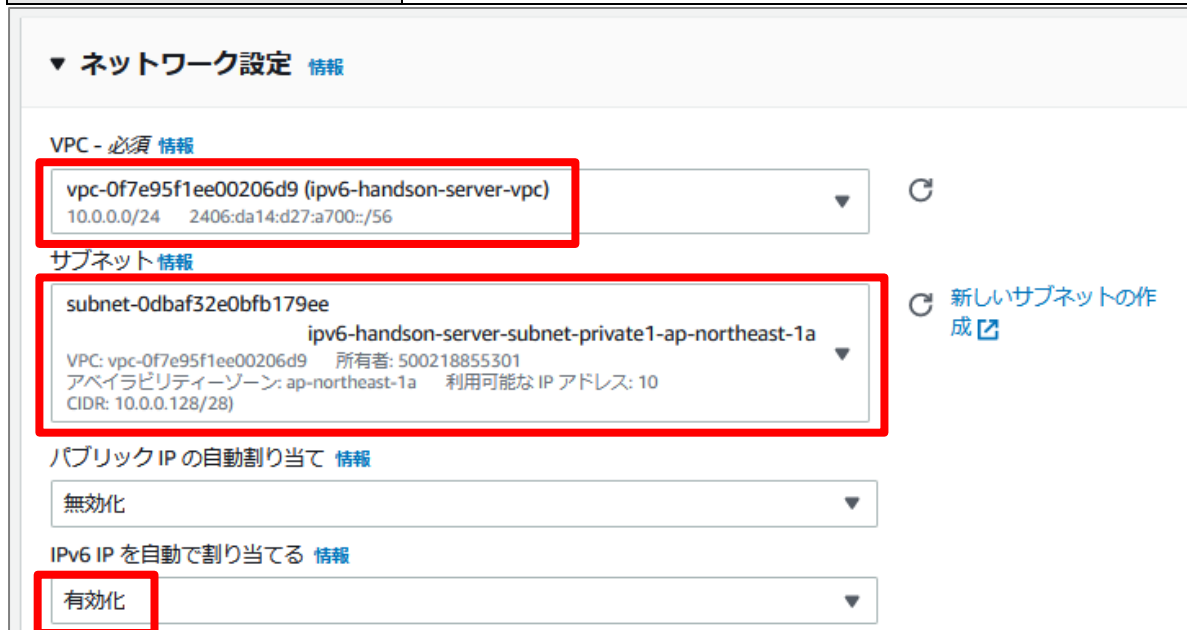
ネットワーク 情報
vpc-0b92530d8d0790cce

サブネット 情報
優先順位なし (アベイラビリティゾーンのデフォルトサブネット)

パブリック IP の自動割り当て 情報
有効化

- ⑦ “ネットワーク設定”が展開され、詳細な設定が可能になります。各項目について、以下の表を参照し変更します。表の後に記載されている変更後のキャプチャ画像と比較し確認してください。

設定項目	パラメータ
ネットワーク	“ipv6-handson- server-vpc ”の名前がついた VPC ID
サブネット	“ipv6-handson-server-subnet- private1 -ap-northeast- 1a ” の名前がついた Subnet ID
パブリック IP の自動割り当て	無効化
IPv6 IP を自動で割り当てる	有効化
セキュリティグループルール 2	タイプ : HTTP Source: ::/0 (IPv6 のデフォルトルート)



▼ ネットワーク設定 情報

VPC - 必須 情報
vpc-0f7e95f1ee00206d9 (ipv6-handson-server-vpc)
10.0.0.0/24 2406:da14:d27:a700::/56

サブネット 情報
subnet-0dbaf32e0bfb179ee
ipv6-handson-server-subnet-private1-ap-northeast-1a
VPC: vpc-0f7e95f1ee00206d9 所有者: 500218855301
アベイラビリティゾーン: ap-northeast-1a 利用可能な IP アドレス: 10
CIDR: 10.0.0.128/28

パブリック IP の自動割り当て 情報
無効化

IPv6 IP を自動で割り当てる 情報
有効化

<中略>

▼ セキュリティグループルール 1 (TCP, 22, 0.0.0.0/0) 削除

タイプ 情報: ssh
プロトコル 情報: TCP
ポート範囲 情報: 22
Source type 情報: Anywhere
Source 情報: 0.0.0.0/0, ::/0
説明 - optional 情報: 例 管理者のデスクトップの SSH

▼ セキュリティグループルール 2 (TCP, 80, ::/0) 削除

タイプ 情報: HTTP
プロトコル 情報: TCP
ポート範囲 情報: 80
Source type 情報: Custom
Source 情報: ::/0
説明 - optional 情報: 例 管理者のデスクトップの SSH

セキュリティグループルールを追加

▶ 高度なネットワーク設定

最初に[セキュリティグループルールを]を押して、タイプ・Source を指定する

- ⑧ “高度な詳細”の左側にある“▶”をクリックし展開します。最下部にあるユーザーデータに主催者から配布されたテキストデータ“userdata.txt”の中身をコピー＆ペーストします。

▼ 高度な詳細 情報

購入オプション 情報
☐ スポットインスタンスをリクエスト
オンデマンド料金を上限とするスポット料金でスポットインスタンスをリクエスト

ドメイン結合ディレクトリ 情報
選択 新しいディレクトリの作成

IAM インスタンスプロフィール 情報
選択 新しい IAM プロファイルの作成

<中略>

ユーザーデータ 情報

```
#!/bin/bash
amazon-linux-https disable
amazon-linux-extras install -y nginx1 php7.2
service nginx start
chkconfig nginx on
echo "<html><h1>HTTP_X_FORWARDED_FOR: ""<?php echo
\$_SERVER['HTTP_X_FORWARDED_FOR']; ?>""</html></h1>" >> /usr/share/nginx
/html/index2.php
```

☐ ユーザーデータは既に base64 エンコードされています

ユーザーデータ

ご注意：別途配布しているテキストファイル“userdata.txt”からコピー＆ペーストしてください。

```
#!/bin/bash
amazon-linux-https disable
amazon-linux-extras install -y nginx1 php7.2
service nginx start
chkconfig nginx on
echo "<html><h1>HTTP_X_FORWARDED_FOR: ""<?php echo
\$_SERVER['HTTP_X_FORWARDED_FOR']; ?>""</html></h1>"
>> /usr/share/nginx/html/index2.php
```

- ⑨ 最下部の“概要”までスクロールし、右下の[**インスタンスを起動**]をクリックします。

▼ 概要

インスタンス数 情報

1

ソフトウェアイメージ (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[続きを読む](#)
ami-0de5311b2a443fb89

垂直サーバータイプ (インスタンスタイプ)

t2.micro

ファイアウォール (セキュリティグループ)

新しいセキュリティグループ

ストレージ (ボリューム)

1 ボリューム - 8 GiB

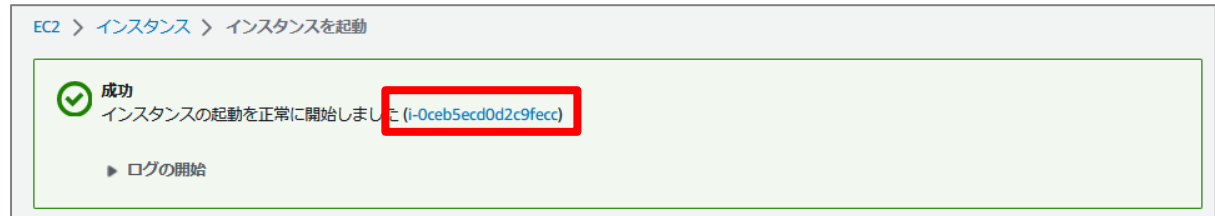
キャンセル

インスタンスを起動

aws

33

- ⑩ インスタンスの作成が開始された旨のメッセージが表示されるので、**割り当てられたリソース ID** をクリックして、次のステップにすすみます。起動したホストが表示されない場合、インスタンスのフィルタリングを解除し、リロードボタンをお試しください。



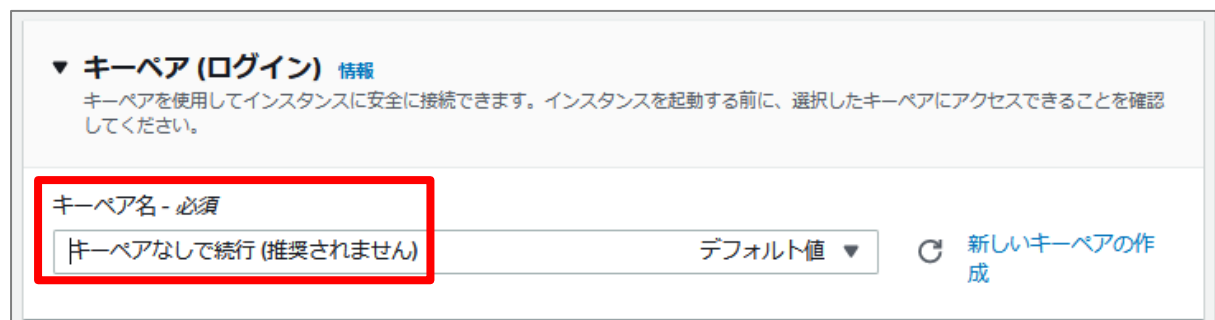
- ⑪ インスタンスが実行中のステータスになったら、他方のサブネットにも Web サーバを起動するために複製します。起動した EC2 インスタンス ID の左にあるチェックボックスにチェックを入れ、"アクション"メニュー>"イメージとテンプレート">"同様のものを起動"を選択します。



- ⑫ "名前とタグ"で"名前"を確認し、末尾を 1a から"1c"へ修正します。



- ⑬ "キーペア"で、"キーペアなしで続行 (推奨されません)"を選択します。



- ⑭ ネットワーク設定で、サブネットを“private2-ap-northeast-1c”の名前が付いた ID に変更します。“IPv6 を自動で割り当てる”を“有効化”に変更します。

▼ ネットワーク設定 情報

VPC - 必須 情報

vpc-0de64268832fc42f0 (ipv6-handson-server-vpc)
10.0.0.0/24 2406:da14:781:1f00::/56

サブネット 情報

subnet-0bbe7b5f4e1fcccfe
ipv6-handson-server-subnet-private2-ap-northeast-1c
VPC: vpc-0de64268832fc42f0 所有者: 500218855301
アベイラビリティゾーン: ap-northeast-1c 利用可能な IP アドレス: 10
CIDR: 10.0.0.144/28

パブリック IP の自動割り当て 情報

無効化

IPv6 IP を自動で割り当てる 情報

有効化

- ⑮ その他は既存のまま最下部の“概要”までスクロールし、“**インスタンスを起動**”を選択します。

▼ 概要

インスタンス数 情報

1

ソフトウェアイメージ (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[続きを読む](#)
ami-0de5311b2a443fb89

垂直サーバータイプ (インスタンスタイプ)

t3.micro

ファイアウォール (セキュリティグループ)

launch-wizard-2

ストレージ (ボリューム)

1 ボリューム - 8 GiB

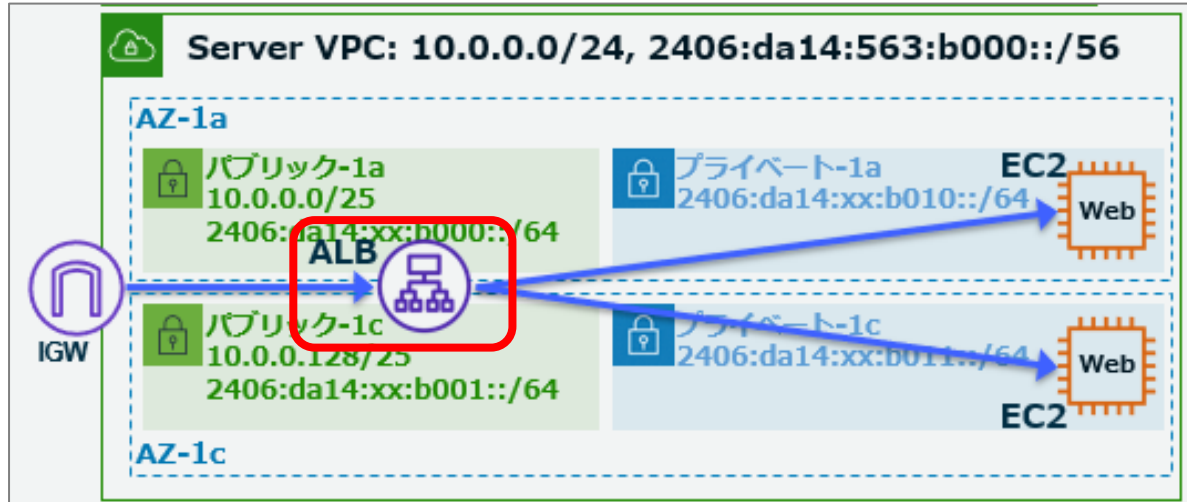
キャンセル

インスタンスを起動

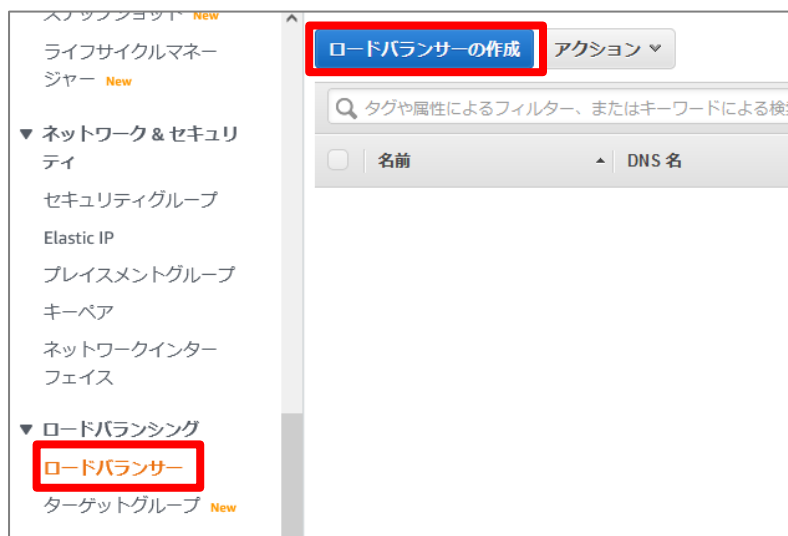
ここまでの作業で、二つの Private サブネットにそれぞれ EC2 インスタンスが起動し、nginx による Web サイトが起動しています。

4.4. Elastic Load Balancer (ALB)の設定

この章では、冗長可用に配置された Web サーバに対し、トラフィックを配分可能にするため、ロードバランサーの導入を行います。



- ① EC2 ダッシュボードのナビゲーションペインで、[ロードバランサー] > [ロードバランサーの作成]をクリックします。



- ② “Application Load Balancer”を示す図の下にある[作成]をクリックします。

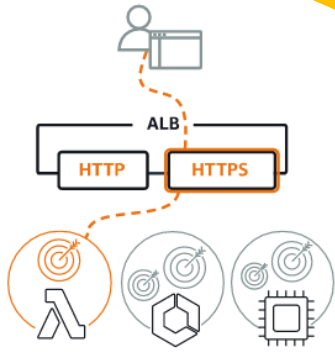
EC2 > ロードバランサー > ロードバランサータイプの選択

ロードバランサータイプの選択

特徴ごとの完全な比較と詳細なハイライトも利用できます。 [詳細はこちら](#)

ロードバランサータイプ

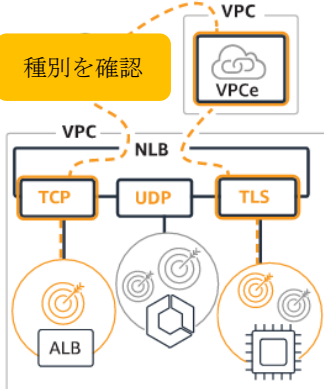
Application Load Balancer 情報



HTTP および HTTPS トラフィックを使用するウェブアプリケーション用に柔軟性の高い機能セットが必要な場合は、Application Load Balancer を選択します。Application Load Balancer はリクエストレベルで動作し、マイクロサービスとコンテナを含む、アプリケーションアーキテクチャを対象とした高度なルーティングおよび可視性機能を提供します。

作成


Network Load Balancer 情報



非常に高いパフォーマンス、大規模な TLS のオフロード、証明書のデプロイの一元管理、UDP のサポート、およびアプリケーションの静的 IP アドレスが必要な場合は、Network Load Balancer を選択します。Network Load Balancer は接続レベルで動作し、パフォーマンスを維持しながら、大量のトラフィックのリクエストを確実に処理することができます。

作成

Gateway Load Balancer 情報



GENEVE をサポートするサードパーティの仮想アプライアンスのフリートをデプロイおよび管理する必要がある場合は、Gateway Load Balancer を選択します。これらのアプライアンスを使用すると、セキュリティ、コンプライアンス、ポリシー制御を向上させることができます。

作成

▶ Classic Load Balancer - 前の世代

- ③ “ロードバランサー名”に[**ipv6-handson-alb**]と入力します。“IP アドレスタイプ”のラジオボタンで [**dualstack**]を選択します。

EC2 > ロードバランサー > Application Load Balancer を作成

Application Load Balancer を作成 [情報](#)

Application Load Balancer は、リクエスト属性に基づいて、着信 HTTP および HTTPS トラフィックを複数のターゲット (Amazon EC2 インスタンス、マイクロサービス、コンテナなど) に分散します。ロードバランサーは接続リクエストを受け取ると、優先度順にリスナールールを評価して適用するルールを決定し、該当する場合は、ルールアクションのターゲットグループからターゲットを選択します。

▶ Application Load Balancer の仕組み

基本的な設定

ロードバランサー名
名前は AWS アカウント内で一意である必要があり、ロードバランサーの作成後に変更することはできません。

ipv6-handson-alb

ハイフンを含む最大 32 文字の英数字を使用できますが、名前の先頭または末尾にハイフンを使用することはできません。

スキーム [情報](#)
ロードバランサーの作成後にスキームを変更することはできません。

☒ インターネット向け
インターネット向けロードバランサーは、クライアントからのリクエストをインターネット経由でターゲットにルーティングします。これには、パブリックサブネットが必要です。 [詳細はこちら](#)

☐ 内部
内部ロードバランサーは、プライベート IP アドレスを使用して、クライアントからターゲットにリクエストをルーティングします。

IP アドレスタイプ [情報](#)
サブネットで使用する IP アドレスのタイプを選択します。

☐ IPv4
社内向けロードバランサーに推奨。

☒ Dualstack
IPv4 と IPv6 アドレスが含まれます。

- ④ 画面下にスクロールし、“VPC”のプルダウンメニューから[**ipv6-handson-server**]の名前が設定された ID を選択します。“マッピング”では、表示された **ap-northeast-1a/1c** の 2 つのチ

エックボックスにチェックを入れ、“サブネット”から“**public-1a/1c**”の名前が付いたサブネットをそれぞれ選択します。その他の項目はデフォルトのままとします。

⑤

Network mapping 情報

ロードバランサーは、IP アドレス設定に従って、選択したサブネットのターゲットにトラフィックをルーティングします。

VPC 情報

ターゲットの仮想プライベートクラウド (VPC) を選択します。インターネットゲートウェイを持つ VPC のみが選択可能になります。選択した VPC は、ロードバランサーの作成後に変更できません。ターゲットの VPC を確認するには、[ターゲットグループ](#) を表示します。

ipv6-handson-server-vpc
vpc-09f2a01d9e93246ea
IPv4: 10.0.0.0/24 IPv6: 2406:da14:74f:d000::/56

↻

マッピング 情報

少なくとも 2 つの Availability Zone と、ゾーンごとに 1 つのサブネットを選択します。ロードバランサーは、これらの Availability Zone のターゲットにのみトラフィックをルーティングします。ロードバランサーまたは VPC でサポートされていない Availability Zone は選択できません。

☒ ap-northeast-1a

サブネット

subnet-0f48c3caba639413d ipv6-handson-server-subnet-public1-ap-northeast-1a ▼

IPv4 の設定

AWS によって割り当て済み

IPv6 の設定

IPv6 アドレス

CIDR 2406:da14:74f:d000::/64 から割り当て済み

☒ ap-northeast-1c

サブネット

subnet-01279abe50ea1f021 ipv6-handson-server-subnet-public2-ap-northeast-1c ▼

IPv4 の設定

AWS によって割り当て済み

IPv6 の設定

IPv6 アドレス

CIDR 2406:da14:74f:d001::/64 から割り当て済み

- ⑥ 更にスクロールし、“セキュリティグループ”で、[新しいセキュリティグループの作成]を選択します。

セキュリティグループ 情報
セキュリティグループは、ロードバランサーへのトラフィックを制御する一連のファイアウォールルールです。

セキュリティグループ

セキュリティグループの選択

新しいセキュリティグループの作成

default sg-0f5c2de52e83255ae X
VPC: vpc-00f1ef3753c3179b4

- ⑦ ブラウザで別タブが開き、“セキュリティグループを作成”へ遷移します。“基本的な詳細”では、以下を入力します。VPC の指定では、入力欄の右側にあるバツ印（×）を押すと一覧が表示されます。

設定項目	パラメータ	要修正
セキュリティグループ名	ipv6-alb-sg	レ
説明	allow-http	レ
VPC	“ipv6-handson-server-vpc”の名前が付いた ID を選択	レ

EC2 > セキュリティグループ > セキュリティグループを作成

セキュリティグループを作成 情報

セキュリティグループは、インスタンスの仮想ファイアウォールとして機能し、インバウンドトラフィックとアウトバウンドトラフィックをコントロールします。新しいセキュリティグループを作成するには、以下のフィールドに入力してください。

基本的な詳細

セキュリティグループ名 情報
ipv6-alb-sg
作成後に名前を編集することはできません。

説明 情報
allow-http

VPC 情報

Q

vpc-0b92530d8d0790cce (デフォルト)
172.31.0.0/16

vpc-0c7f4a245cef8deaa (ipv6-handson-client-VPC)
172.16.0.0/24 2406:da14:5d0:a300::/56

vpc-0f7e95f1ee00206d9 (ipv6-handson-server-vpc)
10.0.0.0/24 2406:da14:d27:a700::/56

ルールがありません。

ルールを追加

- ⑧ 下にスクロールし、“インバウンドルール”では[ルールを追加]をクリックし、以下の2つのルールを追加します。2つのルールが追加できたら、同画面の右下にある[セキュリティグループを作成]をクリックします。

インバウンドルール 情報

このセキュリティグループにはインバウンドルールがありません。

ルールを追加

ルール(1)

設定項目	パラメータ	要修正
タイプ	HTTP	レ
リソースタイプ	Anywhere-IPv4	レ
ソース	0.0.0.0/0	自動入力

ルール(2)

設定項目	パラメータ	要修正
タイプ	HTTP	レ
リソースタイプ	Anywhere-IPv6	レ
ソース	:::/0	自動入力

インバウンドルール 情報

インバウンドルール 1 削除

タイプ 情報: HTTP
 プロトコル 情報: TCP
 ポート範囲 情報: 80
 リソースタイプ 情報: Anywhere-IPv4
 ソース 情報: 0.0.0.0/0 X
 説明 - オプション 情報:

インバウンドルール 2 削除

タイプ 情報: HTTP
 プロトコル 情報: TCP
 ポート範囲 情報: 0
 リソースタイプ 情報: Anywhere-IPv6
 ソース 情報: :::/0 X
 説明 - オプション 情報:

キャンセル **セキュリティグループを作成**

- ⑨ 先ほど作業していた“**Application Load Balancer を作成**”のタブに戻り、右側の更新ボタンをクリックしたのち、セキュリティグループのプルダウンメニューから、**[IPv6-ALB-SG]**を選択します。その後、**[default]**グループを×印をクリックして削除します。（タブの移動後、タイムアウトにより事前に入力していた情報がクリアされていた場合、改めて入力します）

セキュリティグループ 情報
セキュリティグループは、ロードバランサーへのトラフィックを制御する一連のファイアウォールルールです。

セキュリティグループ

セキュリティグループの選択

Q

ipv6-alb-sg VPC: vpc-0203e22f9a26397a2	sg-03e19cc77c87330fe
default VPC: vpc-0203e22f9a26397a2	sg-071e855ba4ffd587f
launch-wizard-1 VPC: vpc-0203e22f9a26397a2	sg-0c92370a6b6dda1fb

セキュリティグループ 情報
セキュリティグループは、ロードバランサーへのトラフィックを制御する一連のファイアウォールルールです。

セキュリティグループ

最大5個のセキュリティグループを選択

新しいセキュリティグループの作成

default sg-0f77e3d73cacdd5e5 X 削除

ipv6-alb-sg sg-0d6429f42cc0b07a4 X

VPC: vpc-0de64268832fc42f0

VPC: vpc-0de64268832fc42f0

- ⑩ 更にスクロールし、“リスナーとルーティング”で**[ターゲットグループの作成]**をクリックします。

リスナーとルーティング 情報
リスナーは、設定したポートとプロトコルを使用して接続リクエストをチェックするプロセスです。リスナー用に定義するルールによって、登録されたターゲットにロードバランサーがリクエストをルーティングする方法が決まります。

▼ リスナー HTTP:80

削除

プロトコル: HTTP

ポート: 80

デフォルトアクション: 転送先

ターゲットグループの選択

ターゲットグループの作成

リスナータグ - 省略可能
リスナーにタグを追加することを確認してください。タグを使用すると、AWS リソースを分類できるため、リソースをより簡単に管理できます。

リスナータグの追加

タグは最大 50 個追加できます。

リスナーの追加

- ⑪ お使いのブラウザで新たなタブとして“グループの詳細の指定”が表示されます。“ターゲットタイプの選択”では、[IP アドレス]を選択します。“ターゲットグループ名”には[ipv6-handson-tg]と入力します。“IP アドレスタイプ”にて[IPv6]のラジオボタンをクリックします。“VPC”では[ipv6-handson-vpc]が選択されていることを確認します。

EC2 > ターゲットグループ > ターゲットグループの作成

ステップ 1
グループの詳細の指定

ステップ 2
ターゲットの登録

グループの詳細の指定

ロードバランサーは、ターゲットグループのターゲットにリクエストをルーティングし、ターゲットでヘルスチェックを実行します。

基本的な設定

このセクションの設定は、ターゲットグループの作成後に変更することはできません。

ターゲットタイプの選択

☐ インスタンス

- 特定の VPC 内のインスタンスへのロードバランシングをサポートします。
- EC2 容量を管理およびスケールするために [Amazon EC2 Auto Scaling](#) をより円滑に使用できるようにします。

☒ IP アドレス

- VPC およびオンプレミスのリソースへのロードバランシングをサポートします。
- 同じインスタンス上の複数の IP アドレスとネットワークインターフェイスへのルーティングを容易にします。
- マイクロサービスベースのアーキテクチャに柔軟性を提供し、アプリケーション間の通信を簡素化します。
- IPv6 ターゲットをサポートし、エンドツーエンドの IPv6 通信と IPv4 から IPv6 への NAT を有効にします。

☐ Lambda 関数

- 単一の Lambda 関数へのルーティングを容易にします。
- Application Load Balancer にのみアクセスできます。

☐ Application Load Balancer

- Network Load Balancer に柔軟性を提供して、特定の VPC 内で TCP リクエストを受け入れてルーティングします。
- Application Load Balancer で静的 IP アドレスおよび PrivateLink を使用することを容易にします。

ターゲットグループ名

ipv6-handson-tg

ハイフンを含む最大 32 文字の英数字を使用できますが、名前の先頭または末尾にハイフンを使用することはできません。

プロトコル ポート

HTTP 80

IP アドレスタイプ

指定された IP アドレスタイプのターゲットのみを、このターゲットグループに含めることができます。

☐ IPv4

☒ IPv6

VPC

ロードバランサーをホストする VPC を選択します。上記で選択した IP アドレスタイプをサポートする VPC のみが、このリストに表示されます。
Register targets サービスは、この VPC から IP アドレスを登録できます。

ipv6-handson-server-vpc
vpc-0de64268832fc42f0
IPv4: 10.0.0.0/24 IPv6: 2406:da14:781:1f00::/56

プロトコルバージョン

☒ HTTP1

HTTP/1.1 を使用してターゲットにリクエストを送信します。これはリクエストプロトコルが HTTP/1.1 または HTTP/2 の場合にサポートされます。

☐ HTTP2

HTTP/2 を使用してターゲットにリクエストを送信します。これはリクエストプロトコルが HTTP/2 または gRPC の場合にサポートされますが、gRPC 固有の機能は使用できません。

☐ gRPC

gRPC を使用してターゲットにリクエストを送信します。これはリクエストプロトコルが gRPC の場合にサポートされます。

- ⑫ 更にスクロールし、右下の[次へ]をクリックします。

ヘルスチェック

関連付けられたロードバランサーは、以下の設定ごとに、登録済みターゲットのステータスをテストするため、登録済みターゲットに対して定期的にリクエストを送信します。

ヘルスチェックプロトコル

HTTP ▼

ヘルスチェックパス

デフォルトパス「/」を使用してルートに ping を実行するか、必要に応じてカスタムパスを指定します。

/

最大文字数は 1024 です。

▶ ヘルスチェックの詳細設定

▶ タグ - 省略可能

ターゲットグループにタグを追加することを検討してください。タグを使用すると、AWS リソースを分類できるため、リソースをより簡単に管理できます。

キャンセル **次へ**

- ⑬ “ターゲットの登録”では“ステップ 2: IP を指定してポートを定義する”の項目で、[**インスタンスの詳細による IP アドレスを検索**]を選択します。続いて下に表示された二つのインスタンス候

補について左端のラジオボタンを両方チェックし、[保留中として以下を含める]をクリックします。

ターゲットの登録

これは、ターゲットグループを作成するためのオプションのステップです。ただし、ロードバランサーがこのターゲットグループにトラフィックをルーティングするようにするには、ターゲットを登録する必要があります。

IP アドレス

ステップ 1: ネットワークを選択する

ターゲットグループ用に選択した VPC の IP アドレス、または VPC 外の IP アドレスを追加できます。このステップに戻り、別のネットワークを選択することで、複数のネットワークソースのターゲットを組み合わせたことができます。

ネットワーク

ipv6-handson-server
vpc-0203e22f9a26397a2
IPv6: 2406:da14:2c5:8800::/56

ステップ 2: IP を指定してポートを定義する

選択したネットワークの IP アドレスを手動で入力できます。または、接続されたインスタンスリソースの詳細に従って IP アドレスを検索および選択することもできます。アタッチされたリソースが変更された場合でも、IP ターゲットグループは引き続き IP アドレスにトラフィックをルーティングすることに注意してください。

☐ IP アドレスを手動で入力

☒ インスタンスの詳細による IP アドレスを検索

インスタンスにアタッチされた IPv6 アドレス (2/2)

検索 インスタンス ID、インスタンス名、セキュリティグループ、ゾーン、またはサブネット ID をフィルタリング

<input checked="" type="checkbox"/>	IPv6 アドレス	インスタンス ID	名前	状態	セキュリティグループ	ゾーン	サブネット ID
<input checked="" type="checkbox"/>	2406:da14:2c5:8811:3466:1b4c:5f3a:64f9	i-0d58feb2ef39fe14	ipv6-handson-server-1c	running	launch-wizard-1	ap-northeast-1c	subnet-0bf23de9aabb4059b
<input checked="" type="checkbox"/>	2406:da14:2c5:8810:d9ce:55c:d063:25ea	i-0c8914e566bbc7400	ipv6-handson-server-1a	running	launch-wizard-1	ap-northeast-1a	subnet-02b7707f7bd17587b

2 個を選択済み

ポート
このターゲットへのルーティング用のポート。

80

1~65535 (複数のポートをカンマで区切ります)

保留中として以下を含める

- ⑭ 画面を下へスクロールすると、先ほど保留中含めた二つのホストが IPv6 アドレスとして登録されています。ヘルスステータスは保留中のままで、画面右下の[ターゲットグループの作成]をクリックします。

ターゲットを確認

ステップ 3: グループに含める IP ターゲットを確認する
ターゲットグループに含める IP ターゲットを確認します。このページのステップ 1 と 2 を繰り返して、IP ターゲットをさらに追加します。ターゲットグループを作成した後、追加のターゲットを登録することもできます。

ターゲット (2) 保留中のすべての削除

すべて < 1 > ⚙

IPv6 アドレスを削除	ヘルスステータス	IP アドレス	ポート	ゾーン
X	保留中	2406:da14:2c5:8810:d9ce:55c:d063:25ea	80	ap-northeast-1a
X	保留中	2406:da14:2c5:8811:3466:1b4c:5f3a:64f9	80	ap-northeast-1c

2 個保留中 キャンセル 戻る ターゲットグループの作成

- ⑮ ターゲットグループが作成されたメッセージが表示されます。現在作業しているウェブブラウザのタブから、先ほどロードバランサーを作成していたタブに戻ります。

🔔 ターゲットグループが正常に作成されました: `ipv6-handson-tg`

EC2 > ターゲットグループ

ターゲットグループ (1) 情報 🔄 アクション ▼ ターゲットグループの作成

< 1 > ⚙

<input type="checkbox"/>	名前	ARN	ポート	プロトコル	ターゲット
<input type="checkbox"/>	ipv6-handson-tg	arn:aws:elasticloadbalancing...	80	HTTP	IP

- ⑩ “リスナーとルーティング”にてデフォルトアクションのターゲットグループを選択する箇所の右にある[更新ボタン]をクリックします。プルダウンメニューで先ほど作成した[ipv6-handson-tg]が表示されますので、これを選択します。

リスナーとルーティング 情報

リスナーは、設定したポートとプロトコルを使用して接続リクエストをチェックするプロセスです。リスナー用に定義するルールによって、登録されたターゲットにロードバランサーがリクエストをルーティングする方法が決まります。

▼ リスナー HTTP:80

削除

プロトコル

HTTP ▼

ポート

80

1~65535

デフォルトアクション

情報

転送先

ターゲットグループの選択

ターゲット

Q |

ipv6-handson-tg

ターゲットの種類: IP, IPv6

HTTP

リスナータグ - 省略可能

リスナーにタグを追加することを検討してください。タグを使用

リスナータグの追加

タグは最大 50 個追加できます。

リスナーの追加

- ⑰ 更にスクロールし、画面右下の[ロードバランサーの作成]をクリックします。

概要

設定を確認して確定します。 [コストの見積もり](#)

<h4>基本的な設定 編集</h4> <p>ipv6-handson-alb</p> <ul style="list-style-type: none"> インターネット向け Dualstack 	<h4>セキュリティグループ 編集</h4> <ul style="list-style-type: none"> ipv6-alb-sg sg-04421b36c3836cd4c 	<h4>ネットワークマッピング 編集</h4> <p>VPC</p> <p>vpc-0ec991b3f0a4d96db ipv6-handson-server-vpc</p> <ul style="list-style-type: none"> ap-northeast-1a subnet-0058163584257900b ipv6-handson-server-subnet-public1-ap-northeast-1a ap-northeast-1c subnet-016f7c1349dc3c670 ipv6-handson-server-subnet-public2-ap-northeast-1c 	<h4>リスナーとルーティング 編集</h4> <ul style="list-style-type: none"> HTTP:80 次をデフォルトにする: ipv6-handson-tg
<h4>アドオンサービス 編集</h4> <p>なし</p>	<h4>タグ 編集</h4> <p>なし</p>		

属性

特定のデフォルト属性がロードバランサーに適用されます。ロードバランサーの作成後に表示および編集できます。

キャンセル

ロードバランサーの作成

- ⑱ ロードバランサーが正常に作成された旨を表示する画面へ推移します。右下の[View load balancer]をクリックします。

ロードバランサーを正常に作成しました: ipv6-handson-alb

注意: ロードバランサーが完全にセットアップされてトラフィックのルーティング準備が整うまでに数分かかる場合があります。また、ターゲットの登録処理が完了して最初のヘルスチェックに合格するまでに数分かかる場合があります。

EC2 > ロードバランサー

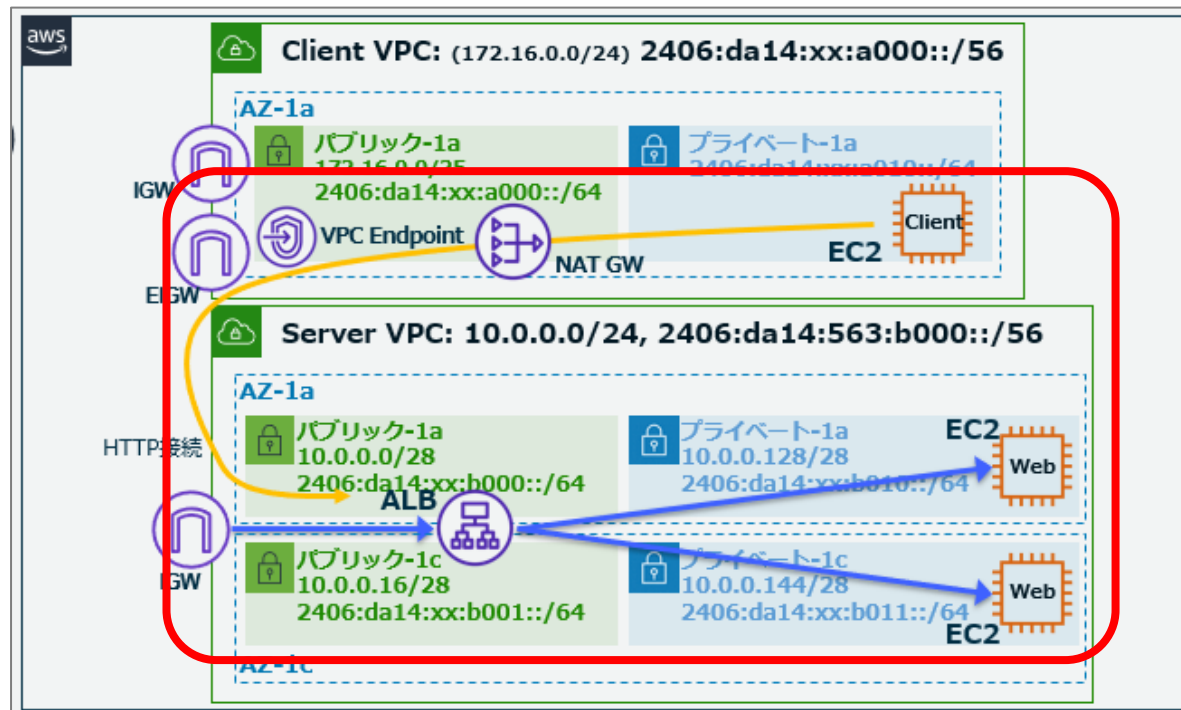
推奨される次のステップ

- ipv6-handson-alb 内の [説明] タブと [リスナー] タブを使用して、ロードバランサーおよびリスナーの属性を確認、カスタマイズ、有効化します。
- ロードバランサーと統合できる他のサービスを見つけてみましょう。ipv6-handson-alb にある [統合サービス] タブにアクセスしてください。

View load balancer

4.5. ALB への接続確認

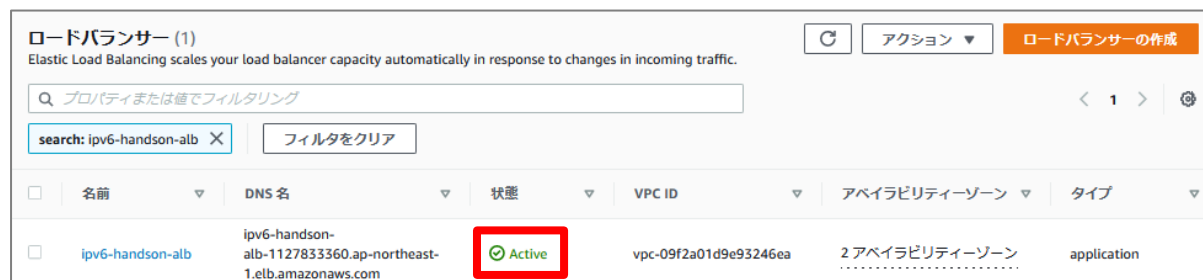
Client VPC の Linux OS から、作成した ALB を経由し、バックエンドの Web サーバへ接続できることを確認します。



- ① マネジメントコンソールの EC2 ダッシュボードで、先ほどの手順で作成したロードバランサーを表示します。“状態”列が“Provisioning”から“Active”に推移するまで待ちます。しばらく待っても状態に変化が無い場合、右上のリロードボタンをクリックします。



↓



- ② DNS 名列に表示されている“ipv6-handson-alb-***.amazonaws.com”という FQDN（文字列）をクリップボードにコピーします。なお、この DNS 名は後の章、DNS 設定でも利用します。受講者のパソコン内でテキストエディタなどを利用し、保存しておいてください。

ロードバランサー (1)						
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.						
<input type="text" value="プロパティまたは値でフィルタリング"/> 検索: ipv6-handson-alb フィルタをクリア						
<input type="checkbox"/>	名前	DNS 名	状態	VPC ID	アベイラビリティゾーン	タイプ
<input type="checkbox"/>	ipv6-handson-alb	ipv6-handson-alb-1127833360.ap-northeast-1.elb.amazonaws.com	Active	vpc-09f2a01d9e93246ea	2 アベイラビリティゾーン	application

- ③ Client VPC 内の Linux OS へ SSM 接続しているタブへ移動し、コピーした FQDN に対して curl コマンドで接続します。Nginx のテストページを示すキーワードが表示されていれば成功です。

curl -v6 http://ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com/ | head

```
[ssm-user@i-0a22092bc4b32884b ~]$ curl -v6 http://ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com/ | head
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
  0   0    0    0    0    0     0      0  0:00:00  0:00:00 --:--:--   0*   Trying
2406:da14:68a:d600:f15b:3084:c36:9fc8:80...
* Connected to ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com
(2406:da14:68a:d600:f15b:3084:c36:9fc8) port 80 (#0)
> GET / HTTP/1.1
> Host: ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com
> User-Agent: curl/7.79.1
> Accept: */*
<<省略>>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /*<![CDATA[*
    body {
      background-color: #fff;
[ssm-user@i-0a22092bc4b32884b ~]$
```

- ④ 前の手順では ALB の DNS 名を利用して、Web サーバへ接続できることを確認できました。アクセス元も IPv6 を使っていることを確認するため、アクセス元 IP アドレスを表示可能な URL へアクセスします。先ほどの手順で利用した URL の最後に[/index2.php]を加えて再度アクセ

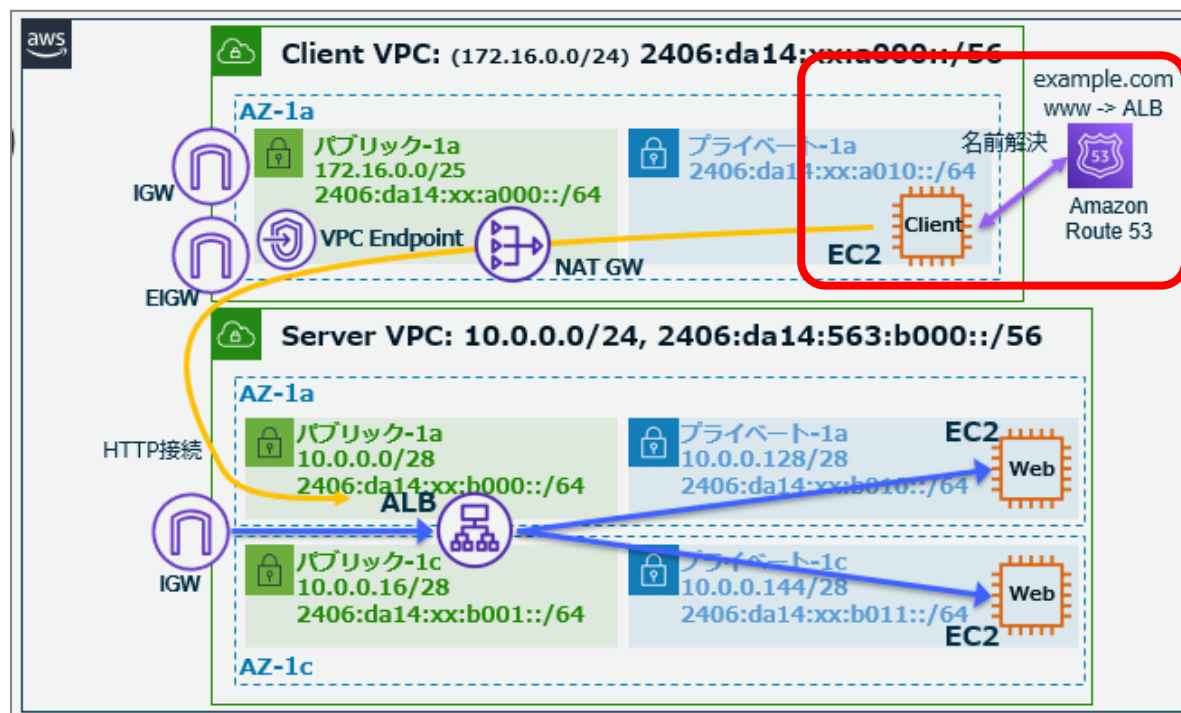
スします。以下のように接続元 Linux OS にアサインされている IPv6 アドレスが表示されていれば、ALB 接続に際し、IPv6 を使ってアクセスしている状態になります。

curl -v6 http://ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com/index2.php

```
[ssm-user@i-0a22092bc4b32884b ~]$ curl -v6 http://ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com/index2.php
* Trying 2406:da14:68a:d601:8397:a006:4f10:d9c:80...
* Connected to ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com
(2406:da14:68a:d601:8397:a006:4f10:d9c) port 80 (#0)
> GET /index2.php HTTP/1.1
> Host: ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com
> User-Agent: curl/7.79.1
> Accept: */*
<<省略>>
<html><h1>HTTP_X_FORWARDED_FOR: 2406:da14:854:fc01:6e23:22ee:720f:81ac</html></h1>
* Connection #0 to host ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com left intact
[ssm-user@i-0a22092bc4b32884b ~]$
```

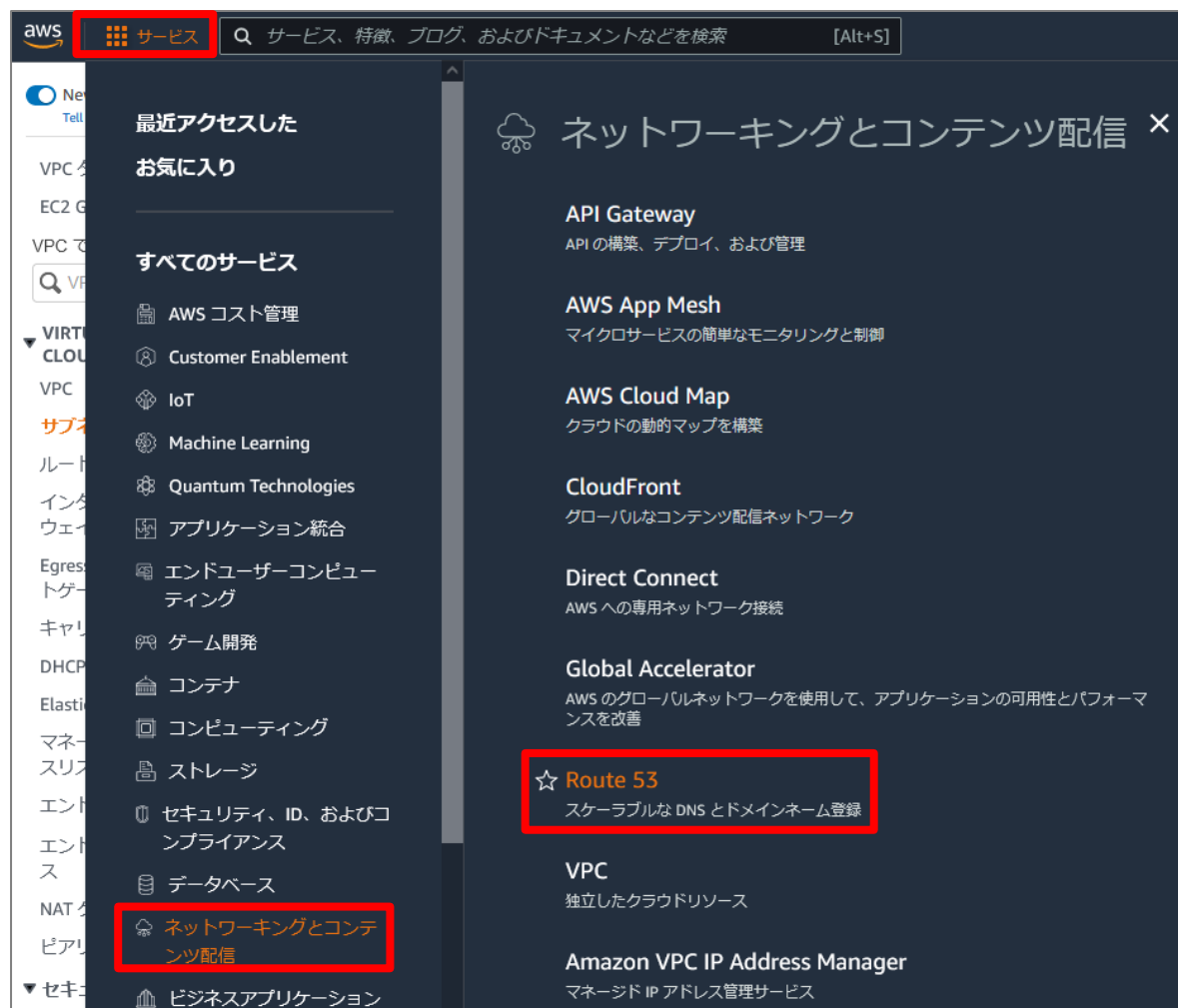
5. DNS で IPv6 を FQDN に登録する

この章では、作成した ALB に対し、Client VPC 内の Linux OS から独自ドメイン名でアクセスする設定を追加します。なお、ここで設定するドメイン名は Client VPC 内のみで名前解決可能です。受講者の手元にあるパソコンからは、同じ名前を利用することはできません。



5.1. Route 53 でプライベートホストゾーンの登録

- ① マネジメントコンソールで[サービス] > [ネットワーキングとコンテンツ配信] > [Route 53] を選択します。



- ② Route 53 のダッシュボードにて[ホストゾーンの作成]をクリックします。なお、以下の画面にならない場合、左側のナビゲーションペインから[ホストゾーン] > [ホストゾーンの作成]をクリックします。（環境により「Route53 はページを更新できませんでした。」という API エラ

ーメッセージや、複数の機能に「エラー」というメッセージが表示されます。本ハンズオンでの影響はありませんので、無視してください。)



- ③ “ドメイン名”の入力欄に本ハンズオンでのみ利用する任意のドメイン名を入力します。特に希望が無い場合、[**example.com**]を入力してください。“タイプ”では[**プライベートホストゾーン**]を選択、“ホストゾーンに関連付ける VPC”では、[**アジアパシフィック(東京)**]、“VPC ID”では

[ipv6-handson-client-VPC]の名前が付けられた ID を選択してください。右下の[ホストゾーンの作成]をクリックします。

Route 53 > ホストゾーン > ホストゾーンの作成

ホストゾーンの作成 情報

ホストゾーン設定

ホストゾーンは、example.com などのドメインおよびそのサブドメインのトラフィックのルーティング方法に関する情報を保持するコンテナです。

ドメイン名 情報
これは、トラフィックをルーティングするドメインの名前です。

example.com

有効な文字 : a-z、0-9、! * \$ % & ' () * + , - / : ; < = > ? @ [\] ^ _ ` { | } . ~

説明 - オプション 情報
この欄で、同じ名前のホストゾーンを区別できます。

ホストゾーンは次の目的で使用されます。

説明は最大 256 文字です。0/256

タイプ 情報
このタイプは、インターネットまたは Amazon VPC でトラフィックをルーティングするかどうかを示します。

☐ パブリックホストゾーン
パブリックホストゾーンは、インターネットのトラフィックのルーティング方法を決定します。

☒ プライベートホストゾーン
プライベートホストゾーンは、Amazon VPC 内でのトラフィックのルーティング方法を決定します。

ホストゾーンに関連付ける VPC 情報

このホストゾーンを使用して 1 つ以上の VPC の DNS クエリを解決するには、当該の VPC を選択します。別の AWS アカウントで作成された VPC をホストゾーンに関連付けるには、AWS CLI などのプログラムの方法を用いる必要があります。

☒ プライベートホストゾーンに関連付ける各 VPC に対して、Amazon VPC 設定 `enableDnsHostnames` および `enableDnsSupport` を true に設定する必要があります。

リジョン 情報 **VPC ID 情報**

アジアパシフィック (東京) [ap-nor... ▼]

IPv6-Client-VPC の名前がついた VPC

タグ 情報
ホストゾーンにタグを適用して、整理、識別しやすくします。

このリソースに関連付けられたタグがありません。

タグは最大であと 50 個追加できます。

キャンセル

- ④ 作成したホストゾーン内に CNAME レコードを追加します。[レコードを作成]をクリックします。

Route 53 > ホストゾーン > example.com

example.com 情報

ゾーンを削除する レコードをテストする クエリログの設定

▶ ホストゾンの詳細 ホストゾーンを編集

レコード (2) | ホストゾンのタグ (0)

レコード (2) 情報

Automatic モードは最適なフィルタ結果に最適化された現在の検索動作です。モードを変更するには、[設定] に移動します。

🔄 レコードを削除 ゾーンファイルをインポート レコードを作成

🔍 プロパティまたは値でレコードをフィルタリングする タイプ ▼ ルーティン... ▼ エイリアス ▼ < 1 > ⚙️

<input type="checkbox"/>	レコード名 ▼	タ... ▼	ルーテ... ▼	差別... ▼	値/トラフィックのルーティング先 ▼
<input type="checkbox"/>	example.com	NS	シンプル	-	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.
<input type="checkbox"/>	example.com	SOA	シンプル	-	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

- ⑤ “レコード名”に任意の名前（特に希望が無ければ[**www**]）を入力します。その他の項目は以下の通り入力します。右下の[**レコードを作成**]をクリックします。

設定項目	パラメータ	要修正
レコード名	www（他の任意のホスト名でも可）	レ
レコードタイプ	CNAME - 別のドメイン名および一部の AWS リソースにトラフィックをルーティングします。	レ
値： ALB 作成後に受講者パソコン内に保存した ALB の DNS 名をコピー＆ペーストします	ipv6-handson-alb-xxxxxxxxxx.ap-northeast-1.elb.amazonaws.com	レ
TTL	300	
ルーティングポリシー	シンプルルーティング	

Route 53 > ホストゾーン > example.com > レコードを作成

▼ レコードの作成方法

レコードのクイック作成 情報 ウィザードに切り替える

▼ レコード 1 削除

レコード名 情報 レコードタイプ 情報

www .example.com CNAME - 別のドメイン名および一部の AWS リソースにトラフィックを...

有効な文字: a-z、0-9、! * # \$ % & ' () * +, - / : ; < = > ? @ [\] ^ _ ` { | } . ~

値 情報 ☒ エイリアス

ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com

複数の値を個別の行に入力します。

TTL (秒) 情報 ルーティングポリシー 情報

300 シンプルルーティング

1 分 1 時間 1 日

推奨値: 60~172800 (2 日間)

別のレコードを追加

キャンセル **レコードを作成**

参考 : Amazon Route 53 のエイリアスレコードは使わないの？

Amazon Route 53 では AWS 独自の機能である“エイリアスレコード”にて ELB のリソース ID を登録することが可能です。現時点では、エイリアスレコードで ELB をホスト登録した場合 IPv4 アドレスのみが名前解決対象となります。本ハンズオンでは IPv4 と IPv6 のデュアルスタックアドレスで名前解決を題材としているため、CNAME レコードとして登録する手順を採用しています。

- ⑥ ALB の FQDN が“www.example.com”の CNAME として登録されていることを確認します。

Route 53 > ホストゾーン > example.com

example.com [情報](#) [ゾーンを削除する](#) [レコードをテストする](#) [クエリログの設定](#)

▶ **ホストゾーンの詳細** [ホストゾーンを編集](#)

レコード (3) | ホストゾーンのタグ (0)

レコード (3) 情報
Automatic モードは最適なフィルタ結果に最適化された現在の検索動作です。モードを変更するには、[設定] に移動します。

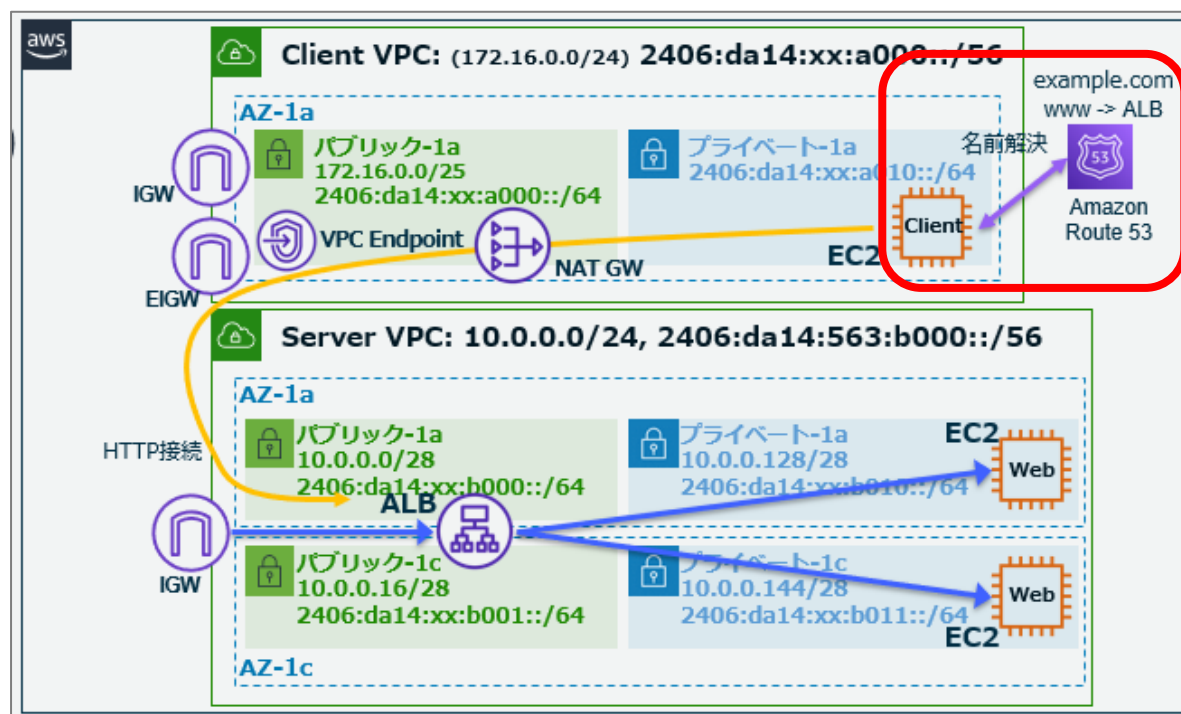
[🔄](#) [レコードを削除](#) [ゾーンファイルをインポート](#) [レコードを作成](#)

🔍 プロパティまたは値でレコードをフィルタリングする タイプ ▼ ルーティン... ▼ エイリアス ▼ < 1 > [⚙️](#)

<input type="checkbox"/>	レコード名 ▼	タ... ▼	ルーテ... ▼	差別... ▼	値/トラフィックのルーティング先 ▼
<input type="checkbox"/>	example.com	NS	シンプル	-	ns-1536.awsdns-00.co.uk, ns-0.awsdns-00.com, ns-1024.awsdns-00.org, ns-512.awsdns-00.net.
<input type="checkbox"/>	example.com	SOA	シンプル	-	ns-1536.awsdns-00.co.uk awsdns-hostmaster.amazon.com 1 7200 900 1209600 86400
<input type="checkbox"/>	www.example.com	CNAME	シンプル	-	ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com

5.2. Linux OS から名前解決の確認

Client VPC 内の Linux OS にログインの上、先ほど登録した FQDN で名前解決ができるか確認します。



- ① Linux OS に SSM 接続しているウェブブラウザのタブに戻り、名前解決をテストします。以下の例では、実行結果を完結表示にするため、“+short”オプションを加えています。

dig www.example.com. +short

```
[ssm-user@i-0a22092bc4b32884b ~]$ dig www.example.com. +short
ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com.
13.113.204.31
54.238.250.243
[ssm-user@i-0a22092bc4b32884b ~]$
```

- ② 現時点の ALB では、IPv6 専用とすることができず、必ず IPv4 と IPv6 の両方のアドレスが付与されます。先ほどの名前解決では、IPv4 が優先されたため、dig のオプションとして“aaaa”を指定し、意図的に IPv6 アドレスを対象とした名前解決を行います。

dig www.example.com. aaaa +short

```
[ssm-user@i-0a22092bc4b32884b ~]$ dig www.example.com. aaaa +short
ipv6-handson-alb-1213898242.ap-northeast-1.elb.amazonaws.com.
2406:da14:68a:d601:8397:a006:4f10:d9c
2406:da14:68a:d600:f15b:3084:c36:9fc8
[ssm-user@i-0a22092bc4b32884b ~]$
```

- ③ curl コマンドで先ほど Route 53 で作成したレコードに対し、IPv6 プロトコルにてアクセスします。Nginx のテストページのキーワードが表示されることを確認します。

curl -v6 http://www.example.com/ | head

```
[ssm-user@i-0a22092bc4b32884b ~]$ curl -v6 http://www.example.com/ | head
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
  0    0    0    0    0    0     0     0      0  0* Trying 2406:da14:68a:d600:f15b:3084:c36:9fc8:80...
* Connected to www.example.com (2406:da14:68a:d600:f15b:3084:c36:9fc8) port 80 (#0)
> GET / HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.79.1
> Accept: */*
<<省略>>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /*<![CDATA[*]
        body {          background-color: #fff;
[ssm-user@i-0a22092bc4b32884b ~]$
```

- ④ 同様にアクセス元 IP アドレスが表示されるよう、URL を変えてアクセスします。正しく接続元 Linux Host の IPv6 アドレスが表示されることを確認します。

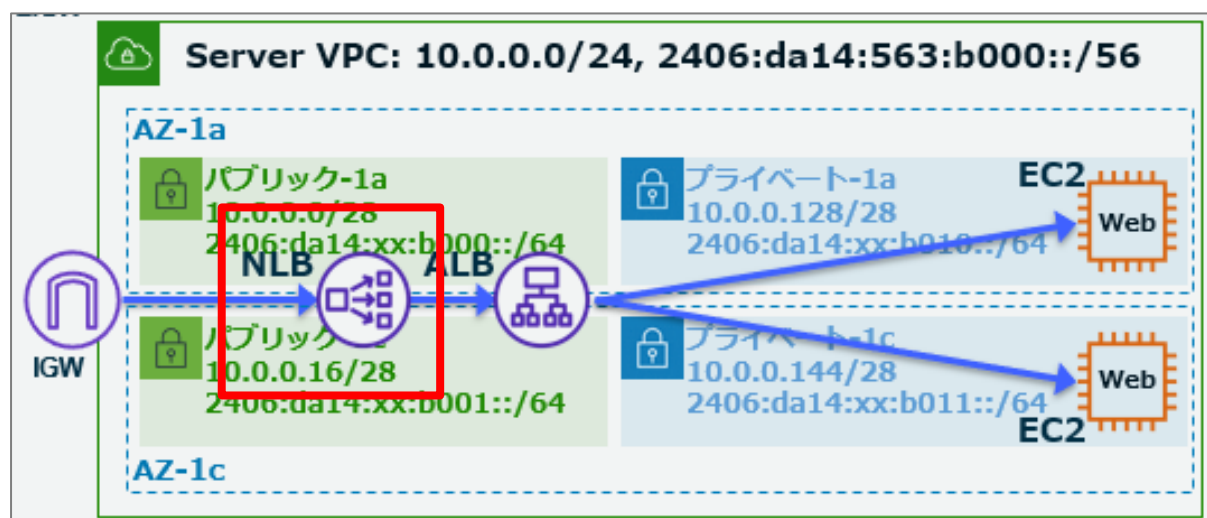
curl -v6 http://www.example.com/index2.php

```
[ssm-user@i-0a22092bc4b32884b ~]$ curl -v6 http://www.example.com/index2.php
* Trying 2406:da14:68a:d600:f15b:3084:c36:9fc8:80...
* Connected to www.example.com (2406:da14:68a:d600:f15b:3084:c36:9fc8) port 80 (#0)
> GET /index2.php HTTP/1.1
> Host: www.example.com
<<省略>>
<html><h1>HTTP_X_FORWARDED_FOR: 2406:da14:854:fc01:6e23:22ee:720f:81ac</html></h1>
* Connection #0 to host www.example.com left intact
[ssm-user@i-0a22092bc4b32884b ~]$
```

6.ALBの前にNLBを配置

IPv6 に特化した仕様ではありませんが、ALB ではスケーリング等の理由により、IP アドレスが変更になる場合があります。接続元環境のファイアーウォールなどにより、接続先 IP アドレスを固定化する要件に対応するためには、ALB の前に Network Load Balancer (NLB) を配置します。

この章では、すでに作成したロードバランサー環境に NLB を追加導入します。

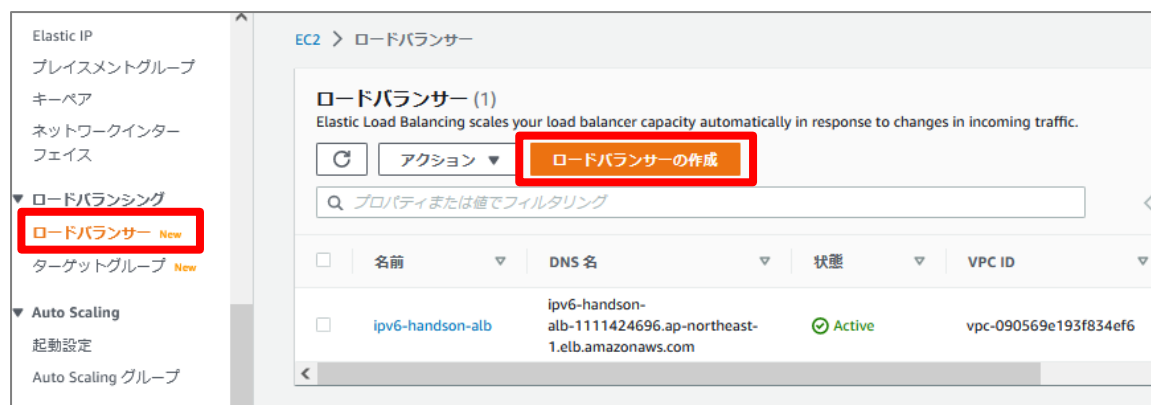


6.1. NLB を作成

- ① 画面左上の[サービス] > [コンピューティング]> “EC2”をクリックし、EC2 ダッシュボードに移動します。



- ② EC2 ダッシュボードのナビゲーションペインで、[ロードバランサー] > [ロードバランサーの作成]をクリックします。



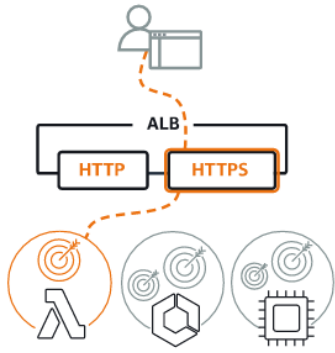
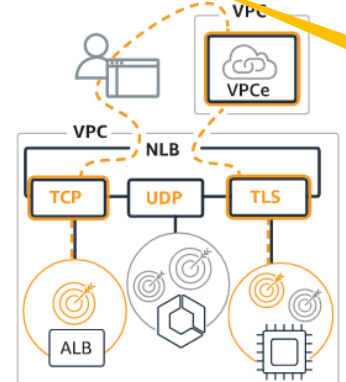

- ③ “Network Load Balancer”を示す図の下にある[作成]をクリックします。

EC2 > ロードバランサー > ロードバランサータイプの選択

ロードバランサータイプの選択

特徴ごとの完全な比較と詳細なハイライトも利用できます。 [詳細はこちら](#)

ロードバランサータイプ

Application Load Balancer 情報	Network Load Balancer 情報	Gateway Load Balancer 情報
 <p>HTTP および HTTPS トラフィックを使用するウェブアプリケーション用に柔軟性の高い機能セットが必要な場合は、Application Load Balancer を選択します。Application Load Balancer はリクエストレベルで動作し、マイクロサービスとコンテナを含む、アプリケーションアーキテクチャを対象とした高度なルーティングおよび可視性機能を提供します。</p> <p>作成</p>	 <p>非常に高いパフォーマンス、大規模な TLS のオフロード、証明書のデプロイの一元管理、UDP のサポート、およびアプリケーションの静的 IP アドレスが必要な場合は、Network Load Balancer を選択します。Network Load Balancer は接続レベルで動作し、非常に低いレイテンシーを維持しながら、1 秒あたり数百万のリクエストを瞬時に処理することができます。</p> <p>作成</p>	 <p>GENEVE をサポートするサードパーティの仮想アプライアンスのフリートをデプロイおよび管理する必要がある場合は、Gateway Load Balancer を選択します。これらのアプライアンスを使用すると、セキュリティポリシーを適用することができます。</p> <p>作成</p>

▶ Classic Load Balancer - 前の世代

- ④ “ロードバランサー名”に[**ipv6-handson-nlb**]と入力します。“IP アドレスタイプ”のラジオボタンで [**dualstack**]を選択します。

EC2 > ロードバランサー > Network Load Balancer の作成

Network Load Balancer の作成 情報

Network Load Balancer は、受信 TCP および UDP トラフィックを複数のターゲット（Amazon EC2 インスタンス、マイクロサービス、コンテナなど）に分散します。ロードバランサーが接続リクエストを受け取ると、リスナー設定で指定されたプロトコルとポート、およびデフォルトアクションとして指定されたルーティングルールに基づいてターゲットを選択します。

▶ Network Load Balancer の仕組み

基本的な設定

ロードバランサー名
名前は AWS アカウント内で一意である必要があり、ロードバランサーの作成後に変更することはできません。

ipv6-handson-nlb

ハイフンを含む最大 32 文字の英数字を使用できますが、名前の先頭または末尾にハイフンを使用することはできません。

スキーム
ロードバランサーの作成後にスキームを変更することはできません。

☒ インターネット向け
インターネット向けロードバランサーは、クライアントからのリクエストをインターネット経由でターゲットにルーティングします。これには、パブリックサブネットが必要です。 [詳細はこちら](#)

☐ 内部
内部ロードバランサーは、プライベート IP アドレスを使用して、クライアントからターゲットにリクエストをルーティングします。

IP アドレスタイプ 情報
サブネットで使用する IP アドレスのタイプを選択します。

☐ IPv4
社内向けロードバランサーに推奨。

☒ **Dualstack**
IPv4 と IPv6 アドレスが含まれます。

- ⑤ 画面下にスクロールし、“VPC”のプルダウンメニューから[**ipv6-handson-server**]の名前が設定された ID を選択します。“マッピング”では、表示された **ap-northeast-1a/1c** の 2 つのチ

エックボックスにチェックを入れ、“サブネット”から“public1”と“public2”の名前が付いたサブネットをそれぞれ選択します。その他の項目はデフォルトのままとします。

VPC

ターゲットの仮想プライベートクラウド (VPC) を選択します。インターネットゲートウェイを持つ VPC のみが選択可能になります。選択した VPC は、ロードバランサーの作成後に変更できません。ターゲットの VPC を確認するには、[ターゲットグループ](#) を表示します。

ipv6-handson-server-vpc

vpc-0ec991b3f0a4d96db

IPv4: 10.0.0.0/24 IPv6: 2406:da14:68a:d600::/56



マッピング

少なくとも 1 つのアベイラビリティゾーンと各ゾーンに 1 つのサブネットを選択します。少なくとも 2 つのアベイラビリティゾーンを選択することをお勧めします。ロードバランサーは、選択したアベイラビリティゾーンのターゲットにのみトラフィックをルーティングします。ロードバランサーまたは VPC でサポートされていないゾーンは選択できません。ロードバランサーが作成されると、サブネットは追加できますが、削除はできません。

☒ ap-northeast-1a

サブネット

subnet-0058163584257900b

ipv6-handson-server-subnet-public1-ap-northeast-1a

IPv4 の設定

IPv4 アドレス

AWS によって割り当て済み

IPv6 の設定

IPv6 アドレス

CIDR 2406:da14:68a:d600::/64 から割...

☒ ap-northeast-1c

サブネット

subnet-016f7c1349dc3c670

ipv6-handson-server-subnet-public2-ap-northeast-1c

IPv4 の設定

IPv4 アドレス

AWS によって割り当て済み

IPv6 の設定

IPv6 アドレス

CIDR 2406:da14:68a:d601::/64 から割...

- ⑥ 更にスクロールし、“リスナーとルーティング”で[ターゲットグループの作成]をクリックします。

リスナーとルーティング 情報
リスナーは、設定したポートとプロトコルを使用して接続リクエストをチェックするプロセスです。リスナー用に定義するルールによって、登録されたターゲットにロードバランサーがリクエストをルーティングする方法が決まります。

▼ リスナー TCP:80 削除

プロトコル ポート デフォルトアクション 情報
TCP : 80 転送先 ターゲットグループの選択 リフレッシュ
1~65535

ターゲットグループの作成

リスナータグ - 省略可能
リスナーにタグを追加することを検討してください。タグを使用すると、AWS リソースを分類できるため、リソースをより簡単に管理できます。

リスナータグの追加
タグは最大 50 個追加できます。|

リスナーの追加

- ⑦ お使いのブラウザで新たなタブとして“グループの詳細の指定”が表示されます。“ターゲットタイプの選択”では、[Application Load Balancer]を選択します。“ターゲットグループ名”には

[ipv6-handson-tg-alb]と入力します。“VPC”では[ipv6-handson-vpc]が選択されていることを確認します。

グループの詳細の指定

ロードバランサーは、ターゲットグループのターゲットにリクエストをルーティングし、ターゲットでヘルスチェックを実行します。

基本的な設定

このセクションの設定は、ターゲットグループの作成後に変更することはできません。

ターゲットタイプの選択

☐ インスタンス

- 特定の VPC 内のインスタンスへのロードバランシングをサポートします。
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

☐ IP アドレス

- VPC およびオンプレミスのリソースへのロードバランシングをサポートします。
- 同じインスタンス上の複数の IP アドレスとネットワークインターフェイスへのルーティングを容易にします。
- マイクロサービスベースのアーキテクチャに柔軟性を提供し、アプリケーション間の通信を簡素化します。
- IPv6 ターゲットをサポートし、エンドツーエンドの IPv6 通信と IPv4 から IPv6 への NAT を有効にします。

☐ Lambda 関数

- 単一の Lambda 関数へのルーティングを容易にします。
- Application Load Balancer にのみアクセスできます。

☒ Application Load Balancer

- Network Load Balancer に柔軟性を提供して、特定の VPC 内で TCP リクエストを受け入れてルーティングします。
- Application Load Balancer で静的 IP アドレスおよび PrivateLink を使用することを容易にします。

ターゲットグループ名

ipv6-handson-tg-alb

ハイフンを含む最大 32 文字の英数字を使用できますが、名前の先頭または末尾にハイフンを使用することはできません。

プロトコル

TCP

ポート

:

80

VPC

ターゲットグループに含める Application Load Balancer を含む VPC を選択します。

ipv6-handson-server-vpc

vpc-0ec991b3f0a4d96db

IPv4: 10.0.0.0/24

IPv6: 2406:da14:68a:d600::/56

- ⑧ 更にスクロールし、右下の[次へ]をクリックします。

ヘルスチェック

関連付けられたロードバランサーは、以下の設定ごとに、登録済みターゲットのステータスをテストするため、登録済みターゲットに対して定期的にリクエストを送信します。

ヘルスチェックプロトコル

HTTP ▼

ヘルスチェックパス

デフォルトパス「/」を使用してルートに ping を実行するか、必要に応じてカスタムパスを指定します。

/

最大文字数は 1024 です。

▶ ヘルスチェックの詳細設定

▶ タグ - 省略可能

ターゲットグループにタグを追加することを検討してください。タグを使用すると、AWS リソースを分類できるため、リソースをより簡単に管理できます。

キャンセル **次へ**

- ⑨ “ターゲットの登録”では” リストから Application Load Balancer を選択するか、Application Load Balancer を作成 します。”のプルダウンメニューから、“**ipv6-handson-alb**”を選択します。続いて、[ターゲットの作成]を選択します。


ターゲットの登録

これは、ターゲットグループを作成するためのオプションのステップです。ただし、ロードバランサーがこのターゲットグループにトラフィックをルーティングするようにするには、ターゲットを登録する必要があります。

Application Load Balancer

単一の Application Load Balancer をターゲットとして指定できます。指定する Application Load Balancer には、作成するターゲットグループと同じポートにリスナーが必要です。

☒ リストから Application Load Balancer を選択するか、Application Load Balancer を作成 [🔗](#) します。

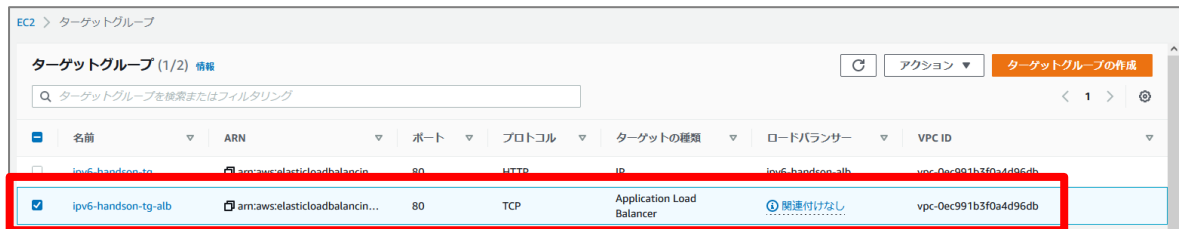
ipv6-handson-alb ▼ 

☐ 後で Application Load Balancer を追加

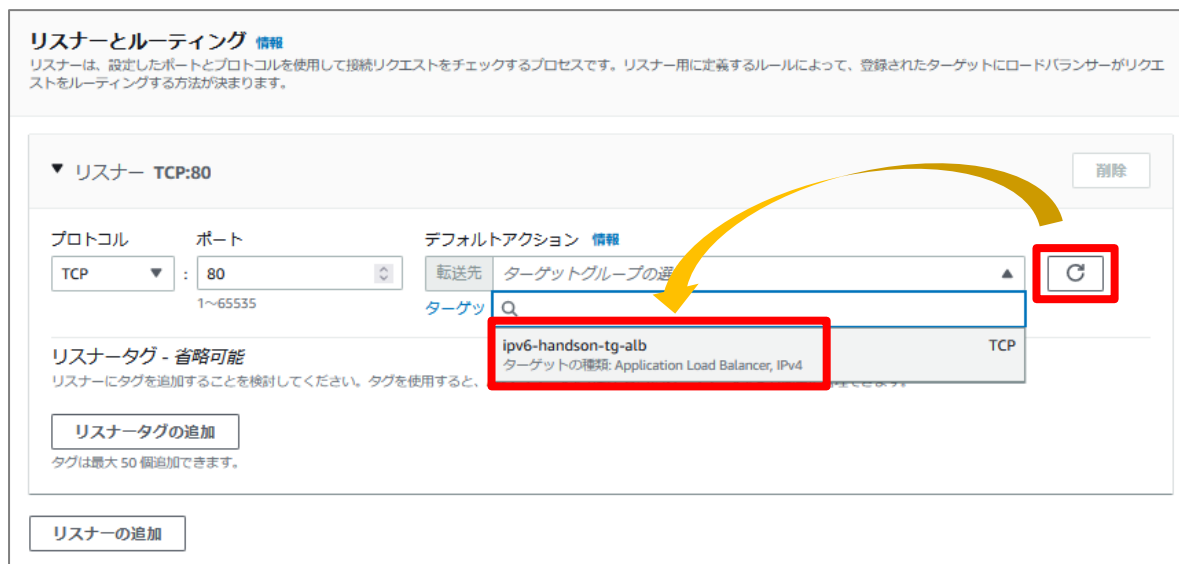
🟢 ipv6-handson-alb にはポート 80 にリスナーがあります。

キャンセル 戻る **ターゲットグループの作成**

- ⑩ ターゲットの種類が“Application Load Balancer”に指定された、新しいグループが作成されていることを確認します。現在作業しているウェブブラウザのタブから、先ほどロードバランサーを作成していたタブに戻ります。



- ⑪ “リスナーとルーティング”にてデフォルトアクションのターゲットグループを選択する箇所の右にある[更新ボタン]をクリックします。プルダウンメニューで先ほど作成した[ipv6-handson-tg-alb]が表示されますので、これを選択します。



- ⑫ 更にスクロールし、画面右下の[ロードバランサーの作成]をクリックします。

概要
設定を確認して確定します。 [コストの見積もり](#)

基本的な設定 編集 ipv6-handson-nlb • インターネット向け • Dualstack	ネットワークマッピング 編集 VPC vpc-0ec991b3f0a4d96db ipv6-handson-server-vpc • ap-northeast-1a subnet-0058163584257900b ipv6-handson-server-subnet-public1-ap-northeast-1a • ap-northeast-1c subnet-016f7c1349dc3c670 ipv6-handson-server-subnet-public2-ap-northeast-1c	リスナーとルーティング 編集 • TCP:80 次をデフォルトにする: ipv6-handson-tg-alb	タグ 編集 なし
--	---	--	------------------------------------

属性

① 特定のデフォルト属性がロードバランサーに適用されます。ロードバランサーの作成後に表示および編集できます。

キャンセル **ロードバランサーの作成**

- ⑬ ロードバランサーが正常に作成された旨を表示する画面へ推移します。右下の[ロードバランサーの表示]をクリックします。

✔ **ロードバランサーを正常に作成しました: ipv6-handson-nlb** ×
注意: ロードバランサーが完全にセットアップされてトラフィックのルーティング準備が整うまでに数分かかる場合があります。また、ターゲットの登録処理が完了して最初のヘルスチェックに合格するまでに数分かかる場合もあります。

EC2 > **ロードバランサー** > Network Load Balancer の作成

Network Load Balancer の作成

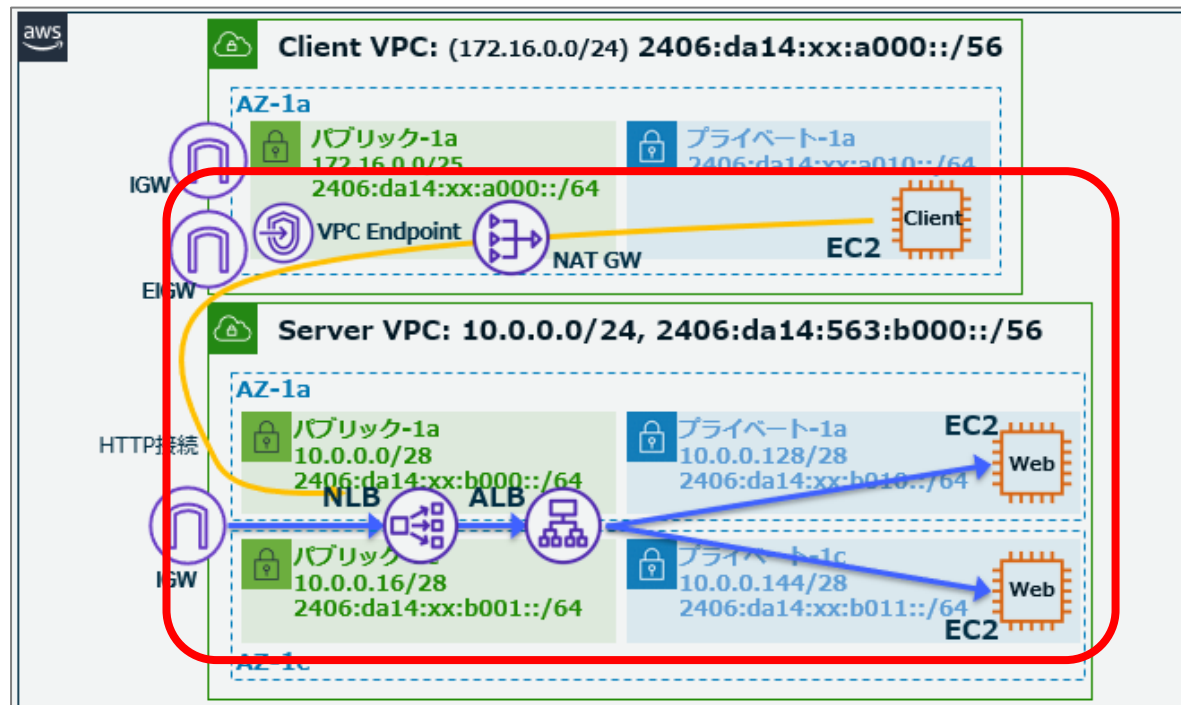
① **推奨される次のステップ**

- ipv6-handson-nlb 内の [説明] タブと [リスナー] タブを使用して、ロードバランサーおよびリスナーの属性を確認、カスタマイズ、有効化します。
- ロードバランサーと統合できる他のサービスを見つけましょう。ipv6-handson-nlb にある [統合サービス] タブにアクセスしてください。
- 静的 IP アドレスを使用すると、クライアントからファイアウォールまたはハードコード IP の許可リストに特定の IP を追加できます。このロードバランサーをプライベート接続用の AWS PrivateLink エンドポイントとして使用することを検討してください。 [AWS VPC マネジメントコンソール](#)

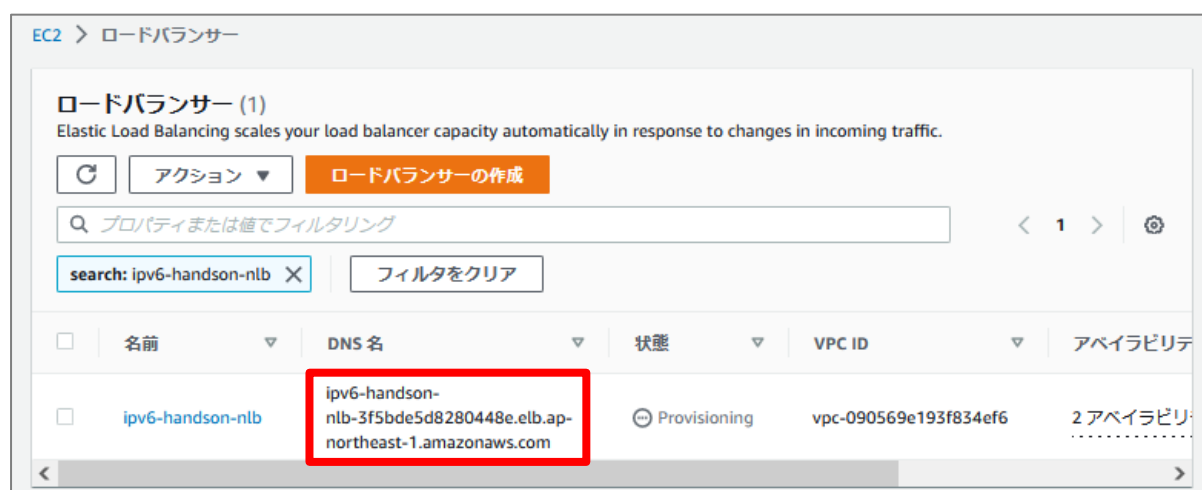
ロードバランサーを表示

6.2. NLB への接続確認

Client VPC の Linux OS から、作成した NLB を経由し、ターゲットの ALB からバックエンドの Web サーバへ接続できることを確認します。



- ① マネジメントコンソールの EC2 ダッシュボードで、先ほどの手順で作成したロードバランサー “ipv6-handson-nlb” を表示します。“状態”列が“プロビジョニング”から **Active** に推移するまで待ちます。しばらく待っても状態に変化が無い場合、右上のリロードボタンをクリックします。DNS 名列に記載の“DNS 名”をクリップボードにコピーします。



- ② Client VPC 内の Linux OS へ SSM 接続しているタブへ移動し、コピーした FQDN に対して curl コマンドで接続します。Nginx のテストページを示すキーワードが表示されていれば成功です。

curl -v6 http://ipv6-handson-nlb-1213898242.ap-northeast-1.elb.amazonaws.com/ | head

```
[ssm-user@i-0a22092bc4b32884b ~]$ curl -v6 http://ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com/ | head
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0   0   0    0    0    0     0      0  --:--:-- --:--:-- --:--:--    0*   Trying
2406:da14:68a:d600:a1fe:1ef:52c:ab75:80...
* Connected to ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com
(2406:da14:68a:d600:a1fe:1ef:52c:ab75) port 80 (#0)
> GET / HTTP/1.1
> Host: ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com
> User-Agent: curl/7.79.1
> Accept: */*
<<省略>>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test Page for the Nginx HTTP Server on Amazon Linux</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <style type="text/css">
      /*<![CDATA[*]
      body {
        background-color: #fff;
[ssm-user@i-0a22092bc4b32884b ~]$
```

- ③ NLB ではロードバランサーノードが利用する IP アドレスが固定されます。名前解決をしてこれらの IP を確認します。

dig ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com. +short

dig ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com. aaaa +short

```
[ssm-user@i-0a22092bc4b32884b ~]$ dig ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com. +short
52.199.196.246
18.179.107.244
[ssm-user@i-0a22092bc4b32884b ~]$ dig ipv6-handson-nlb-bf8b4403396ede68.elb.ap-northeast-1.amazonaws.com. aaaa +short
2406:da14:68a:d600:a1fe:1ef:52c:ab75
2406:da14:68a:d601:6076:bba9:b33d:3fa3
[ssm-user@i-0a22092bc4b32884b ~]$
```

以上、本ハンズオンの設定作業はすべて終了です。

7. ハンズオン環境のクリーニング

今回のハンズオンでは、当日限り利用可能な AWS アカウントを利用しているため、受講者がリソースを削除する必要はありません。

本ハンズオンテキストを利用し、受講者が所有している AWS アカウントを利用して環境を構築する場合、以下の項目を漏れなく削除いただくことで課金が停止します。

7.1. クリーニング対象リソース

- Route 53 リソースレコード
- Route 53 プライベートホストゾーン
- Application Load Balancer (ALB)
- Network Load Balancer (NLB)
- ALB で利用していたターゲットグループ
- NLB で利用していたターゲットグループ
- Web サーバ用 EC2 インスタンス（2 つのサブネットで起動していたリソース）
- 接続先 VPC（サブネットを含む）
- Client VPC（CloudFormation スタックの削除）
- CloudFormation テンプレート保管用 S3 バケット（"cf-templates-***-ap-northeast-1"の名称）

8. ドキュメント情報

8.1. 更新履歴

Ver #	Date	Author	Revision Description
1.0	2022.03.31	Nobuaki Kikuchi	初版
1.1	2022.10.07	Nobuaki Kikuchi	VPC マネージメントコンソールキャプチャ更新
1.2	2022.11.18	Nobuaki Kikuchi	ALB マネージメントコンソールキャプチャ更新 Server Yum 仕様変更のため、EOIGW を採用