



AWS IPv6 ハンズオン

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	2023.10.06	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. ハイブリッド 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. 外部サイトへの http 接続確認.....	17
4. 公開ウェブサーバー環境として接続先環境を作成	20
4.1. 接続先 VPC 環境	20
4.2. VPC 内のネットワーク環境作成.....	20
4.3. Web サーバの配置	25
4.4. Elastic Load Balancer (ALB)の設定	35
4.5. ALB への接続確認	47
5. DNS で IPv6 を FQDN に登録する	51
5.1. Route 53 でプライベートホストゾーンの登録	51
5.2. Linux OS から名前解決の確認	56
6. ALB の前に NLB を配置 (オプション項目)	59
6.1. NLB を作成	60
6.2. NLB への接続確認	69
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。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、シェル環境からコマンドを実行したり、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。

aws お問い合わせ サポート 日本語 アカウント **コンソールにサインイン**

製品 ソリューション 料金 ドキュメント 学ぶ パートナーネットワーク AWS Marketplace カスタマーサポート > Q

AWS マネジメントコンソール 概要 特徴 よくある質問

AWS マネジメントコンソール

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

もう一度ログインする

AWS の詳細を見る

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 >](#)



aws

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:

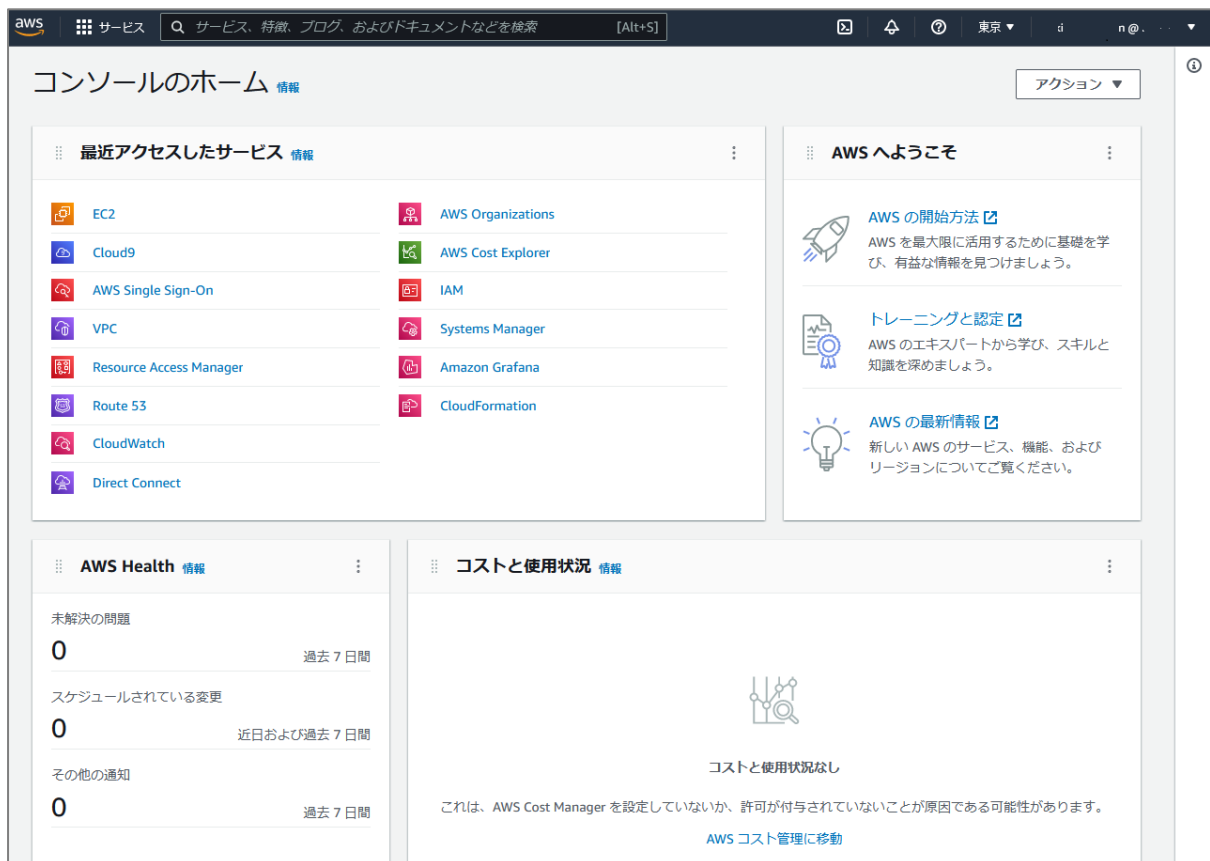
このアカウントを記憶する

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?

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.



aws サービス

サービス、特徴、ブログ、およびドキュメントなどを検索 [Alt+S]

東京 d n@

コンソールのホーム

アクション

最近アクセスしたサービス

- 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

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

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。

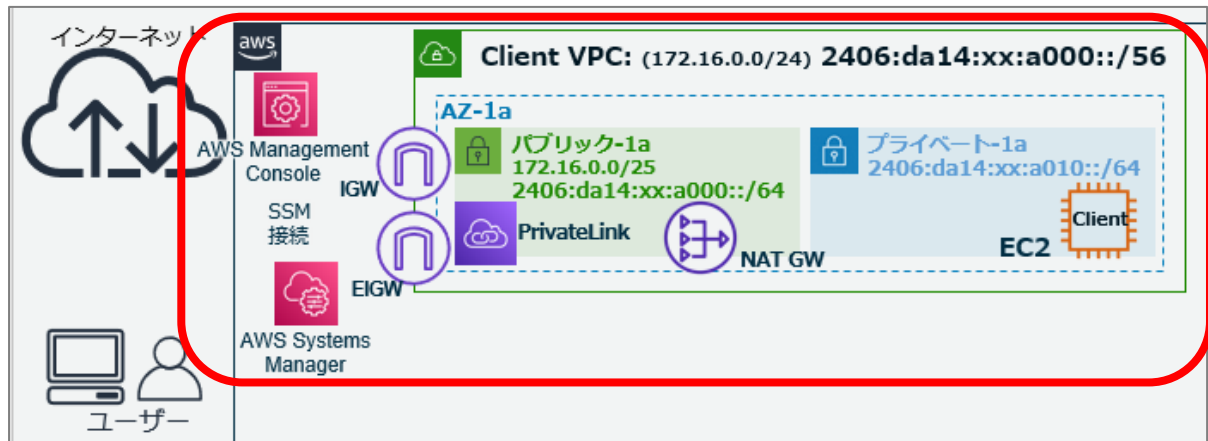
[AWS コスト管理に移動](#)

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 'Stack Creation' page in the AWS CloudFormation console. The page is divided into four steps: Step 1 (Template specification), Step 2 (Stack details), Step 3 (Stack options), and Step 4 (Review). The 'Template specification' section is active. It includes a 'Prerequisites - Template preparation' section with three radio buttons: 'Template preparation complete' (selected), 'Use sample template', and 'Create template with Designer'. Below that is the 'Template specification' section, which includes a 'Template source' section with two radio buttons: 'Amazon S3 URL' and 'Upload template file' (selected). A red box highlights the 'Upload template file' radio button. Below the 'Upload template file' radio button is a 'File selection' button, also highlighted with a red box. The text next to the button says 'File selection' and 'File not selected'. At the bottom right, there are 'Cancel' and '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-1li6cdhhhcjv-ap-northeast-1/202208787W-IPv6-HandsOn-Client.yaml`

キャンセル

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

- 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](#)

キー	値	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](#)

- 全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**

- 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。

変更セットの作成 キャンセル 戻る 送信

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

The screenshot shows the AWS CloudFormation console for the stack 'ipv6-handson'. The stack is in the 'CREATE_IN_PROGRESS' state, highlighted with a red box. The 'イベント' (Events) tab is selected, showing one event with the status 'CREATE_IN_PROGRESS' and the reason 'User Initiated'.

タイムスタンプ	論理 ID	ステータス	状況の理由
2021-10-25 18:05:55 UTC+0900	ipv6-handson	CREATE_IN_PROGRESS	User Initiated

↓

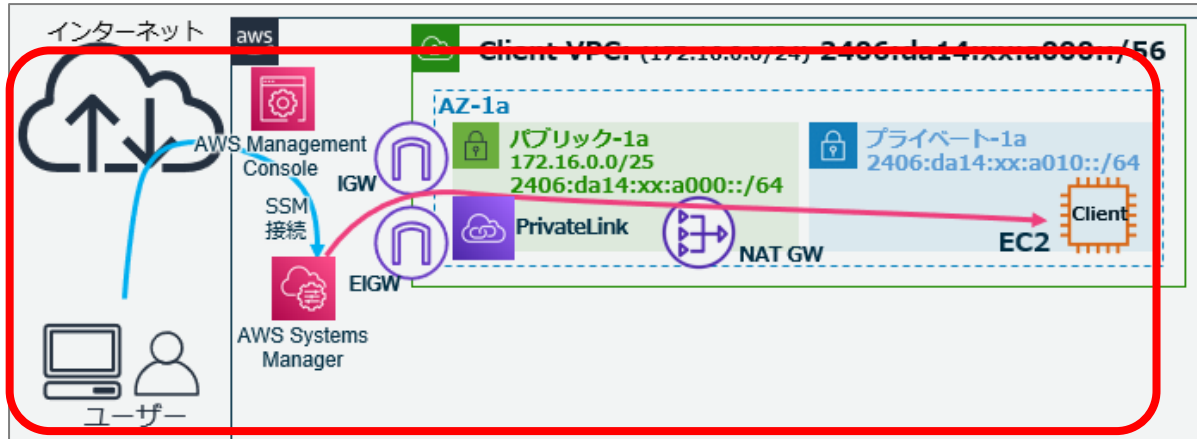
The screenshot shows the AWS CloudFormation console for the stack 'ipv6-handson'. The stack is now in the 'CREATE_COMPLETE' state, highlighted with a red box. The 'イベント' (Events) tab is selected, showing 44 events. The first two events are in the 'CREATE_COMPLETE' state, and the third event is in the 'CREATE_IN_PROGRESS' state with the reason 'Resource creation Initiated'.

タイムスタンプ	論理 ID	ステータス	状況の理由
2021-10-25 18:09:10 UTC+0900	ipv6-handson	CREATE_COMPLETE	-
2021-10-25 18:09:08 UTC+0900	BastionLinuxS erver	CREATE_COMPLETE	-
2021-10-25 18:08:35 UTC+0900	BastionLinuxS erver	CREATE_IN_PROGRESS	Resource creation Initiated

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。

CloudFormation > スタック > ipv6-handson

スタック (2)

ipv6-handson
2022-03-28 15:21:21 UTC+0900
CREATE_COMPLETE

ipv6-handson

スタックの情報 | イベント | **リソース** | 出力 | パラメータ | テンプレート

リソース (22)

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 EC2 instance. The instance name 'ipv6-handson-BastionLinuxServer' is highlighted with a red box. Below the instance list, the 'インスタンス概要' (Instance Summary) section is visible, with the 'インスタンスの状態' (Instance State) field highlighted in red, showing '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
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	パブリック IPv4 アドレス
i-0a7b5bce6e49709e6 (ipv6-handson-BastionLinuxServer)	-
IPv6 アドレス	インスタンスの状態
2406:da14:3d:3901:1d8c:5a76:de77:487b	実行中
ホスト名のタイプ	プライベートリソース DNS 名
リソース名: i-0a7b5bce6e49709e6.ap-northeast-1.compute.internal	i-0a7b5bce6e49709e6.ap-northeast-1.compute.internal

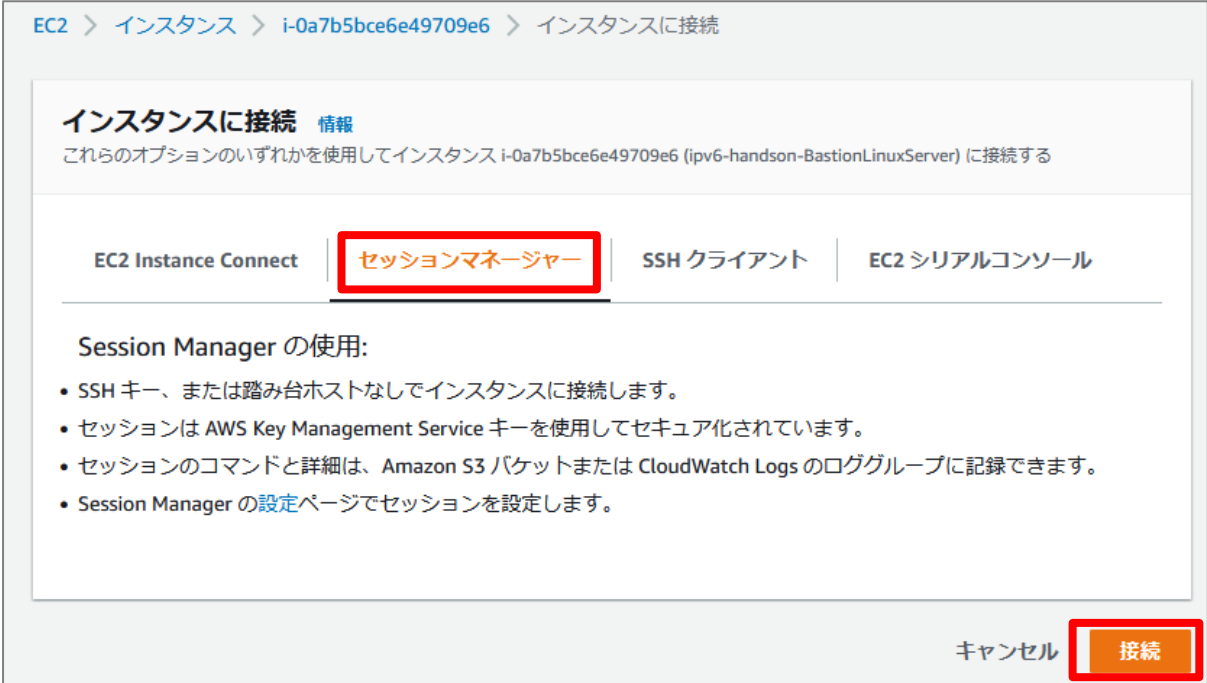
③ インスタンスの状態が**実行中**となっていることを確認したのち、画面上の**[接続]**を選択します。

The screenshot shows the AWS Management Console interface for the same EC2 instance. The '接続' (Connect) button in the top right corner is highlighted with a red box. The instance status is still '実行中' (Running).

接続

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
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。



EC2 > インスタンス > i-0a7b5bce6e49709e6 > インスタンスに接続

インスタンスに接続 情報

これらのオプションのいずれかを使用してインスタンス i-0a7b5bce6e49709e6 (ipv6-handson-BastionLinuxServer) に接続する

EC2 Instance Connect | **セッションマネージャー** | SSH クライアント | EC2 シリアルコンソール

Session Manager の使用:

- SSH キー、または踏み台ホストなしでインスタンスに接続します。
- セッションは AWS Key Management Service キーを使用してセキュア化されています。
- セッションのコマンドと詳細は、Amazon S3 バケットまたは CloudWatch Logs のロググループに記録できます。
- Session Manager の [設定ページ](#) でセッションを設定します。

キャンセル **接続**

- 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”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。

```
セッション ID: MasterKey-0199f86493c68d98f   インスタンス ID: i-03e8473e0bb7da006
```

```
sh-4.2$ bash
[ssm-user@i-03e8473e0bb7da006 bin]$ cd
[ssm-user@i-03e8473e0bb7da006 ~]$
```

- 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. 外部サイトへの http 接続確認

CloudFormation テンプレートを用いて作成した環境では、すでに curl コマンドを利用して、インターネット上の Web サイトへ http 接続可能な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 コマンド : "dig ifconfig.io aaaa +short"を実行し、名前解決により IPv6 アドレスが入手できている事を確認します。(実際の IP アドレスは異なる可能性があります)

dig ifconfig.io aaaa +short

```
[ssm-user@i-0927c232a12443f15 ~]$ dig ifconfig.io aaaa +short
2606:4700:e6::ac40:c310
2606:4700:e6::ac40:c210
[ssm-user@i-0927c232a12443f15 ~]$
```

- ② 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。）

また、最下部に接続元となっているクライアント（EC2 インスタンス）の IPv6 アドレスが表示されていることを確認します。

curl -v6 http://ifconfig.io/

```
[ssm-user@i-0927c232a12443f15 ~]$ curl -v6 http://ifconfig.io/
* Trying 2606:4700:e6::ac40:c210:80...
* Connected to ifconfig.io (2606:4700:e6::ac40:c210) port 80 (#0)
> GET / HTTP/1.1
> Host: ifconfig.io
> User-Agent: curl/7.79.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Thu, 02 Feb 2023 14:08:04 GMT
< 中略 >
<
2406:da14:342:2901:b8cf:a60d:ba4b:93af
* Connection #0 to host ifconfig.io left intact
```

必要に応じて、再度、EC2 インスタンスの IPv6 アドレスを確認します。

ip address show dev eth0

```
[ssm-user@i-0927c232a12443f15 ~]$ ip address show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:39:a3:af:05:41 brd ff:ff:ff:ff:ff:ff
    inet 169.254.24.173/32 scope global dynamic eth0
        valid_lft 2388sec preferred_lft 2388sec
    inet6 2406:da14:342:2901:b8cf:a60d:ba4b:93af/128 scope global dynamic
        valid_lft 382sec preferred_lft 72sec
    inet6 fe80::439:a3ff:feaf:541/64 scope link
        valid_lft forever preferred_lft forever
[ssm-user@i-0927c232a12443f15 ~]$
```

- ③ 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>>
* Connection state changed (MAX_CONCURRENT_STREAMS == 128)!
< HTTP/2 200
< date: Thu, 02 Feb 2023 14:13:25 GMT
<<中略>>
<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
```

- ④ 参考：その他の IPv6 接続を確認できるサイト

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

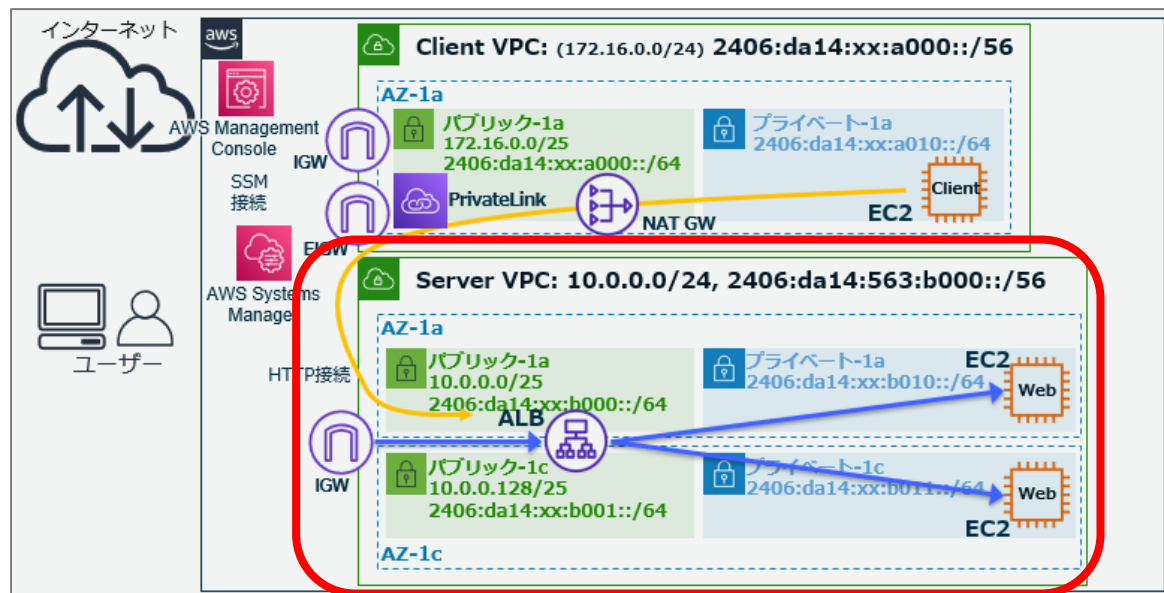
<https://www.kddi.com/>

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。

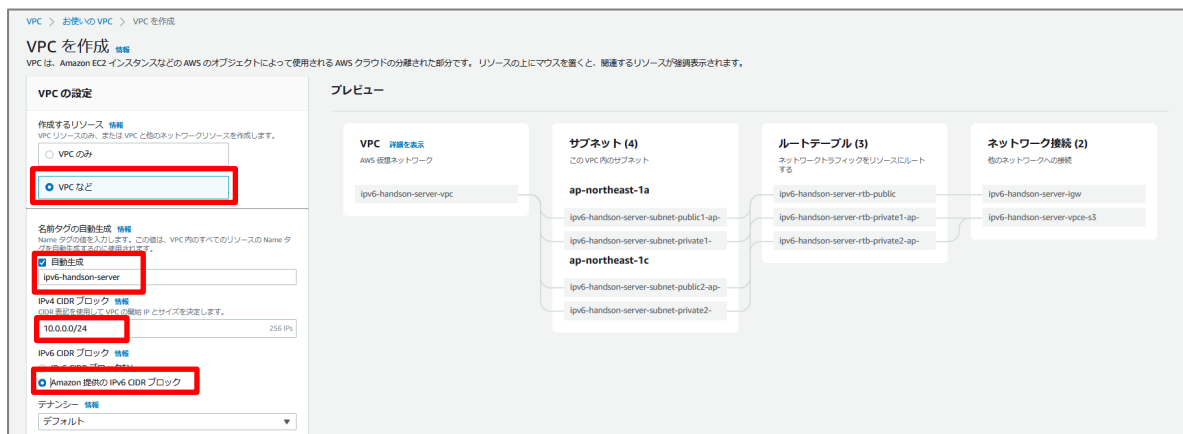
- ① 画面右上のリージョンセレクターにおいて、東京リージョンが選択されていることを確認します。
[サービス] > [ネットワーキングとコンテンツ配信] > [VPC]を選択し、VPC ダッシュボードを表示します。（検索機能から“VPC”を検索する方法でも可能）



- 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	
アベイラビリティゾーン (AZ)	2	
AZ のカスタマイズ	第 1 アベイラビリティゾーン: ap-northeast-1a 第 2 アベイラビリティゾーン: 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。

VPC ワークフローの作成

成功

▼ 詳細

- 成功 VPC を作成: [vpc-0565bed27b9ca119a](#)
- 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-0565bed27b9ca119a](#)
- 成功 S3 エンドポイントを作成: [vpce-0802cde735eb1486b](#)
- 成功 サブネットを作成: [subnet-0d421d2587f62a301](#)
- 成功 サブネットを作成: [subnet-04ece1df7a23fa789](#)
- 成功 サブネットを作成: [subnet-043ccb447d62af996](#)
- 成功 サブネットを作成: [subnet-0bb1fa364a71e153c](#)
- 成功 インターネットゲートウェイの作成: [igw-047abfa6fd48afb54](#)
- 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-0d118beb8fc535e9d](#)
- 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 Egress Only インターネットゲートウェイを作成: [eigw-0c58eb284dcccdbbd](#)
- 成功 ルートテーブルを作成: [rtb-058822a4bcab4ced3](#)
- 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-05c0ee91b637d1bd9](#)
- 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-0802cde735eb1486b](#)

VPC を表示

VPC > お使いの VPC > vpc-0565bed27b9ca119a

vpc-0565bed27b9ca119a / ipv6-handson-server-vpc アクション ▼

詳細 情報

VPC ID vpc-0565bed27b9ca119a	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-04bac7450674aa46a	メインルートテーブル rtb-0c70f0f0835acfe28	メインネットワーク ACL acl-0afc50c56af962ccb
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クラウドグループ) 2406:da14:aff:3d00::/56 (ap-northeast-1) Associated
ネットワークアドレスの使用状況メトリクス 無効	Route 53 リゾルバー DNS ファイアウォールルールグループ -	所有者 ID 115657431263	

リソースマップ 新規 CIDR フローログ | タグ

CIDR 情報

アドレスタイプ	CIDR	ネットワークボーダークラウドグループ	プール	ステータス
IPv4	10.0.0.0/24	-	-	Associated
IPv6	2406:da14:aff:3d00::/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
VPC でフィルタリング:
VPC の選択

サブネット (4) 情報

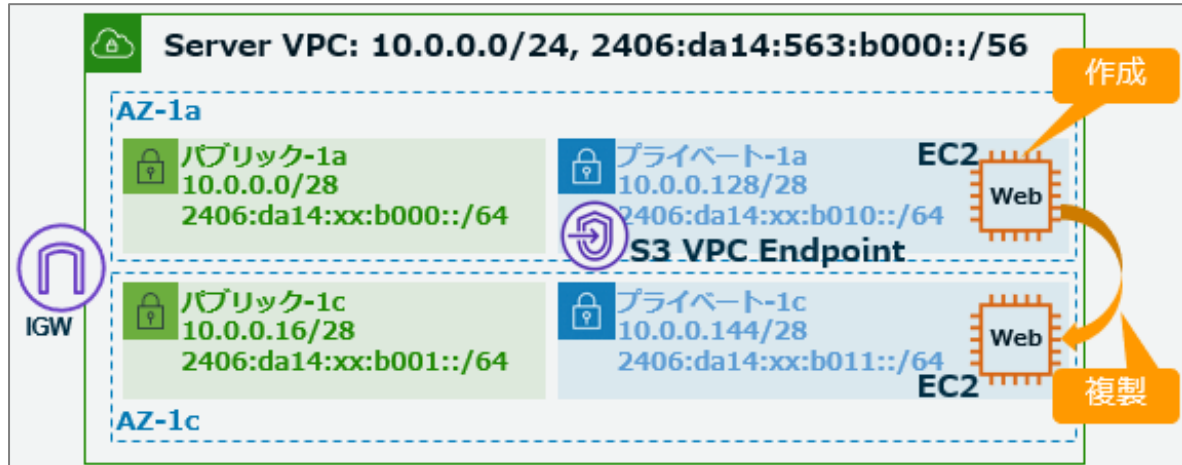
サブネットをフィルター
search: ipv6-handson-server X | フィルターをクリア

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

サブネット

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, the navigation sidebar has 'EC2 ダッシュボード' highlighted with a red box. The main content area is titled 'リソース' (Resources) and shows a table of resource counts for the 'アジアパシフィック (東京)' region. Below this, there are three sections: 'インスタンスを起動' (Start Instance), 'サービス状態' (Service Status), and 'サーバーを移行' (Migrate Server). The 'インスタンスを起動' section has a red box around the 'インスタンスを起動' button.

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

- 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' (Start Instance) page in the AWS Management Console. The breadcrumb navigation is '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' (Info) link. Below the heading, there is a paragraph of introductory text. The '名前とタグ' (Name and Tags) section is highlighted with a red box. The '名前' (Name) field is filled with 'ipv6-handson-server-1a' and is also highlighted with a red box. There is a 'さらにタグを追加' (Add more tags) link next to the 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 2 AMI”を選択します。インスタンスタイプは“t2.micro”から変更しません。

▼ 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。

Q 何千ものアプリケーションイメージと OS イメージを含むカタログ全体を検索します。

最新 | **クイックスタート**



macOS


Ubuntu


Windows


Red Hat


SUSE Linux


🔍 その他の AMI を開
 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 無料利用枠の対象 ▼

ami-0adac58024a7f03bb (64 ビット (x86)) / ami-0ebf9e4700190286e (64 ビット (Arm))

仮想化: hvm ENA 有効: true ルートデバイスタイプ: ebs

説明

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230926.0 x86_64 HVM gp2

アーキテクチャ AMI ID

64 ビット (x86) ▼

ami-0adac58024a7f03bb 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

t2.micro 無料利用枠の対象

ファミリー: t2 1 vCPU 1 GiB メモリ 現行世代: true

オンデマンド Windows 基地 料金: 0.0198 USD 1 時間あたり

オンデマンド SUSE 基地 料金: 0.0152 USD 1 時間あたり

オンデマンド RHEL 基地 料金: 0.0752 USD 1 時間あたり

オンデマンド Linux 基地 料金: 0.0152 USD 1 時間あたり

ソフトウェアがプリインストールされた AMI には追加料金がかかります

すべての世代

[インスタンスタイプを比較](#)



28

- 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 編集

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

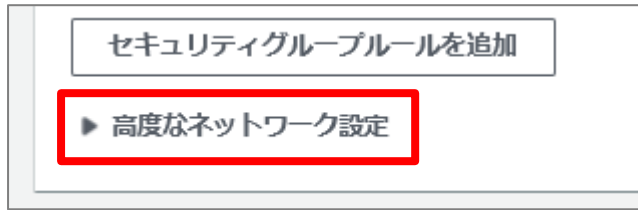
タイプ 情報: HTTP
プロトコル 情報: TCP
ポート範囲 情報: 80
Source type 情報: Custom
Source 情報: ::/0
説明 - optional 情報: 例 管理者のデスクトップの SSH

セキュリティグループルールを追加

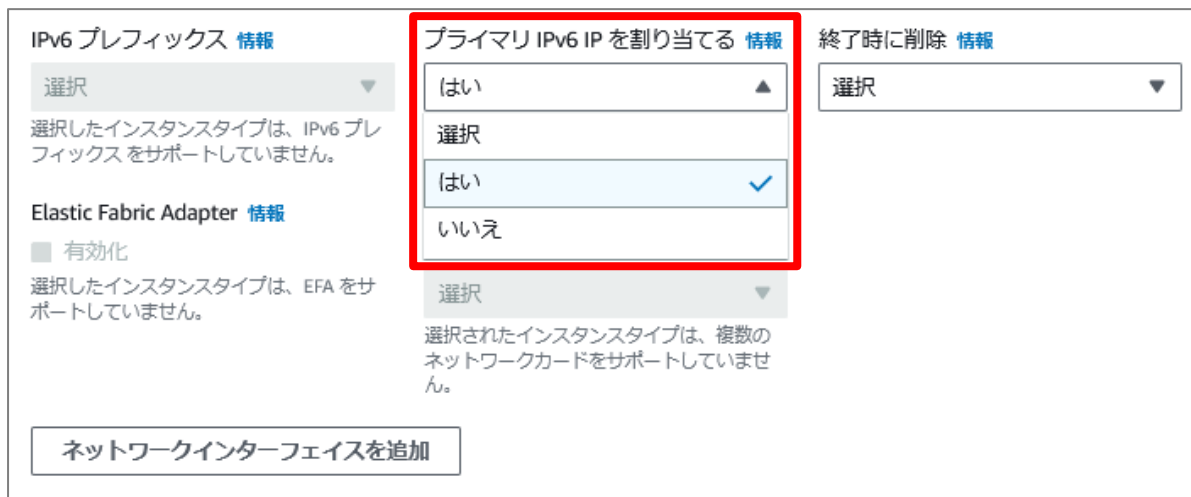
▶ 高度なネットワーク設定

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

- ⑧ “高度なネットワーク設定”の左側にある”▶”をクリックし展開します。



- ⑨ “プライマリ IPv6 IP を割り当てる”のプルダウンメニューを開き[はい]を選択します。



- ⑩ “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。（「高度なネットワーク設定」ではないので、ご注意ください）



<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

<p>ユーザーデータ</p> <p>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。</p>	<pre>#!/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</pre>
---	--

- ⑪ 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

- ⑫ 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。



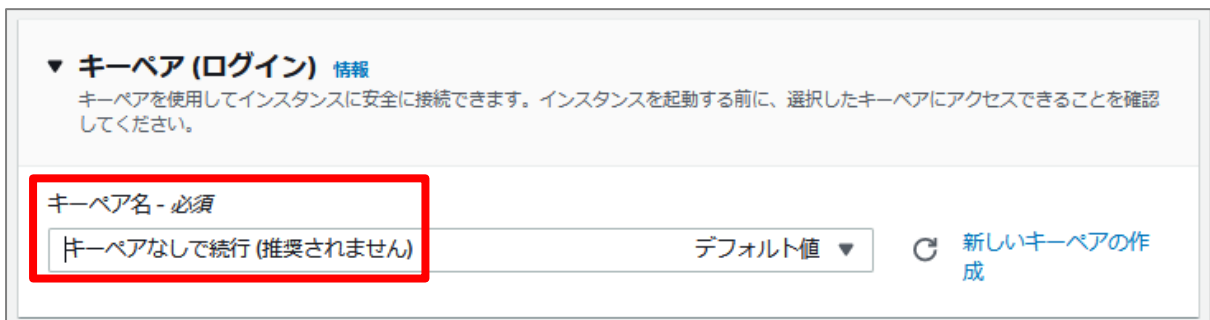
- ⑬ インスタンスが実行中のステータスになったら、他方のサブネットにも Web サーバを起動するために複製します。起動した EC2 インスタンス ID の左にあるチェックボックスにチェックを入れ、"アクション"メニュー->"イメージとテンプレート">"同様のものを起動"を選択します。



- ⑭ "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。



- ⑮ "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。



- ⑩ 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-0bbe7b5f4e1fccfe
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

- ⑪ “プライマリ IPv6 IP を割り当てる”のプルダウンメニューを開き[はい]を選択します。

IPv6 プレフィックス 情報

選択

選択したインスタンスタイプは、IPv6 プレフィックスをサポートしていません。

Elastic Fabric Adapter 情報

■ 有効化

選択したインスタンスタイプは、EFA をサポートしていません。

プライマリ IPv6 IP を割り当てる 情報

はい

選択

はい ✓

いいえ

選択

選択されたインスタンスタイプは、複数のネットワークカードをサポートしていません。

終了時に削除 情報

選択

ネットワークインターフェイスを追加

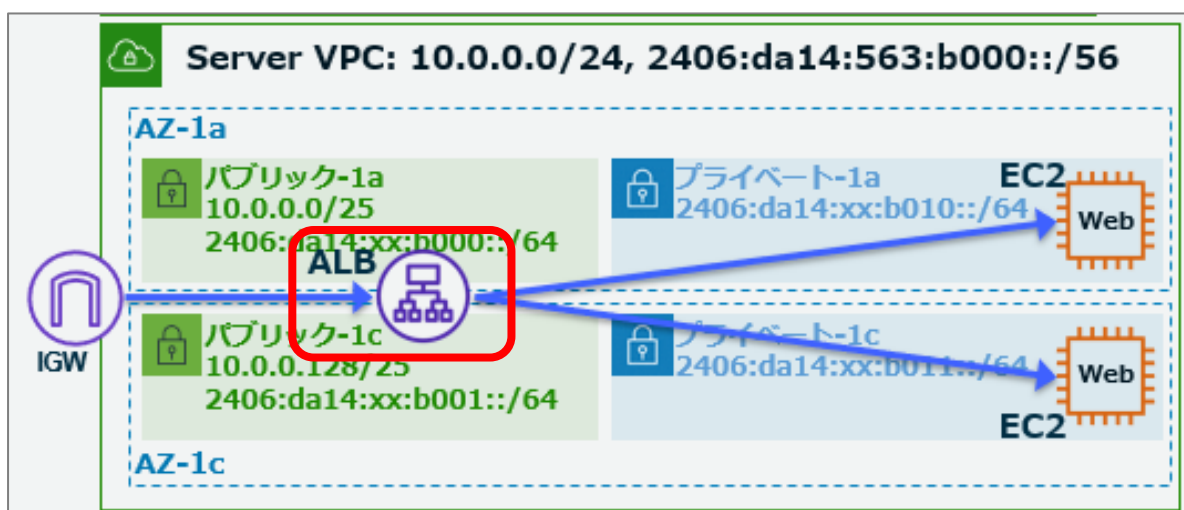
- ⑱ その他は既存のまま最下部の“概要”までスクロールし、“**インスタンスを起動**”を選択します。



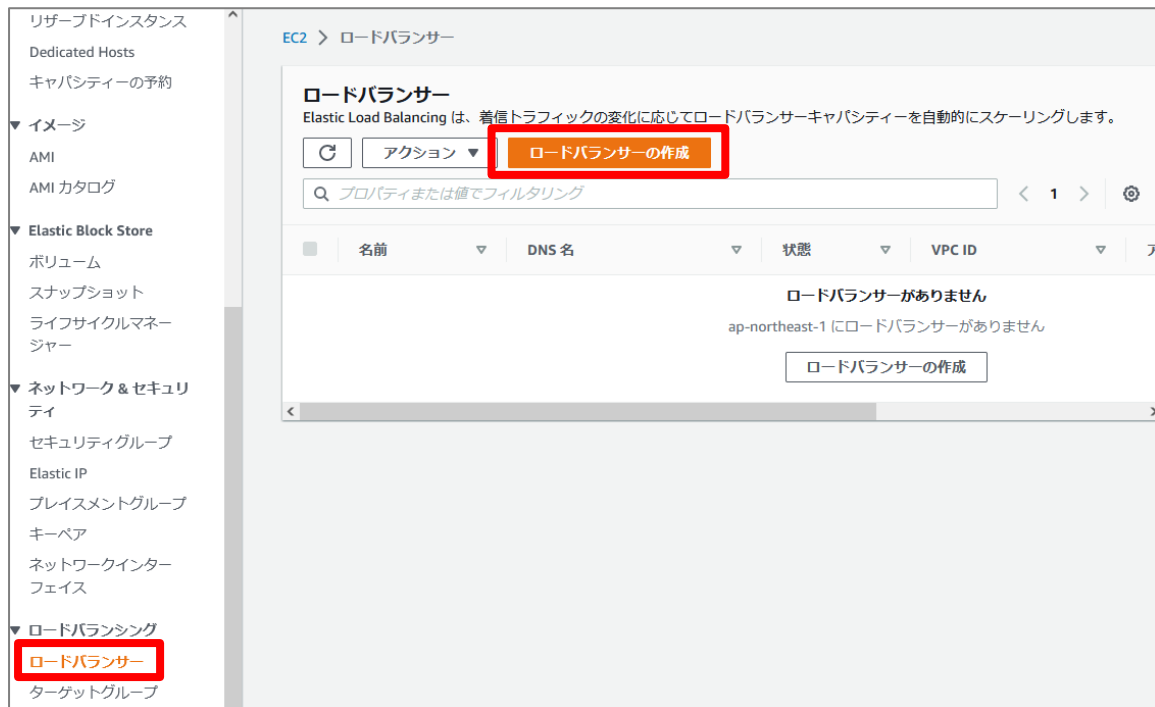
ここまでの作業で、二つの Private サブネットにそれぞれ EC2 インスタンスが起動し、nginx による Web サイトが起動しています。

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、すべてのリクエストを確実に処理することができます。

作成

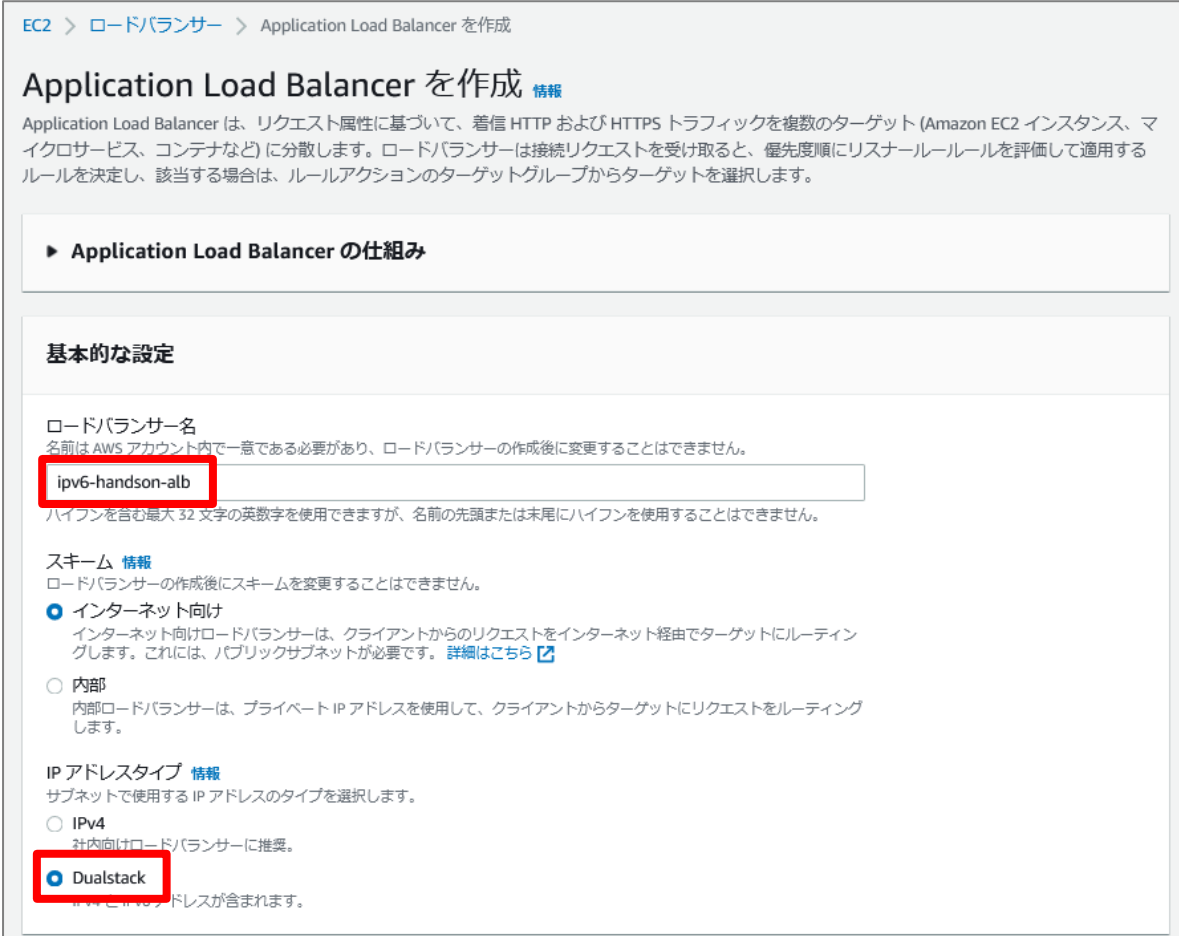
Gateway Load Balancer 情報

GENEVE をサポートするサードパーティー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。

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-vpc**]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

つのチェ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

ネットワークマッピング 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) を選択するか、[新しい VPC を作成](#) できます。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-0565bed27b9ca119a
IPv4: 10.0.0.0/24 IPv6: 2406:da14:aff:3d00::/56

マッピング 情報
少なくとも 2 つのアベイラビリティゾーンと、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アベイラビリティゾーンは選択できません。

ap-northeast-1a (apne1-az4)

サブネット
subnet-0d421d2587f62a301 ipv6-handson-server-subnet-public1-ap-northeast-1a

IPv4 アドレス
AWS によって割り当て済み

ap-northeast-1c (apne1-az1)

サブネット
subnet-04ecce1df7a23fa789 ipv6-handson-server-subnet-public2-ap-northeast-1c

IPv4 アドレス
AWS によって割り当て済み

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。既存のセキュリティグループを選択するか、[新しいセキュリティグループを作成](#) できます。

セキュリティグループ 新たに作成

最大 5 個のセキュリティグループを選択

default
sg-0df1e25cbfb8ed385 VPC: vpc-0565bed27b9ca119a

- 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

作成後に名前を編集することはできません。

説明 情報

VPC 情報

vpc-0b92530d8d0790cce 172.31.0.0/16	(デフォルト)	
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		ールがありません。

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

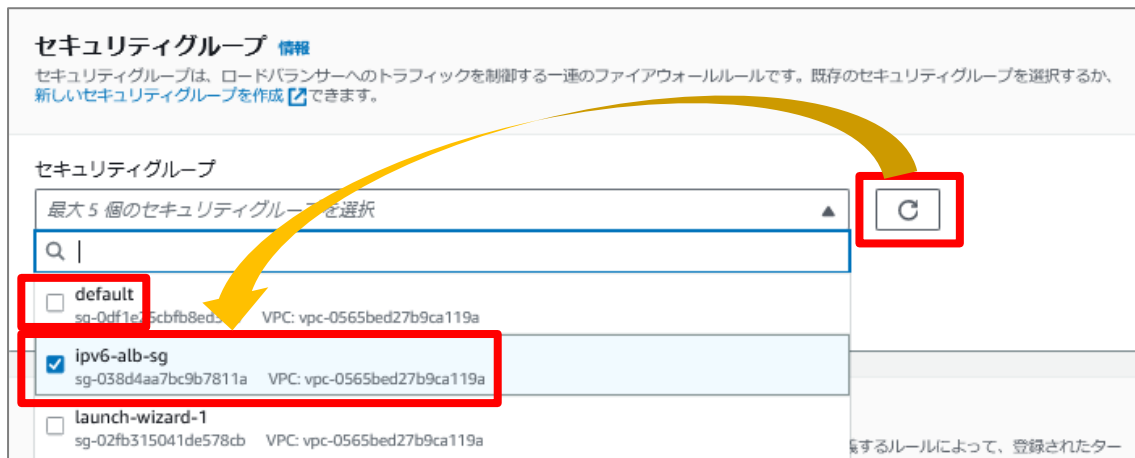
インバウンドルール 2 削除

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

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]グループのチェックボックス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)



- 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。



- 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、[インスタンス]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。

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
1~65535

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

Instances that you register must have an assigned primary IPv6 address. This is configured on the instances default network interface (eth0). [Learn more](#)

VPC

ターゲットグループに含めるインスタンスを含む 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。

ipv6-handson-server-vpc

vpc-05655bed27b9ca119a
IPv4: 10.0.0.0/24 IPv6: 2406:da14:aff:3d00::/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**

属性

④ 特定のデフォルト属性がターゲットグループに適用されます。ターゲットグループの作成後に表示および編集できます。

▶ **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“使用可能なインスタンス”の項目で、“インスタンス ID”の左にあるラジオボタン選択し、候補のインスタンスすべてにチェックを入れます。続いて下に表示された、**[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には、ターゲットを登録する必要があります。

使用可能なインスタンス (2/2)

Only instances that have an assigned primary IPv6 address are available for selection. Instances without an assigned IPv6 address are indicated as **Unassigned** in the Primary IPv6 address column. Click the **Unassigned** status for instructions on assigning a primary IPv6 address.

🔍 プロパティまたは値でリソースをフィルタリング

Hide instances with unassigned primary IPv6 address

<input checked="" type="checkbox"/>	インスタンス ID	名前	状態	セキュリティグループ	ゾーン
<input checked="" type="checkbox"/>	i-042be31fcfecbc61d	ipv6-handson-server-1a	🟢 実行中	ipv6-alb-sg	ap-northeast-1a
<input checked="" type="checkbox"/>	i-00405d83ecbb7b19f	ipv6-handson-server-1c	🟢 実行中	ipv6-alb-sg	ap-northeast-1c

2 個を選択済み

選択したインスタンスのポート
 選択したインスタンスにトラフィックをルーティングするためのポート。

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

ターゲット (2) 保留中のものをすべて削除

🔍 プロパティまたは値でリソースをフィルタリング Show only pending

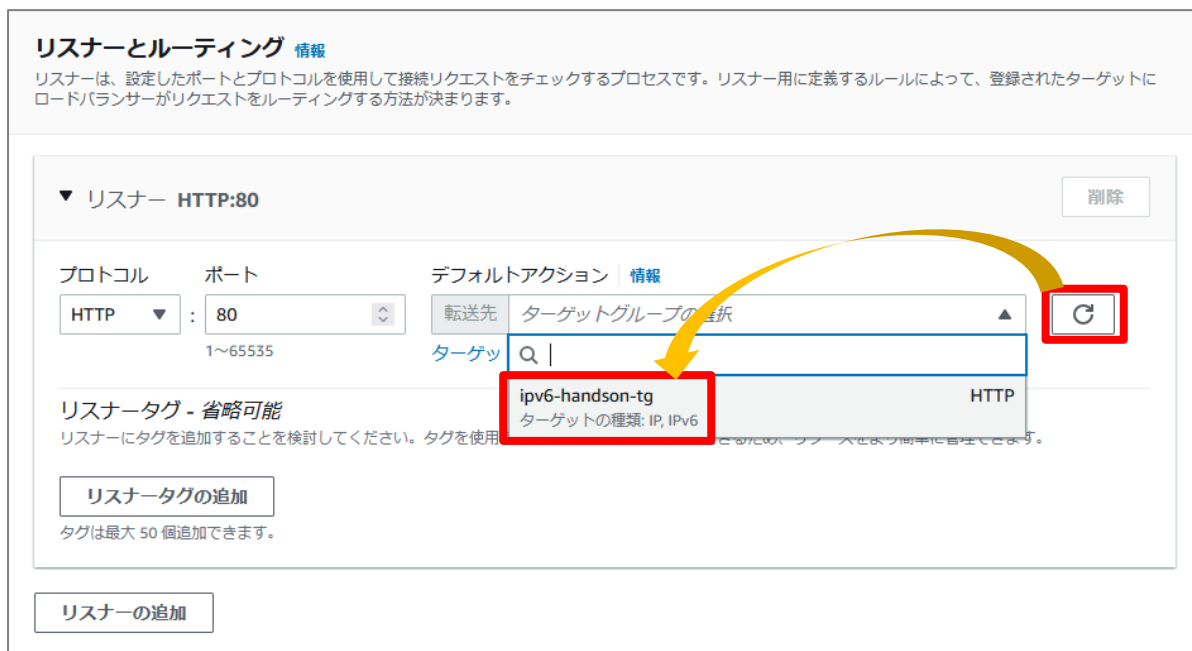
削除	ヘルスステータス	インスタンス ID	名前	ポート	状態	セキュリティグループ	ゾーン	Primary IPv6 address	サブネット ID
<input type="checkbox"/>	保留中	i-042be31fcfecbc61d	ipv6-handson-server-1a	80	🟢 実行中	ipv6-alb-sg	ap-northeast-1a	2406:da14:aff:3d02:a43f:c528:b54:5b0a	subnet-043ccb447d62af996
<input type="checkbox"/>	保留中	i-00405d83ecbb7b19f	ipv6-handson-server-1c	80	🟢 実行中	ipv6-alb-sg	ap-northeast-1c	2406:da14:aff:3d03:252d:7e:95ad:49c1	subnet-0bb1fa364a71e153c

2 個を保留中 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。



- ⑯ “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。



- 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](#)

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

属性

i 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。

o **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もあります。

EC2 > ロードバランサー

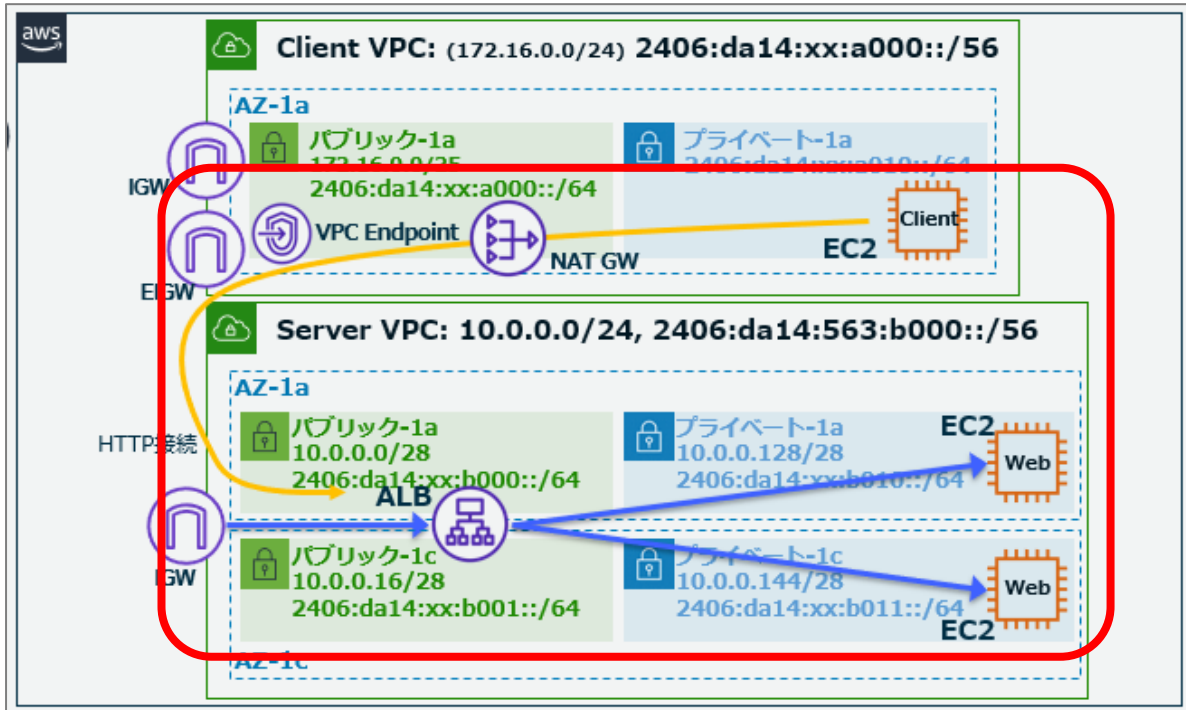
i **推奨される次のステップ**

- 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。

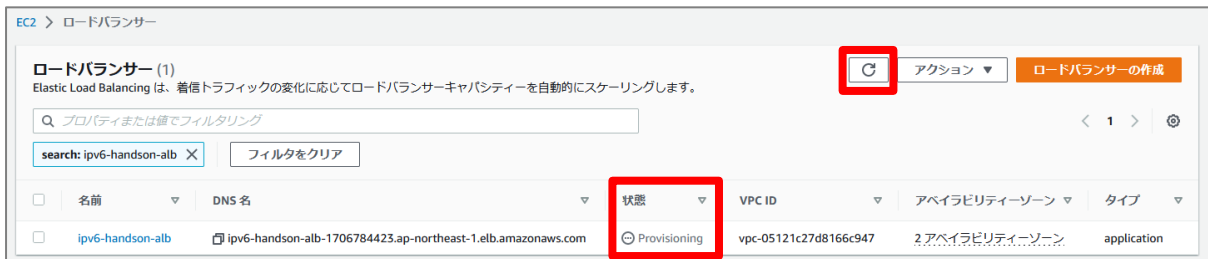
View load balancer

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。先頭にあるコピーボタンを利用してください。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。



- 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-1706784423.ap-northeast-1.elb.amazonaws.com/ | head
```

```
[ssm-user@i-0927c232a12443f15 ~]$ curl -v6 http://ipv6-handson-alb-1706784423.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  0*   Trying
2406:da14:9d7:ad00:693a:ffba:da1b:aca7:80...
* Connected to ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com
(2406:da14:9d7:ad00:693a:ffba:da1b:aca7) port 80 (#0)
> GET / HTTP/1.1
> Host: ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com
> User-Agent: curl/7.79.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Fri, 03 Feb 2023 00:56:29 GMT
<<省略>>
* Connection #0 to host ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com left intact
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
```

- 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

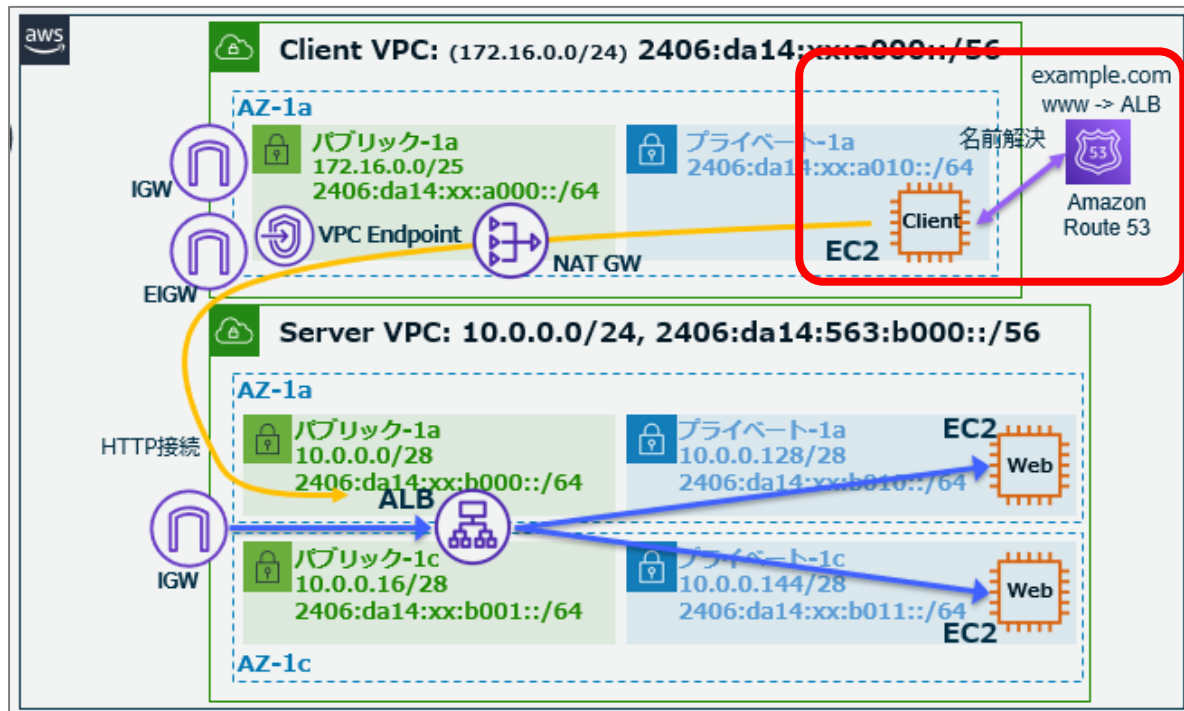
へアクセスします。先ほどの手順で利用した URL の最後に[/index2.php]を加えて再度アクセスします。以下のように接続元 Linux OS にアサインされている IPv6 アドレスが表示されていれば、ALB 接続に際し、IPv6 を使ってアクセスしている状態になります。

curl -v6 http://ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com/index2.php

```
[ssm-user@i-0a22092bc4b32884b ~]$ curl -v6 http://ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com/index2.php
* Trying 2406:da14:9d7:ad00:693a:ffba:da1b:aca7:80...
* Connected to ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com (2406:da14:9d7:ad00:693a:ffba:da1b:aca7) port 80 (#0)
> GET /index2.php HTTP/1.1
> Host: ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com
<<省略>>
<
<html><h1>HTTP_X_FORWARDED_FOR: 2406:da14:342:2901:b8cf:a60d:ba4b:93af</html></h1>
* Connection #0 to host ipv6-handson-alb-1706784423.ap-northeast-1.elb.amazonaws.com left intact
[ssm-user@i-0927c232a12443f15 ~]$
```

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] > [ネットワークとコンテンツ配信] > [Route 53] を選択します。Route 53 のダッシュボードにて[開始する]を選択します。



- ② “開始する”の画面では、[ホストゾーンを作成]を選択し、右下の[開始する]をクリックします。

開始する 情報

開始点を選択

ドメインを登録

アプリケーションにアクセスするためにユーザーが使用する、example.com などの名前を登録します。



ドメインを移管

別のドメインレジストラに登録された Route 53 にドメイン名を移管できます。



ホストゾーンを作成

ホストゾーンは、example.com などのドメインの DNS クエリに応答する方法を Route 53 に指示します。



ヘルスチェックを設定

ヘルスチェックでは、アプリケーションとウェブリソースをモニタリングし、正常なリソースに DNS クエリを誘導します。



トラフィックフローを設定

複雑な設定で複数のエンドポイントのポリシーを簡単に作成できるビジュアルツールです。



リゾルバーを設定

VPC とネットワークの間の DNS クエリをルーティングさせるリージョンのサービスです。



キャンセル 開始する

- 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
この場で、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 アジアパシフィック (東京) [ap-nor... ▼] **VPC ID 情報** vpc-089949a378d5ae6c4 X VPC を削除

VPC を追加

ipv6-handson-client-VPC の名前がついた VPC を選択

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。
タグを追加
タグは最大であと 50 個追加できます。

キャンセル **ホストゾーンの作成**

- 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-xxxxxxxxx.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 .example.com レコードタイプ 情報

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

値 情報 エイリアス

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


推奨値: 60~172800 (2日間)

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

▶ ホストゾンの詳細 ホストゾーンを編集

レコード (3) | ホストゾンのタグ (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。

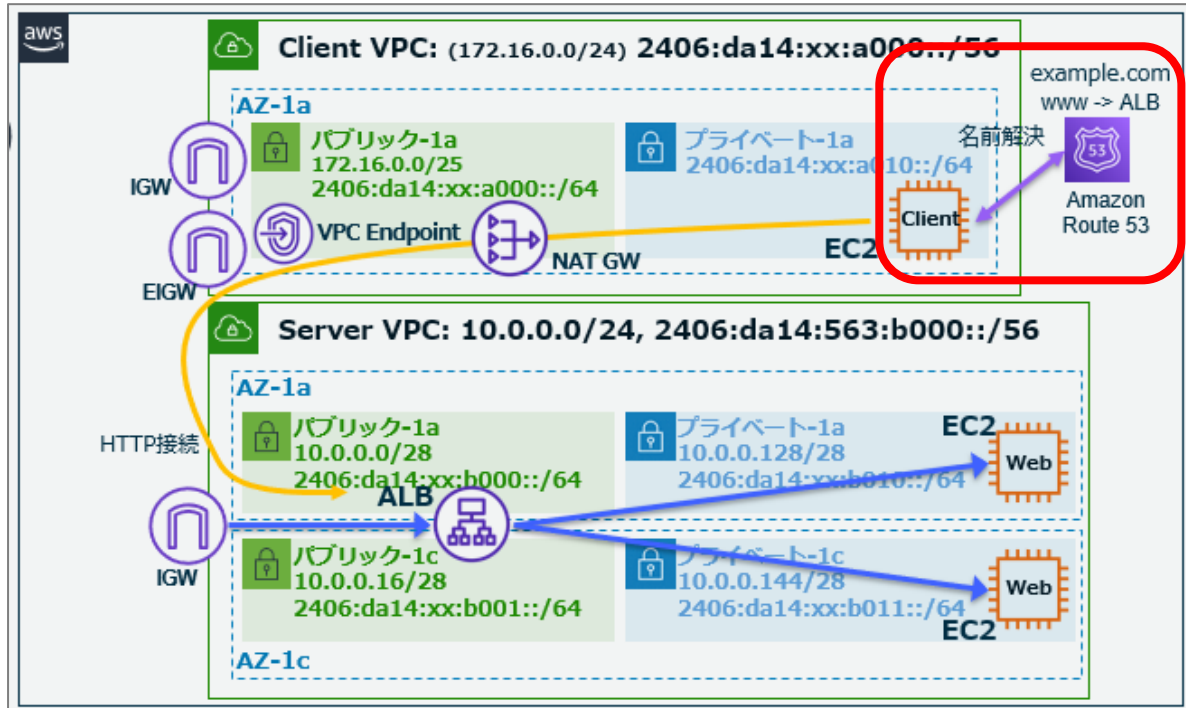
🔄 レコードを削除 ゾーンファイルをインポート レコードを作成

🔍 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  --:--:--  --:--:--  --:--:--    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>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
[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。

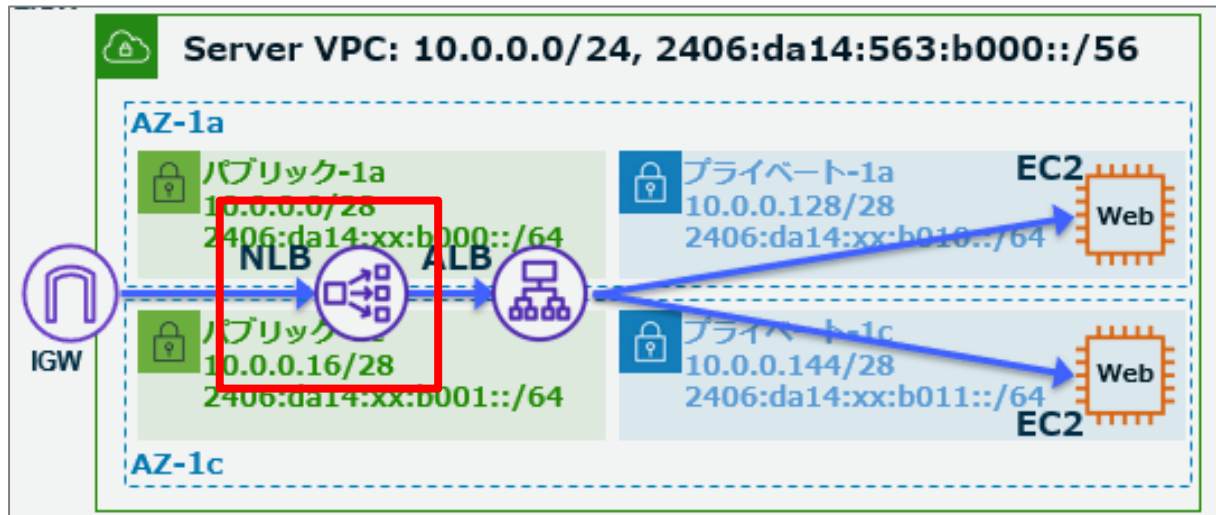
curl -v6 http://www.example.com/index2.php

```
[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（オプション項目）

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。

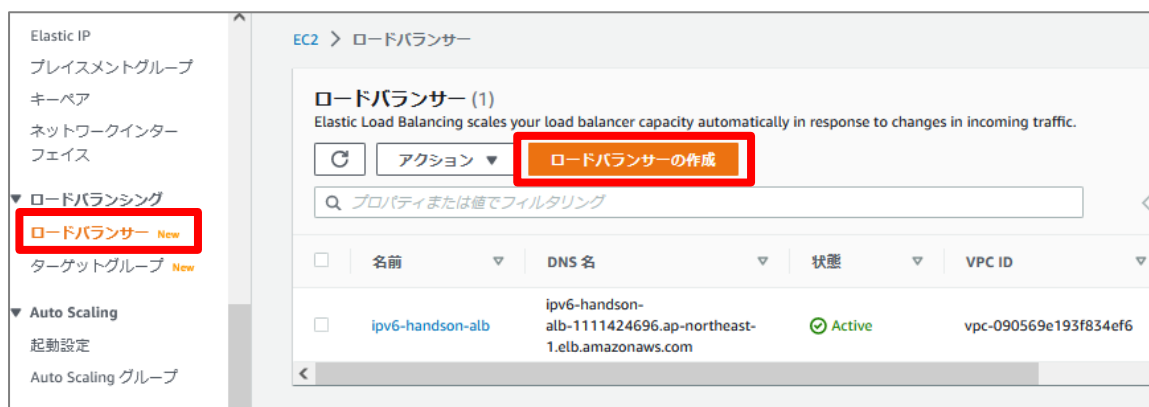


6.1. NLB を作成

- 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。



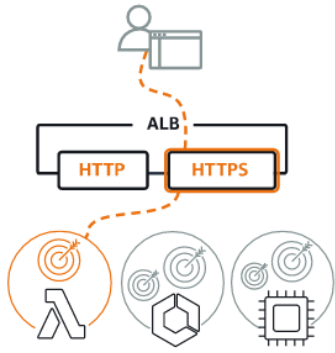
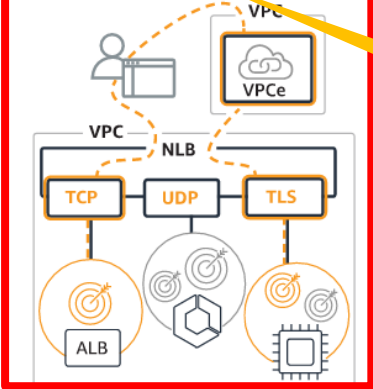

③ “Network Load Balancer”を示す図の下にある[作成]をクリックします。

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

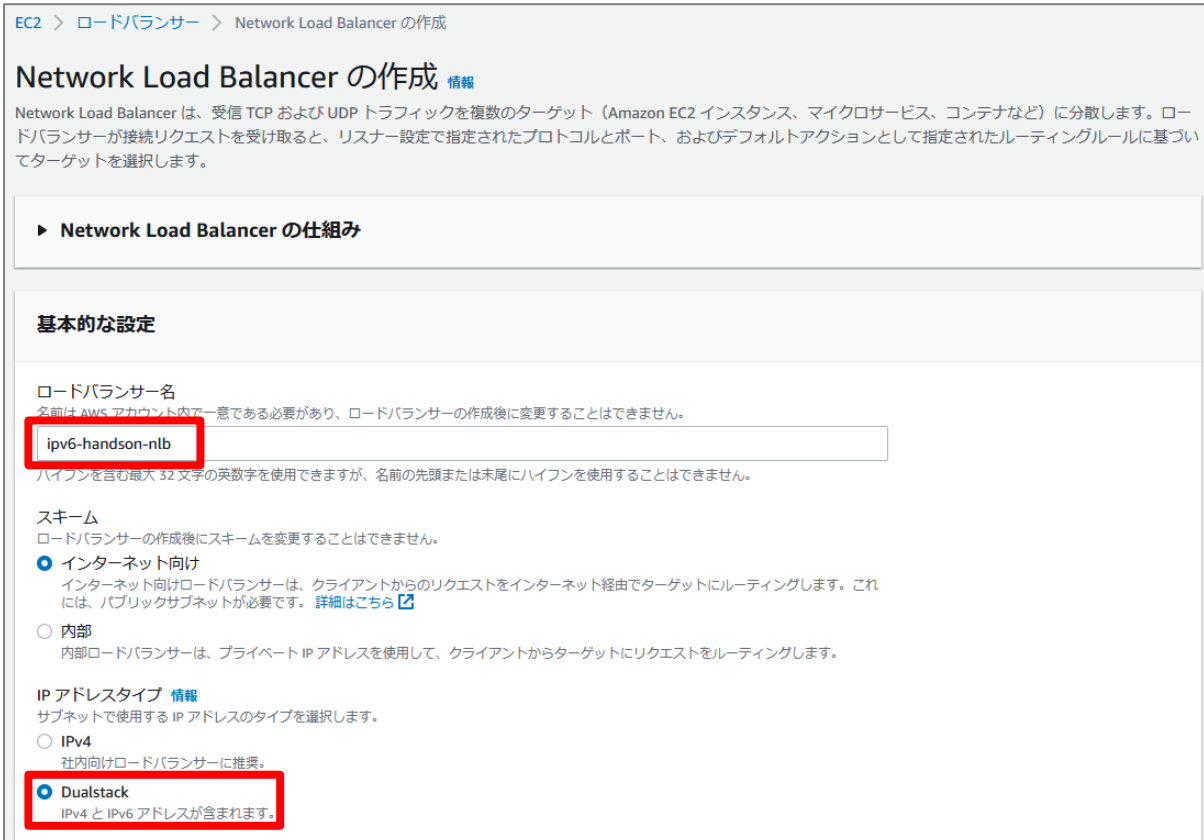
Application Load Balancer 情報	Network Load Balancer 情報	Gateway Load Balancer 情報
 <p>HTTP および HTTPS トラフィックを使用するウェブアプリケーション用に柔軟性の高い機能セットが必要な場合は、Application Load Balancer を選択します。Application Load Balancer はリクエストレベルで動作し、マイクロサービスとコンテナを含む、アプリケーションアーキテクチャを対象とした高度なルーティングおよび可視性機能を提供します。</p> <p><input type="button" value="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同時に処理することができます。</p> <p><input type="button" value="作成"/></p>	 <p>GENEVE をサポートするサードパーティー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</p> <p><input type="button" value="作成"/></p>

種別を確認

クリック

▶ 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ます。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 “セキュリティグループ”では[**ipv6-alb-sg**]を利用します。

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。既存のセキュリティグループを選択するか、[新しいセキュリティグループを作成](#)  できます。

セキュリティグループ - 推奨

Security groups support on Network Load Balancers can only be enabled at creation by including at least one security group. You can change security groups after creation. The security groups for your load balancer must allow it to communicate with registered targets on both the listener port and the health check port. For PrivateLink Network Load Balancers, security group rules are enforced on PrivateLink traffic; however, you can turn off inbound rule evaluation after creation within the load balancer's Security tab or using the API.

最大 5 個のセキュリティグループを選択 ▲

ipv6-alb-sg
 sg-0645f77a2ee9f44a9 VPC: vpc-0db0078edf4fa8ac0

launch-wizard-1
 sg-071eb401e85ba38fb VPC: vpc-0db0078edf4fa8ac0

default
 sg-0dd7b18e482925df4 VPC: vpc-0db0078edf4fa8ac0

するルールによって、登録されたター

- ⑦ 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 <small>情報</small>	
TCP ▼	: 80 <small>1~65535</small>	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

- ⑧ 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

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

TCP : 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。

vpc-0ec991b3f0a4d96db
IPv4: 10.0.0.0/24 IPv6: 2406:da14:68a:d600::/56

- ⑨ 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**

- ⑩ “ターゲットの登録”では”リストから”Application Load Balancer を登録”を[今すぐ登録]とし
 たまま、”Application Load Balancer”のプルダウンメニューから、”ipv6-handson-alb”を選
 択します。続いて、[ターゲットグループの作成]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 を登録

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。

今すぐ登録

後で登録

Application Load Balancer
 リストから Application Load Balancer を選択するか、新しい Application Load Balancer を作成してリストを更新し
 て選択します。Application Load Balancer は、ここで [作成できます](#)。🔗

Application Load Balancer を選択 ▲

ipv6-handson-alb
スキーム: internet-facing 状態: active

🔔 選択した Application Load Balancer は、ポート 80 にリスナーを持っている必要がありま
 す。

ロードバランサーを表示

キャンセル

戻る

ターゲットグループの作成

- ⑪ 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た
 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。

EC2 > ターゲットグループ

ターゲットグループ (1/2) 情報 🔄 アクション ターゲットグループの作成

📌	名前	ARN	ポート	プロトコル	ターゲットの種類	ロードバランサー	VPC ID
<input type="checkbox"/>	ipv6-handson-tg	arn:aws:elasticloadbalancin...	80	HTTP	IP	ipv6-handson-alb	vpc-0ec991b3f0a4d96db
<input checked="" type="checkbox"/>	ipv6-handson-tg-alb	arn:aws:elasticloadbalancin...	80	TCP	Application Load Balancer	🔗 関連付けなし	vpc-0ec991b3f0a4d96db

- ⑫ “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。

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: TCP | 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

ターゲット: G

ipv6-handson-tg-alb TCP

ターゲットの種類: Application Load Balancer, IPv4

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、

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

- ⑬ 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](#)

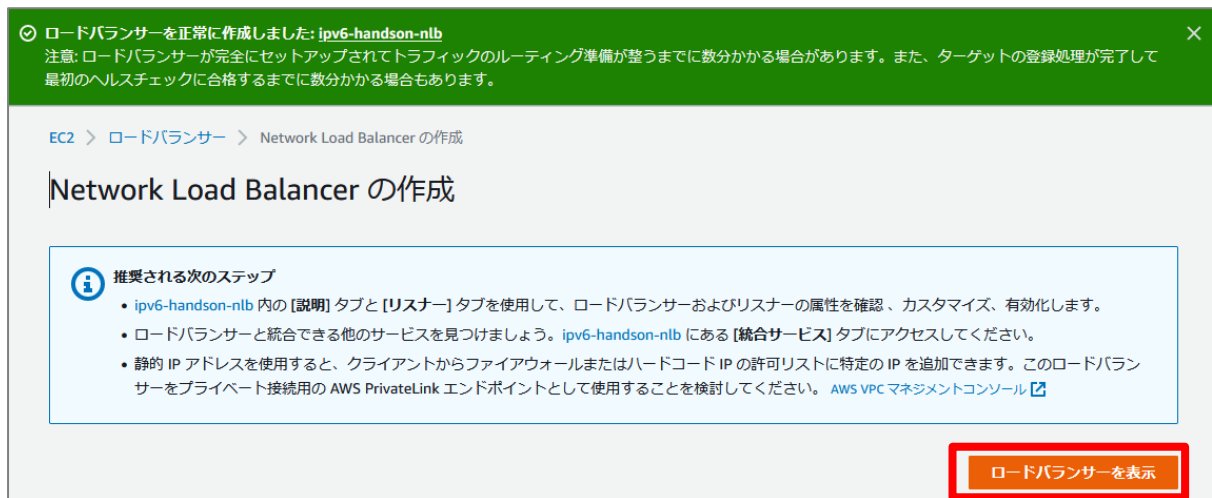
<p>基本的な設定 <small>編集</small></p> <p>ipv6-handson-nlb</p> <ul style="list-style-type: none"> インターネット向け Dualstack 	<p>ネットワークマッピング <small>編集</small></p> <p>VPC vpc-0ec991b3f0a4d96db</p> <p>ipv6-handson-server-vpc</p> <ul style="list-style-type: none"> ap-northeast-1a subnet-0058163584257900b ipv6-handson-server-subnet-public1-ap-northeast-1a ap-northeast-1c subnet-016f7c1349dc3c670 ipv6-handson-server-subnet-public2-ap-northeast-1c 	<p>リスナーとルーティング <small>編集</small></p> <ul style="list-style-type: none"> TCP:80 次をデフォルトにする: ipv6-handson-tg-alb 	<p>タグ <small>編集</small></p> <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。

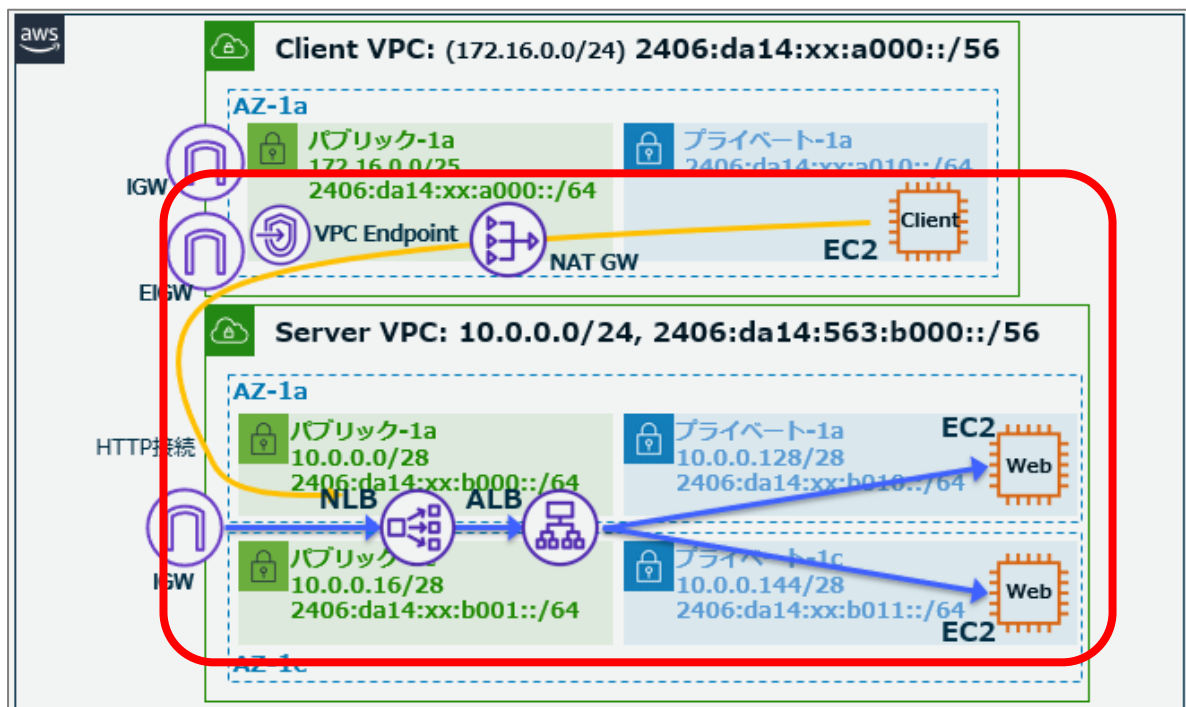
キャンセル ロードバランサーの作成

- ⑭ 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。



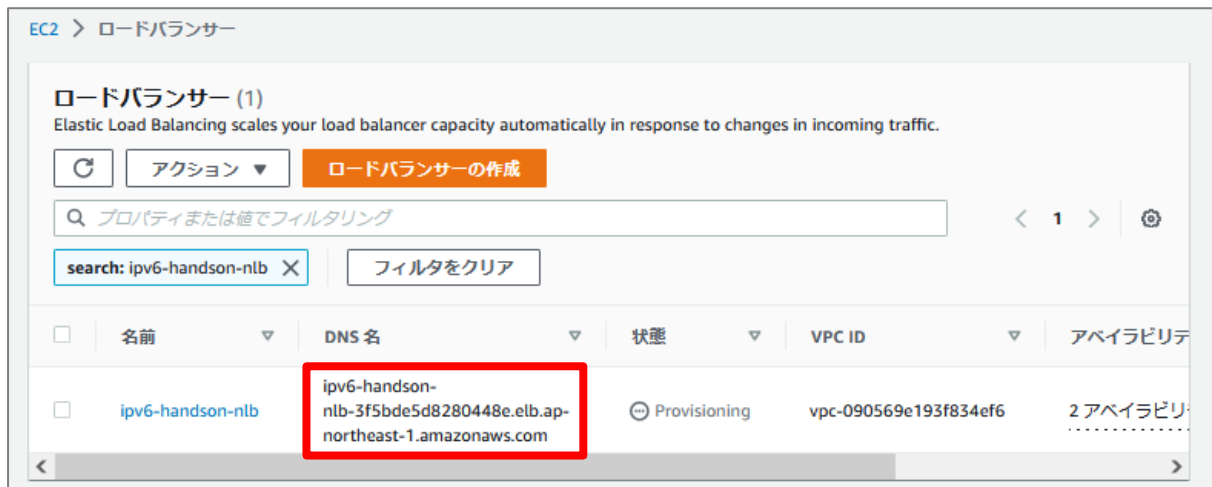
6.2. NLB への接続確認

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   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>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
[ssm-user@i-0a22092bc4b32884b ~]$
```

参考 : NLB から ALB にリクエストを転送する際には、NLB のプライベート IPv4 アドレスが利用されます。このため、curl で、URL の末尾に index2.php を付与したリクエストを実施した場合、NLB の各サブネットでアサインされた IPv4 アドレスが、アクセス元として表示されます。

```
[ssm-user@i-00bb239182105a70a ~]$ curl -v6 http://ipv6-handson-nlb-87ef76047b5b79ff.elb.ap-northeast-1.amazonaws.com/index2.php
* Trying [2406:da14:f3a:ca01:78f3:e62d:f29c:177d]:80...
* Connected to ipv6-handson-nlb-87ef76047b5b79ff.elb.ap-northeast-1.amazonaws.com (2406:da14:f3a:ca01:78f3:e62d:f29c:177d) port 80
<<省略>>
<html><h1>HTTP_X_FORWARDED_FOR: 10.0.0.28</html></h1>
* Connection #0 to host ipv6-handson-nlb-87ef76047b5b79ff.elb.ap-northeast-1.amazonaws.com left intact
[ssm-user@i-00bb239182105a70a ~]$
```

- 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。

[ipv4]

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

[ipv6]

```
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 ~]$
```

参考 : Apache Bench を実行して、ALB を Scale Out を観測する例 :

httpd パッケージに含まれる Apache Bench をインストール

```
$ sudo yum install httpd
```

以下のコマンドを複数回実行し、負荷を発生させる

```
$ ab -n 20000000 -c 10000 http://ipv6-handson-nlb-*****.elb.ap-northeast-1.amazonaws.com/
```

以下のコマンドを別のターミナルで実行し、名前解決で得られるホストの数から ALB と NLB のスケーリング状況を確認 (NLB はスケールアウトせず、同一の IP で負荷を処理する) します。

なお、ALB 自体の性能が向上し、前述のコマンドだけでは ALB のスケールアウトが確認できない場合があります。その場合、バックエンドの Web Server を増やし、apache bench を実行するクライアントホストも追加する必要があります。（主催者から提供している AWS アカウントを利用した、過度な負荷テストはお控えください。）

ALB

```
$ watch -d "dig ipv6-handson-alb-****.ap-northeast-1.elb.amazonaws.com aaaa +short | sort"
```

NLB

```
$ watch -d "dig ipv6-handson-nlb-*****.elb.ap-northeast-1.amazonaws.com aaaa +short | sort"
```

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	2023.02.03	Nobuaki Kikuchi	Advance 版を基に、DNS64/NAT64 を削除、NLB をオプション扱いに変更
1.2	2023.10.09	Nobuaki Kikuchi	ALB の Target にて Instance Type に対応