



AWS IPv6 体験

AWS で作る IPv6 対応 Web サイト ハンズオンテキスト(Linux CLI 版)

Document Version	1.0	Effective Date	2021.11.17
Creation Date	2021.11.17	Document Author	Nobuaki Kikuchi
Last Revision Date	2021.11.17	Revised By	Nobuaki Kikuchi

目次

1. イントロダクション	3
1.1. 目的.....	3
1.2. 対象者.....	3
1.3. 準備物.....	3
2. はじめに	4
2.1. 操作環境の確認.....	4
3. 擬似オンプレミス環境として接続元環境を作成	6
3.1. 接続元 VPC 環境.....	6
3.2. 踏み台用 VPC 環境の作成.....	6
3.3. Linux OS へのログイン.....	14
4. 公開ウェブサーバー環境として接続先環境を作成	19
4.1. 接続先 VPC 環境.....	19
4.2. VPC 内のネットワーク環境作成.....	19
4.3. Web サーバの配置.....	30
4.4. Web サーバへの接続テスト.....	38
4.5. 冗長化用 Web サーバ作成.....	41
4.6. Elastic Load Balancer (ALB)の設定.....	46
4.7. ALB への接続確認.....	60
5. DNS で IPv6 を FQDN に登録する	63
5.1. Route 53 でプライベートホストゾーンの登録.....	64
5.2. Linux OS から名前解決の確認.....	69
6. ハンズオン環境のクリーニング	72
6.1. クリーニング対象リソース.....	72
7. ドキュメント情報	73
7.1. 更新履歴.....	73

1. イントロダクション

1.1. 目的

IPv4/IPv6 のハイブリッド環境を 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 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 にアクセスいただきます。

このセミナーを通し、IPv6 環境を身近に感じて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。

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作業ができる方が対象です。

(過去に AWS 利用経験が無い方でも、ハンズオンテキストをご覧いただければ手順に沿って進める事が可能です。)

また、講師の説明を参考にお手元のパソコンから SSM で AWS 上の Linux ホストにログインできる方。

1.3. 準備物

パソコン :

Linux OS、Mac OS 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

受信可能なメールアドレス :

認証時にメール受信が必要となります

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。



The screenshot shows the AWS Management Console homepage. At the top, there is 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', and a highlighted 'コンソールにサインイン' button. Below the navigation bar, there are links for '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', and 'カスタマーサポート'. The main heading is 'AWS マネジメントコンソール' with a sub-heading 'AWS クラウドにアクセスして管理するために必要なものを 1 つのウェブインターフェイスに集結'. Below this is a 'もう一度ログインする' button. The 'AWS の詳細を見る' section contains four featured articles:

- 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デベロッパー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:

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?

AWS でのワークロードの 起動に役立つリソースセンター

どなたでも簡単にAWSを開始できるチュートリアルや
中・上級者向けのユースケース別ガイド、トレーニング等
をご活用ください

詳細はこちら »

日本語

利用規約 プライバシーポリシー © 1996-2021, Amazon Web Services, Inc. or its affiliates.

aws サービス

サービス、機能、マーケットプレースの製品、ドキュメント [Alt+S]

AWS マネジメントコンソール

AWS のサービス

▼ 最近アクセスしたサービス

- CloudFormation
- VPC
- EC2

▼ すべてのサービス

- コンピューティング
 - EC2
 - Lightsail
 - Lambda
 - Batch
 - Elastic Beanstalk
 - Serverless Application Repository
 - AWS Outposts
 - EC2 Image Builder
- コンテナ
 - Elastic Container Registry
 - Elastic Container Service
 - Elastic Kubernetes Service
- Machine Learning
 - Amazon SageMaker
 - Amazon Augmented AI
 - Amazon CodeGuru
 - Amazon DevOps Guru
 - Amazon Comprehend
 - Amazon Forecast
 - Amazon Fraud Detector
 - Amazon Kendra
 - Amazon Lex
 - Amazon Personalize
 - Amazon Polly
 - Amazon Rekognition
 - Amazon Textract
 - Amazon Transcribe

外出先でも AWS リソースに常時接続

AWS コンソールモバイルアプリを iOS または Android モバイルデバイスにダウンロードします。詳細はこちらから

AWS を試す

Amazon Redshift

データレイクにクエリを拡張できる、高速かつシンプルで、費用対効果の高いデータウェアハウス。詳細はこちらから

AWS Fargate を使ってサーバーレスコンテナを実行

AWS Fargate が実行され、サーバーやクラスターを管理することなく、コンテナをスケールアップします。詳細はこちらから

Amazon S3 によるスケールアップで耐久性に優れたセキュアなバックアップと復元

他のお客様が AWS でどのようにコストを削減するバックアップおよび復元ソリューションを構築して

フィードバック 日本語

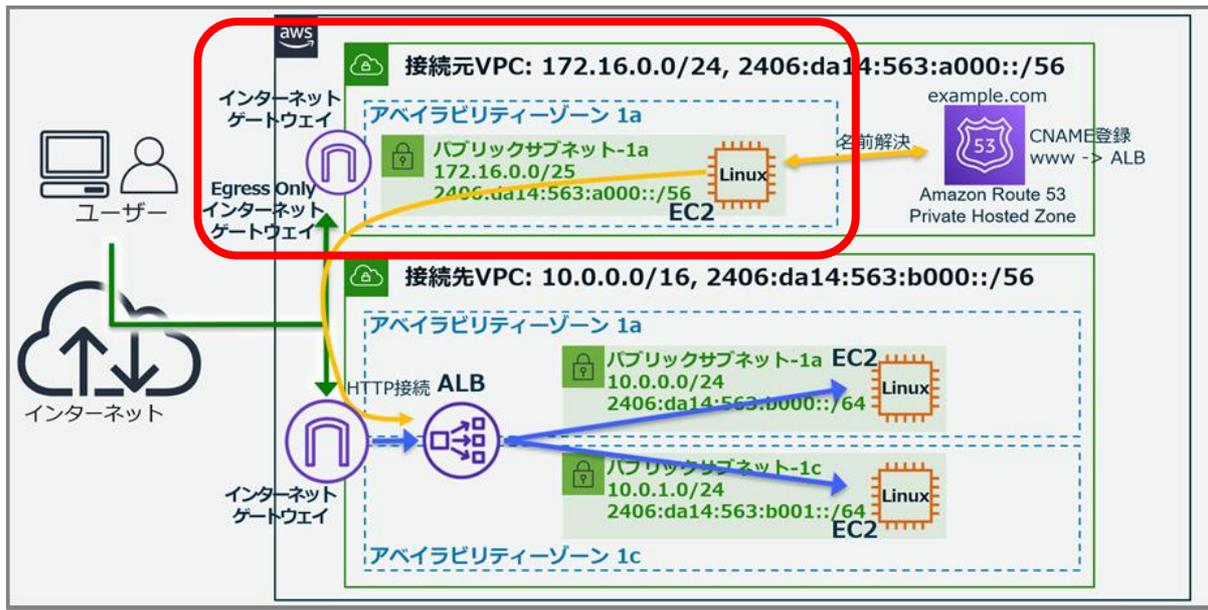
© 2008 - 2021, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約 Cookie の設定

3. 擬似オンプレミス環境として接続元環境を作成

3.1. 接続元 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、接続元 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 により接続元 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てください。



- 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 interface for creating a stack. The page title is "スタックの作成" (Stack Creation). On the left, there is a navigation pane with steps: "ステップ1 テンプレートの指定" (Step 1: Specify template), "ステップ2 スタックの詳細を指定" (Step 2: Specify stack details), "ステップ3 スタックオプションの設定" (Step 3: Configure stack options), and "ステップ4 レビュー" (Step 4: Review). The main content area is titled "スタックの作成" and contains several sections:

- 前提条件 - テンプレートの準備** (Prerequisites - Template preparation): Includes a section for "テンプレートの準備" (Template preparation) with three radio buttons: "テンプレートの準備完了" (Template preparation complete), "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).
- テンプレートの指定** (Template specification): Includes a section for "テンプレートソース" (Template source) with 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). The "Upload template file" option is selected and highlighted with a red box.
- テンプレートファイルのアップロード** (Upload template file): Includes a "ファイルの選択" (Select file) button, which is also highlighted with a red box. Below it, it says "ファイルが選択されていません" (No file selected) and "JSON または YAML 形式のファイル" (JSON or YAML format file).

At the bottom right, there are buttons for "キャンセル" (Cancel) and "次へ" (Next).

- 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“LinuxIPv6.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

ステップ 2
スタックの詳細を指定

ステップ 3
スタックオプションの設定

ステップ 4
レビュー

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

LinuxIPv6.yaml

JSON または YAML 形式のファイル

S3 URL: `https://s3-ap-northeast-1.amazonaws.com/cf-templates-1f8h5oztswewv-ap-northeast-1/20212988oc-LinuxIPv6.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"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

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。

AllowSshV4ipRange
IPv4 range to allow SSH (0.0.0.0/0 is All allow). If you want to change the allow CIDR, please change this paramater.

LinuxLatestAmi

NameTagPrefix
Prefix of Name tags.

SshConnectionPort
Port for SSH connection (22:Linux SSH Protocol)

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。

IAM ロ...	Sample-role-name	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**

- 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

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-1f8h5oztswewv-ap-northeast-1/20212988oc-LinuxIPv6.yaml

スタックの説明
AWS IPv6 Hands-On (Linux Version)

[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

i 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。

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 main panel shows the 'EVENTS' tab with one event listed:

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

↓

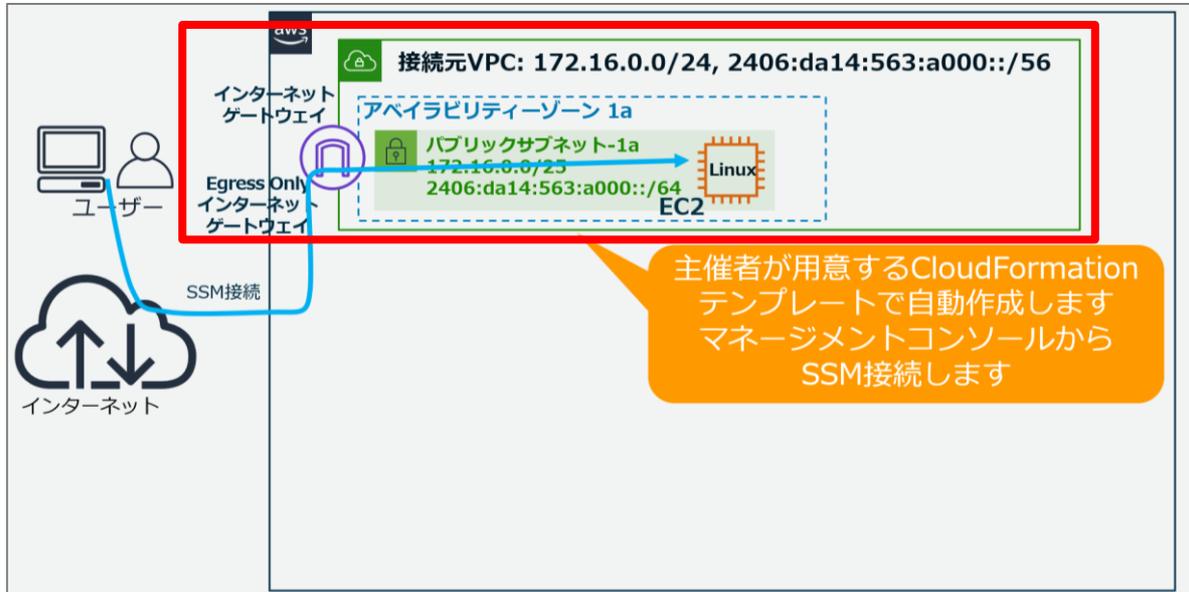
The screenshot shows the AWS CloudFormation console for the stack 'ipv6-handson' after it has completed. The stack is now in the 'CREATE_COMPLETE' state, highlighted with a red box. The main panel shows the 'EVENTS' tab with 44 events listed:

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	BastionLinuxServer	CREATE_COMPLETE	-
2021-10-25 18:08:35 UTC+0900	BastionLinuxServer	CREATE_IN_PROGRESS	Resource creation Initiated

この状態で、接続元 VPC 環境が作成されました。

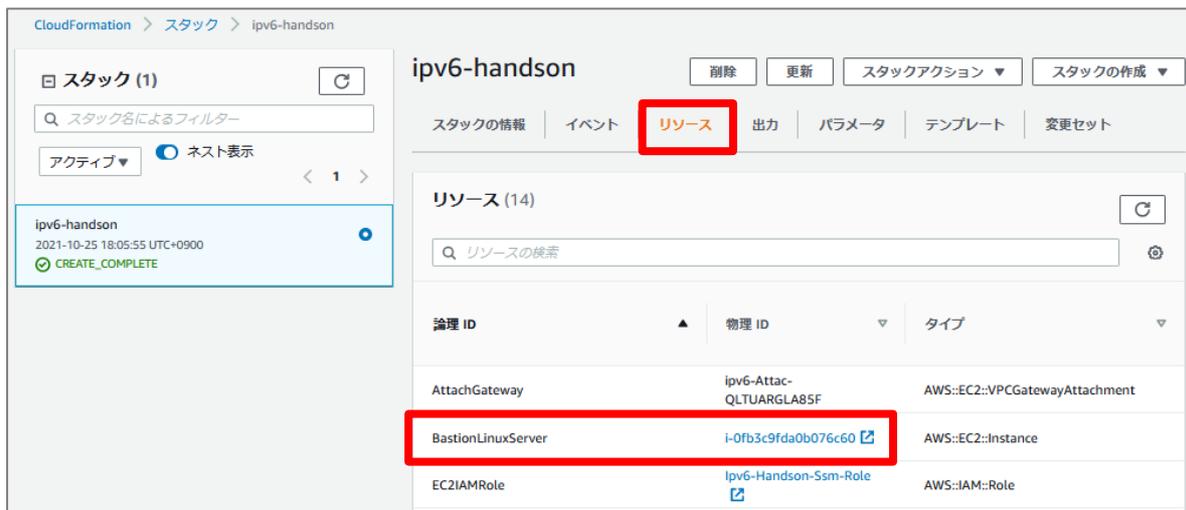
3.3. Linux OS へのログイン

先ほど作成した接続元 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 で接続元 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。



- 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

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. At the top, there are buttons for '接続' (Connect), 'インスタンスの状態' (Instance State), and 'アクション' (Actions). Below these is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance State, Instance Type, Status Checks, and Alarm Status. One instance, 'bastion linux server' with ID 'i-0fb3c9fda0b076c60', is shown with a state of '実行中' (Running). Below the table, a detailed view of the instance is shown, with a red box highlighting the 'インスタンスの状態' (Instance State) field, which also displays '実行中' (Running).

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、画面上的[**接続**]を選択します。

This screenshot is similar to the previous one, but with a red box highlighting the '接続' (Connect) button in the top navigation bar. The instance 'bastion linux server' is still shown in the '実行中' (Running) state in the table below.

- 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-0fb3c9fda0b076c60 > 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-0fb3c9fda0b076c60 (bastion linux server) 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。



セッション ID: MasterKey-00b5b1f5b4a85e82f インスタンス ID: i-0fb3c9fda0b076c60 終了

```
sh-4.2$
```

- ⑥ ifconfig コマンドを利用して、Linux ホストに IPv4/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。

ifconfig eth0

```
sh-4.2$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.16.0.10 netmask 255.255.255.128 broadcast 172.16.0.127
    inet6 fe80::406:b7ff:febc:a4b prefixlen 64 scopeid 0x20<link>
    inet6 2406:da14:2b7:500:8434:75ee:5d3d:b25c prefixlen 128 scopeid 0x0<global>
    ether 06:06:b7:bc:0a:4b txqueuelen 1000 (Ethernet)
    RX packets 45547 bytes 63910623 (60.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22785 bytes 1531171 (1.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sh-4.2$
```

- ⑦ 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 https://www.v6pc.jp/
```

```
sh-4.2$ curl -v6 https://www.v6pc.jp/
* Trying 2400:6700:ff00::36f8:7262:443...
* Connected to www.v6pc.jp (2400:6700:ff00::36f8:7262) port 443 (#0)
<<中略>>
> User-Agent: curl/7.76.1
> Accept: */*
<<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$
```

- ⑧ 参考：その他の IPv6 接続を確認できるサイト、コマンド

```
https://www.kame.net/
```

```
https://www.kddi.com/
```

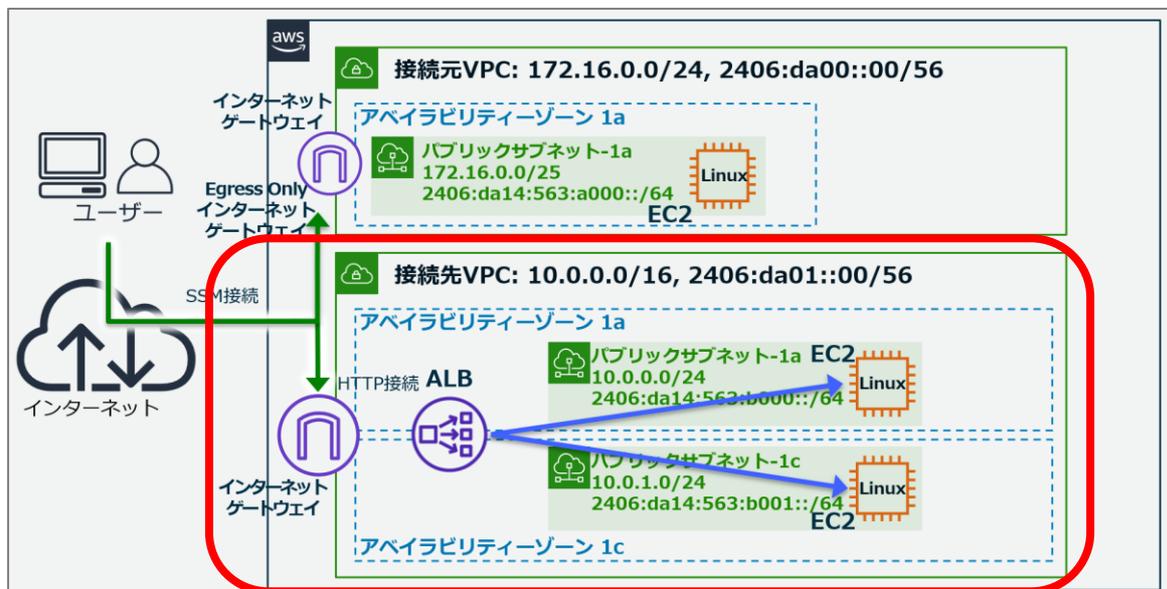
```
traceroute6 www.kddi.com
```

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 OS への SSM 接続環境は、後ほど利用します。後続の作業に差し支えなければ、ブラウザのタブを取っておくことができますが、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、ブラウザのタブを閉じ、再度 [接続] をクリックしてください。

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] > [ネットワークとコンテンツ配信] > [VPC]を選択し、VPC ダッシュボードを表示します。

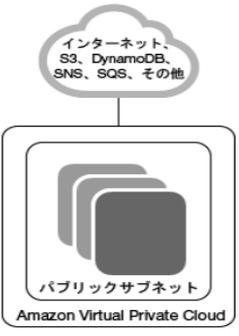


- ② [VPC ウィザードの起動]をクリックします。



- ③ “ステップ 1”の画面で、一番上の**“1 個のパブリックサブネットを持つ VPC”**が選択されていることを確認の上、**[選択]**をクリックします。

ステップ 1: VPC 設定の選択

1 個のパブリックサブネットを持つ VPC	インターネットに直接アクセスできる AWS クラウドの分離されたプライベート空間でインスタンスは実行されます。インスタンスのインバウンドおよびアウトバウンドのネットワークトラフィックを厳重に管理するには、ネットワークアクセス ACL とセキュリティグループを使用します。	
パブリックとプライベートサブネットを持つ VPC	作成: /24 サブネットを持つ /16 ネットワークです。パブリックサブネットインスタンスは Elastic IP またはパブリック IP を使用してインターネットにアクセスします。	
パブリックとプライベートサブネットおよびハードウェア VPN アクセスを持つ VPC	Important: If you are using a Local Zone with your VPC follow this link to create your VPC.	
プライベートのサブネットのみで、ハードウェア VPN アクセスを持つ VPC		選択

④ “ステップ 2”の画面で、以下の項目を設定します。

設定項目	パラメータ	要修正
IPv4 CIDR ブロック	10.0.0.0/16	
IPv6 CIDR ブロック	● Amazon が提供した IPv6 CIDR ブロック	レ
VPC 名	IPv6-Web-VPC	レ
パブリックサブネットの IPv4 CIDR	10.0.0.0/24	
パブリックサブネットの IPv6 CIDR	カスタム IPv6 CIDR を指定する	レ
	xxxx:xxxx:xxxx:xx[00]::/64	
アベイラビリティゾーン	ap-northeast-1a	レ
サブネット名	IPv6-Web-Subnet-1a	レ
DNS ホスト名を有効化	はい	
ハードウェアのテナンシー	デフォルト	

ステップ 2: 1 個のパブリックサブネットを持つ VPC

IPv4 CIDR ブロック: 10.0.0.0/16 (65531 利用可能な IP アドレス)

IPv6 CIDR ブロック: IPv6 CIDR ブロックなし
 Amazon が提供した IPv6 CIDR ブロック
 IPv6 CIDR block owned by me

VPC 名: IPv6-Web-VPC

パブリックサブネットの IPv4 CIDR: 10.0.0.0/24 (251 利用可能な IP アドレス)

パブリックサブネットの IPv6 CIDR: カスタム IPv6 CIDR を指定する
 xxxx:xxxx:xxxx:xx[00]::/64

アベイラビリティゾーン: ap-northeast-1a

サブネット名: IPv6-Web-Subnet-1a

VPC を作成した後は、より多くのサブネットを追加できます。

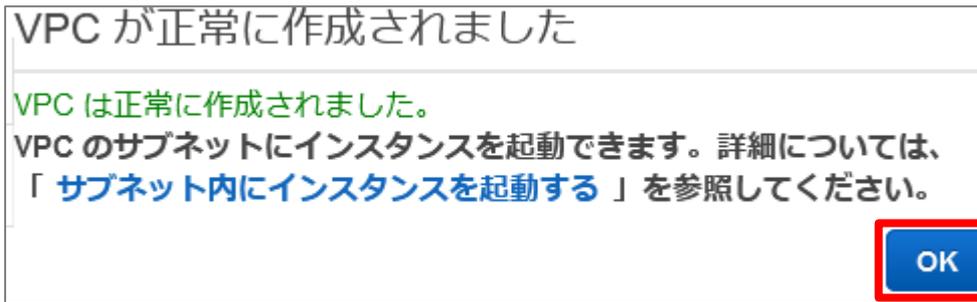
サービスエンドポイント

DNS ホスト名を有効化: はい いいえ

ハードウェアのテナンシー: デフォルト

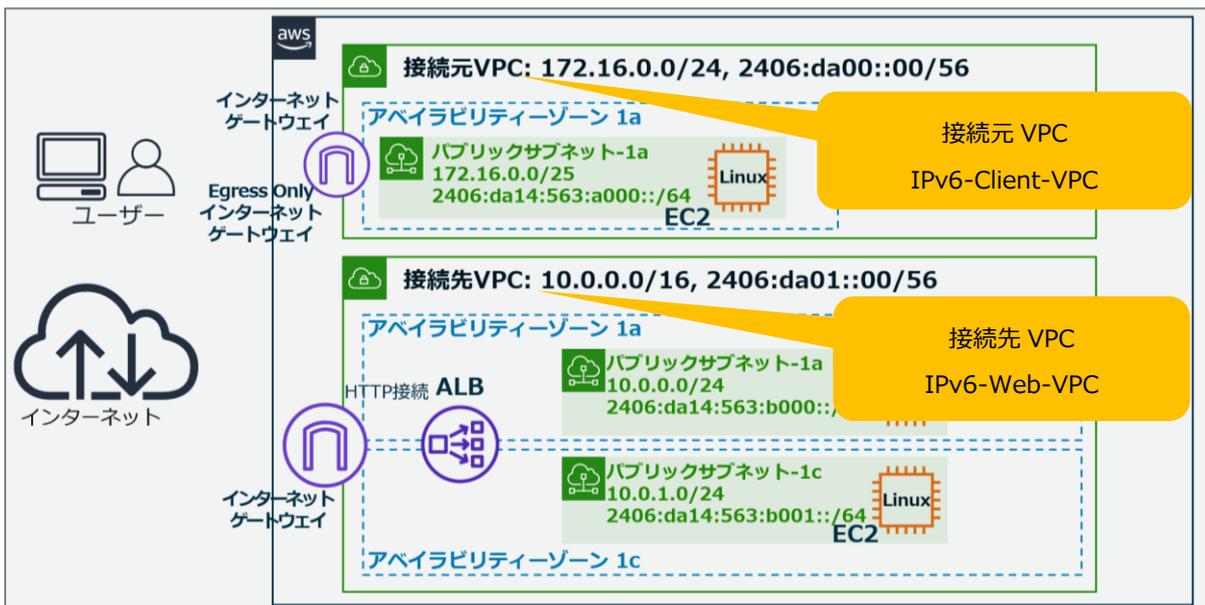
キャンセルして終了

- ⑤ VPC が正常に作成された旨が表示されます。右下の[OK]をクリックしてください。



- ⑥ このステップまで完了すると、接続元 VPC と公開用 Web サイトを配置する 2 つの VPC が作成されます。以下の状態となっていることを確認してください。（その他に、アカウント作成時に登録される Default VPC[172.31.0.0/16]が表示されますが、こちらは利用しません。）

お使いの VPC (3) 情報					
Name	VPC ID	状態	IPv4 CIDR	IPv6 CIDR (ネットワークポーターグループ)	
IPv6-Web-VPC	vpc-0ad9f362d8a8011ba	Available	10.0.0.0/16	2406:da14:a42:5400::/56 (ap-northeast-1)	Associated
IPv6-Client-VPC	vpc-05afab097cc27b423	Available	172.16.0.0/24	2406:da14:2b7:5000::/56 (ap-northeast-1)	Associated
-	vpc-021293269f81833e1	Available	172.31.0.0/16	-	



- ⑦ 作成した“IPv6-Web-VPC”内に、冗長化のため、2つ目のサブネットを作成します。VPCダッシュボードのナビゲーションペイン（画面左側に表示されるメニュー）から“サブネット”を選択します。



- ⑧ この段階では、Web サーバ設置用の IPv6-Web-VPC には“IPv6-Web-Subnet-1a”のサブネットのみが作成されています。右上の[サブネットを作成]をクリックします。

The screenshot shows the AWS VPC console displaying a list of subnets. The 'サブネットを作成' (Create Subnet) button is highlighted with a red box. The table below shows the list of subnets.

<input type="checkbox"/>	Name	サブネット ID	状態	VPC	IPv4 CIDR
<input checked="" type="checkbox"/>	IPv6-Client-HybridPublicSubnet	subnet-02cb9d669cf6af536	Available	vpc-05afab097cc27b423 IPv6-Client-V...	172.16.0.0/25
<input type="checkbox"/>	IPv6-Web-Subnet-1a	subnet-021d1982871feaf87	Available	vpc-0ad9f362d8a8011ba IPv6-Web-VPC	10.0.0.0/24
<input type="checkbox"/>	-	subnet-06c7d6b84b2f6cba1	Available	vpc-021293269f81833e1	172.31.32.0/20
<input type="checkbox"/>	-	subnet-0805f9fd1749fbf27	Available	vpc-021293269f81833e1	172.31.0.0/20
<input type="checkbox"/>	-	subnet-0d90dafdc8567c947	Available	vpc-021293269f81833e1	172.31.16.0/20

⑨ “サブネットを作成”の画面で、以下の内容を設定します。

設定項目	パラメータ	要修正
VPC ID	“IPv6-Web-VPC”の名前がついた VPC ID	レ
サブネット名	IPv6-Web-Subnet-1c	レ
アベイラビリティゾーン	ap-northeast-1c	レ
IPv4 CIDR ブロック	10.0.1.0/24	レ
IPv6 CIDR ブロック	カスタム Ipv6	レ
	xxxx:xxxx:xxxx:xx[01]::/64	レ

VPC > サブネット > サブネットを作成

サブネットを作成 情報

VPC

VPC ID
この VPC にサブネットを作成します。

vpc-0ad9f362d8a8011ba (IPv6-Web-VPC) ▲

Q |

vpc-0ad9f362d8a8011ba (IPv6-Web-VPC)
10.0.0.0/16 2406:da14:a42:5400::/56 (ap-northeast-1)

vpc-05afab097cc27b423 (IPv6-Client-VPC)
172.16.0.0/24 2406:da14:2b7:500::/56 (ap-northeast-1)

vpc-021293269f81833e1 (デフォルト)
172.31.0.0/16

(次の画像に続く)

サブネットの設定

サブネットの CIDR ブロックとアベイラビリティゾーンを指定します。

サブネット 1 (1 個中)

サブネット名
「Name」というキーと、指定した値を使用してタグを作成します。

名前の長さは最大 256 文字です。

アベイラビリティゾーン [情報](#)
サブネットが存在するゾーンを選択するか、Amazon が選択するゾーンを受け入れます。

IPv4 CIDR ブロック [情報](#)

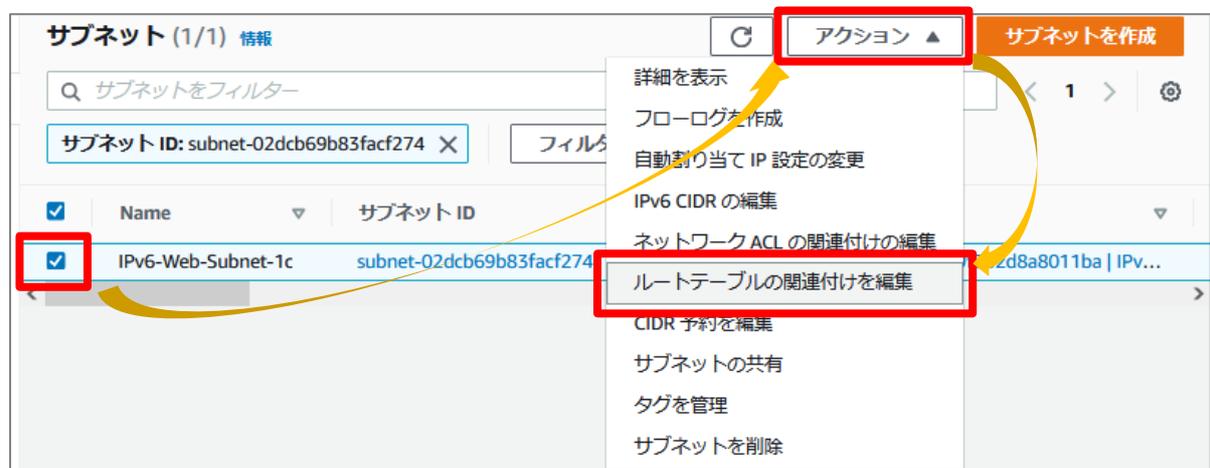
IPv6 CIDR ブロック [情報](#)
IPv6 CIDR ブロックをサブネットに割り当てるかどうかを指定します。

▼ タグ - オプション

キー	値 - オプション
<input type="text" value="Name"/>	<input type="text" value="IPv6-Web-Subnet-1c"/>

さらに 49 個のタグ を追加できます。

- ⑩ 正常にサブネットが作成された旨が表示されれば成功です。続けて、ルートテーブルを変更します。“IPv6-Web-Sunet-1c”の左側のボックスにチェックを入れ、右上の[アクション] > [ルートテーブルの関連付けを編集]を選択します。



- ⑪ “ルートテーブル ID”のプルダウンメニューから“メインルートテーブル”の表記が無い項目を選択します。このルートテーブルは先ほどの VPC ウィザードで作成され、すでにもう一方のサブネットでも利用されています。デフォルトゲートウェイとして、IPv4 の“0.0.0.0/0”および IPv6

の "::/0" に対し、"igw-xxxx..." から始まるインターネットゲートウェイのリソース ID が指定されていることが確認できます。右下の[保存]をクリックします。

VPC > サブネット > subnet-02dcb69b83facf274 > ルートテーブルの関連付けを編集

ルートテーブルの関連付けを編集 情報

サブネットルートテーブル設定

サブネット ID
☰ subnet-02dcb69b83facf274

ルートテーブル ID
 🔄

メインルートテーブル

"メインルートテーブル" である事との表記

Security Analyzer の実行 ✕

ルート (4)

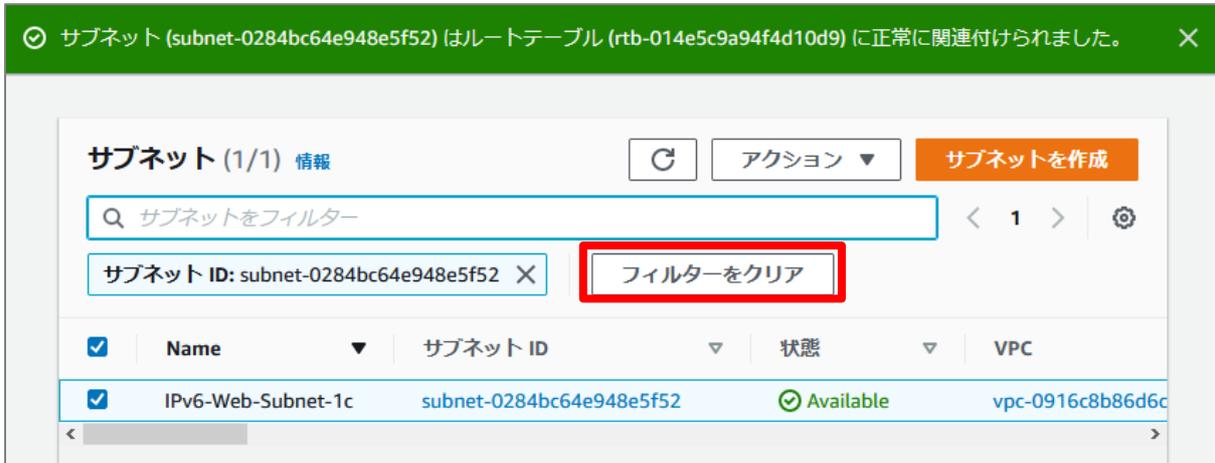
< 1 > ⚙️

送信先	ターゲット
10.0.0.0/16	local
0.0.0.0/0	igw-0b9a8cf406b422d88
2406:da14:a42:5400::/56	local
::/0	igw-0b9a8cf406b422d88

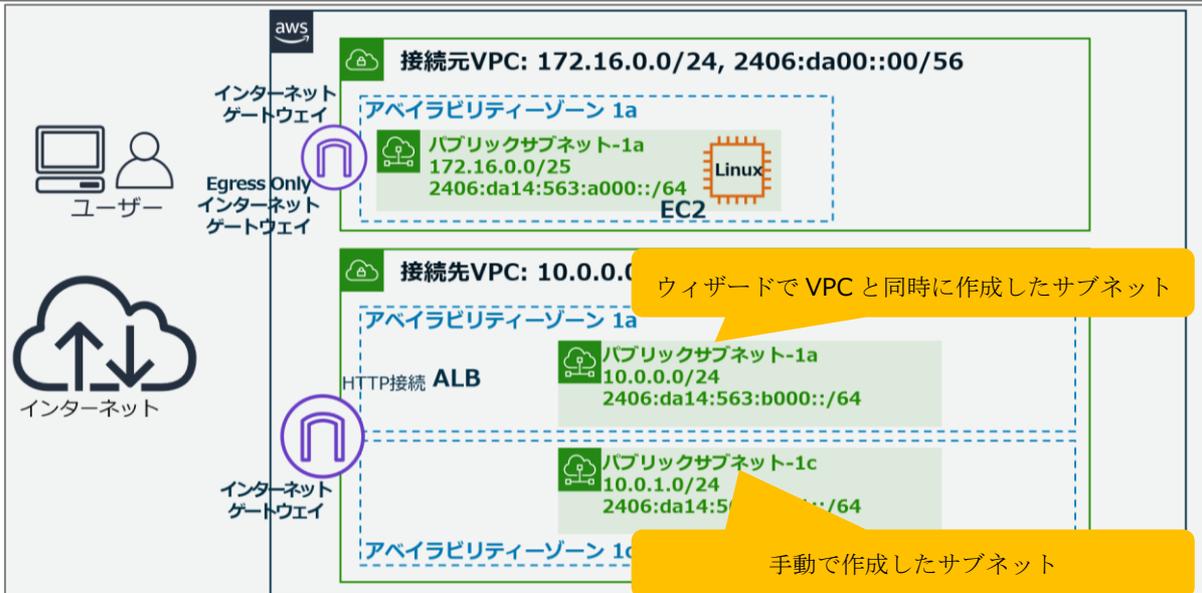
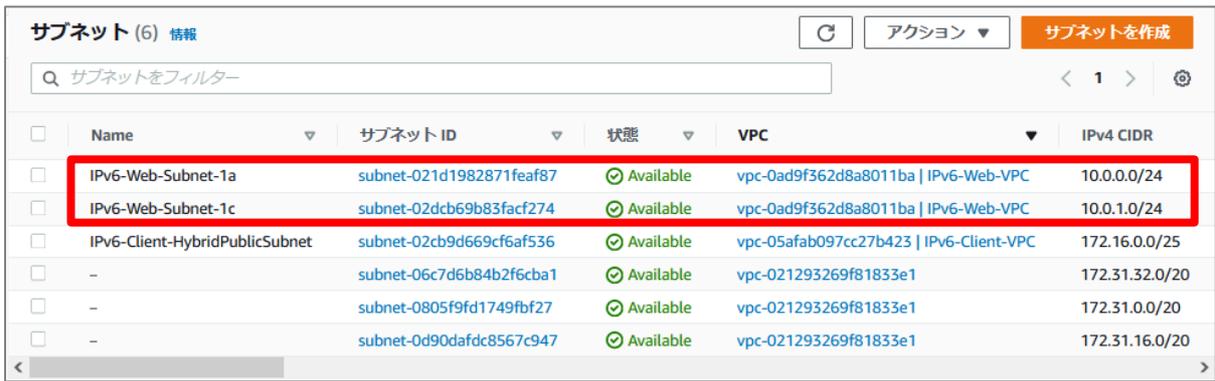
"igw-xxx" から始まるターゲットがあることを確認

キャンセル 保存

- ⑫ サブネットの変更が完了した旨のメッセージが表示されます。[フィルターをクリア]をクリックし、サブネットの一覧を確認します。

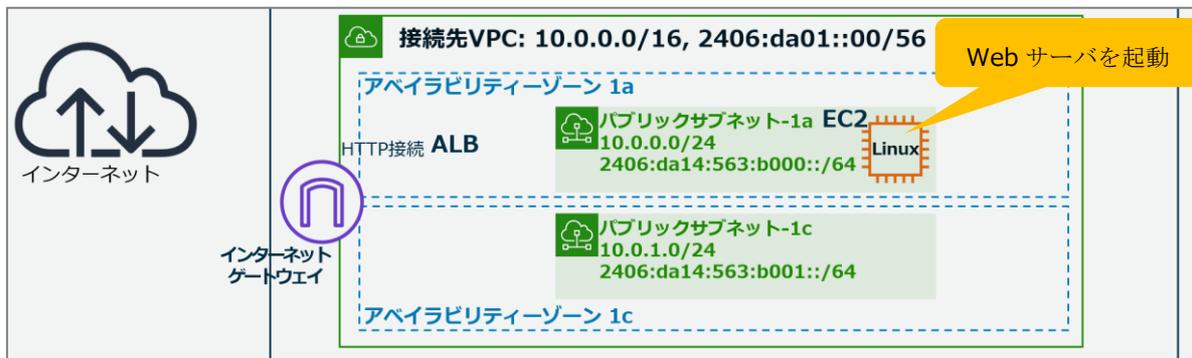


- ⑬ 接続先 VPC(IPv6-Web-VPC)内に 2 つのサブネットが作成されていることを確認します。
(Default VPC 内に 3 つのサブネットが作成されていますが、本ハンズオンでは利用しません。)



4.3. Web サーバの配置

先ほど作成したサブネット内に、Web サーバを起動します。



- ① 画面左上の[サービス] > [EC2]をクリックし、EC2 ダッシュボードに移動します。画面中央から[インスタンスを起動] > [インスタンスの起動]をクリックしてください。



以下の画面と異なる場合、画面左のナビゲーションペインから[EC2 ダッシュボード]を選択します。



② “ステップ 1”では“Amazon Linux 2”の右にある[選択]をクリックします。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 1: Amazon マシンイメージ (AMI) キャンセルして終了

AMI は、インスタンスの作成に必要なソフトウェア構成 (OS、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。AMI は、AWS が提供するもの、ユーザーコミュニティが提供するもの、または AWS Marketplace に掲載されているものを選択できます。独自の AMI のいずれかを選択することもできます。

検索用語を入力して AMI を検索します (「Windows」など) Systems Manager パラメータによる検索

クイックスタート 39 AMI 中の 1 ~ 39

マイ AMI

AWS Marketplace

コミュニティ AMI

無料利用枠のみ ⓘ

 Amazon Linux 無料利用枠の対象	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-09d28faae2e9e7138 (64 ビット x86) / ami-0661694c01306dbc9 (64 ビット Arm) Amazon Linux 2 には 5 年間のサポートが含まれます。Amazon EC2、systemd 219、GCC 7.3、Glibc 2.26、Binutils 2.29.1 で最適なパフォーマンスを発揮できるように調整された Linux カーネル 4.14、および、追加の最新のソフトウェアパッケージを提供します。 ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい	<input checked="" type="radio"/> 64 ビット (x86) <input type="radio"/> 64 ビット (Arm) 選択
 Windows 無料利用枠の対象	Microsoft Windows Server 2019 Base - ami-08e18b2fe66cdf6ab Microsoft Windows 2019 Datacenter edition. [English] ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい	選択 64 ビット (x86)
 Red Hat 無料利用枠の対象	Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0dc185deadd3ac449 (64 ビット x86) / ami-0be00ad3801d80205 (64 ビット Arm) Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい	<input checked="" type="radio"/> 64 ビット (x86) <input type="radio"/> 64 ビット (Arm) 選択

- ③ “ステップ 2”ではデフォルトの“t2.micro”が選択された状態で画面右下の[次のステップ: インスタンス詳細の設定]をクリックします。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 2: インスタンスタイプの選択

Amazon EC2 では、異なるユースケースに合わせて最適化されたさまざまなインスタンスタイプが用意されています。インスタンスとは、アプリケーションを実行できる仮想サーバーです。インスタンスタイプはさまざまな CPU、メモリ、ストレージ、ネットワークキャパシティの組み合わせによって構成されているため、使用するアプリケーションに合わせて適切なリソースの組み合わせを柔軟に選択できます。インスタンスタイプおよびそれをコンピューティングのニーズに適用する方法に関する [詳細はこちら](#)。

フィルター条件: すべてのインスタンスファミリー 現行世代 列の表示/非表示

現在選択中: t2.micro (- ECU, 1 vCPU, 2.5 GHz, -, 1 GiB メモリ, EBS のみ)

	ファミリー	タイプ	vCPU	メモリ (GiB)	インスタンスストレージ (GB)	EBS 最適化利用	ネットワークパフォーマンス	IPv6 サポート
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS のみ	-	低から中	はい
<input checked="" type="checkbox"/>	t2	t2.micro	1	1	EBS のみ	-	低から中	はい
<input type="checkbox"/>	t2	t2.small	1	2	EBS のみ	-	低から中	はい
<input type="checkbox"/>	t2	t2.medium	2	4	EBS のみ	-	低から中	はい
<input type="checkbox"/>	t2	t2.large	2	8	EBS のみ	-	低から中	はい
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS のみ	-	中	はい
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS のみ	-	中	はい
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	t3	t3.micro	2	1	EBS のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	t3	t3.small	2	2	EBS のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	t3	t3.medium	2	4	EBS のみ	はい	最大 5 ギガビット	はい

キャンセル 戻る 確認と作成 **次のステップ: インスタンスの詳細の設定**

- ④ “ステップ 3”では以下の項目を指定の上、右下の[次のステップ: ストレージの追加]をクリックします。（指定が無い項目は、デフォルトのままとしてください）

設定項目	パラメータ
ネットワーク	“IPv6-Web-VPC”の名前がついた VPC ID
サブネット	“IPv6-Web-Subnet-1a” の名前がついた Subnet ID
自動割り当てパブリック IP	有効
IPv6 IP の自動割り当て	有効
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-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

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 3: インスタンスの詳細の設定

要件に合わせてインスタンスを設定します。同じ AMI からの複数インスタンス作成や、より低料金を実現するためのスポットインスタンスのリクエスト、インスタンスへのアクセス管理ロール割り当てなどを行うことができます。

インスタンス数 [Auto Scaling グループに作成する](#)

購入のオプション スポットインスタンスのリクエスト

ネットワーク [新しい VPC の作成](#)

サブネット [新しいサブネットの作成](#)
250 個の IP アドレスが利用可能

自動割り当てパブリック IP

IPv6 IP の自動割り当て

<中略>

▼ 高度な詳細

Enclave 有効

アクセス可能なメタデータ

メタデータのバージョン

メタデータトークンレスポンスのホップ制限

ユーザーデータ テキストで ファイルとして 入力はすでに base64 でエンコード済み

```
#!/bin/bash
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>" >>
```

キャンセル 戻る **確認と作成** **次のステップ: ストレージの追加**

- ⑤ “ステップ 4”ではすべての項目を変更せず、右下の[次のステップ: タグの追加]をクリックします。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 4: ストレージの追加

インスタンスは次のストレージデバイス設定を使用して作成されます。インスタンスに追加の EBS ボリュームやインスタンスストアボリュームをアタッチするか、ルートボリュームの設定を編集することができます。また、インスタンスを作成してから追加の EBS ボリュームをアタッチすることもできますが、インスタンスストアボリュームはアタッチできません。Amazon EC2 のストレージオプションに関する [詳細](#) はこちらをご覧ください。

ボリュームタイプ	デバイス	スナップショット	サイズ (GiB)	ボリュームタイプ	IOPS	スループット (MB/秒)	終了時に削除	暗号化
ルート	/dev/xvda	snap-046af4cb76d7d6635	8	汎用 SSD (gp2)	100 / 3000	該当なし	<input checked="" type="checkbox"/>	暗号化 <input type="checkbox"/>

新しいボリュームの追加

無料利用枠の対象であるお客様は 30 GB までの EBS 汎用 (SSD) ストレージまたはマグネティックストレージを取得できます。無料利用枠の対象と使用制限に関する [詳細](#) はこちら。

キャンセル 戻る 確認と作成 **次のステップ: タグの追加**

- ⑥ “ステップ 5”では[タグの追加]をクリックし、以下の内容を入力の上、右下の[次のステップ: セキュリティグループの設定]をクリックします。

設定項目	パラメータ
キー	Name
値	IPv6-Web-Server-1a

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 5: タグの追加

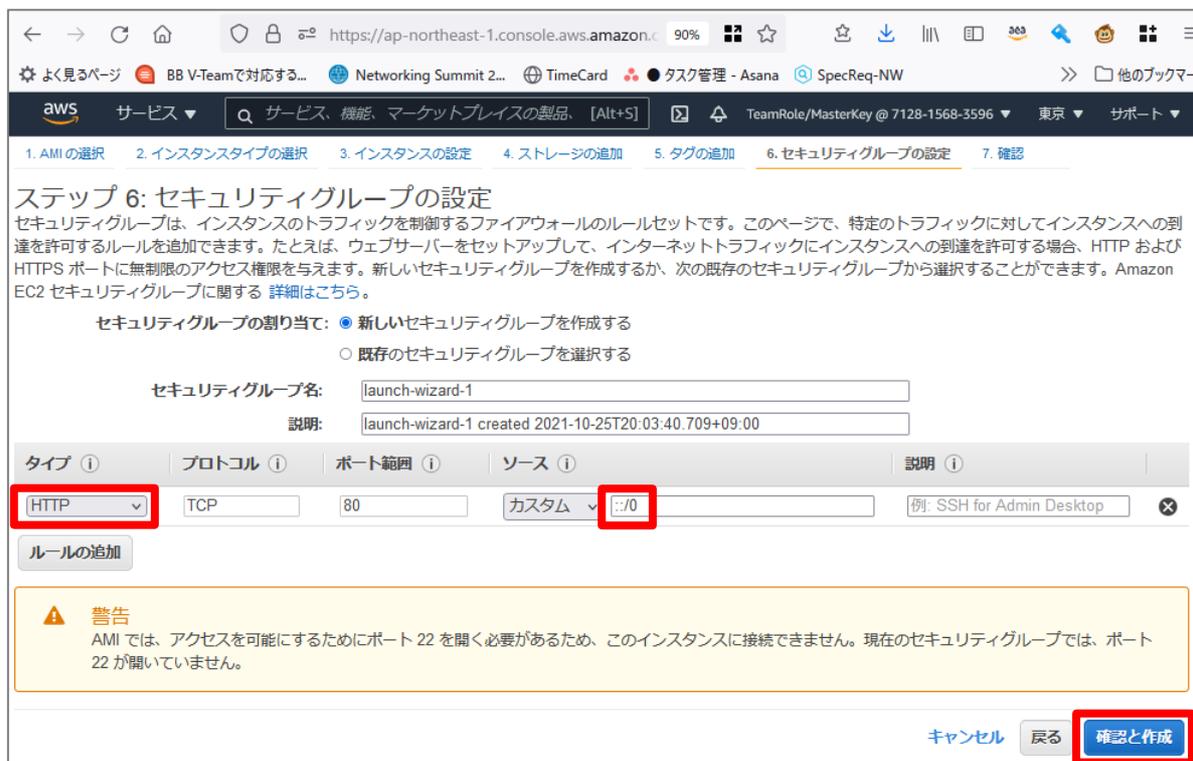
タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、キーに「Name」、値に「Webserver」を使用してタグを定義することができます。タグのコピーは、ボリューム、インスタンス、またはその両方に適用できます。タグは、すべてのインスタンスとボリュームに適用されます。Amazon EC2 リソースのタグ付けに関する [詳細](#) はこちら。

キー (最大 128 文字)	値 (最大 256 文字)	インスタンス	ボリューム	ネットワークインターフェイス
Name	IPv6-Web-Server-1a	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

別のタグを追加 (最大 50 個のタグ)

キャンセル 戻る 確認と作成 **次のステップ: セキュリティグループの設定**

- ⑦ “ステップ 6”ではタイプで“SSH”が指定されているプルダウンメニューから“HTTP”を選択し、“ソース”は“0.0.0.0,”を削除し、IPv6 すべてを意味する“::/0”のみを残します。画面右下の[確認と作成]をクリックします。(本番環境では、メンテナンス用の SSH 等のプロトコルを設定しますが、本ハンズオンでは HTTP による疎通確認のみを行うため、SSH の設定を上書きしていません。)



The screenshot shows the AWS Management Console interface for configuring a security group. The page title is "ステップ 6: セキュリティグループの設定". Below the title, there is a description of security groups and a section for "セキュリティグループの割り当て" (Security Group Assignment) with radio buttons for "新しいセキュリティグループを作成する" (Create new security group) and "既存のセキュリティグループを選択する" (Select existing security group). The "新しいセキュリティグループを作成する" option is selected. Below this, there are input fields for "セキュリティグループ名" (Security Group Name) and "説明" (Description). The "セキュリティグループ名" field contains "launch-wizard-1" and the "説明" field contains "launch-wizard-1 created 2021-10-25T20:03:40.709+09:00".

The main configuration table has the following columns: "タイプ" (Type), "プロトコル" (Protocol), "ポート範囲" (Port Range), "ソース" (Source), and "説明" (Description). The "タイプ" dropdown is set to "HTTP", "プロトコル" is "TCP", "ポート範囲" is "80", "ソース" is "カスタム" (Custom) with a dropdown arrow, and the source IP range is "::/0". The "説明" field contains "例: SSH for Admin Desktop".

Below the table, there is a "ルールの追加" (Add rule) button. A warning message is displayed in a yellow box: "警告 AMI では、アクセスを可能にするためにポート 22 を開く必要があるため、このインスタンスに接続できません。現在のセキュリティグループでは、ポート 22 が開いていません。"

At the bottom right, there are three buttons: "キャンセル" (Cancel), "戻る" (Back), and "確認と作成" (Confirm and Create), which is highlighted with a red box.

- ⑧ “ステップ 7”ではこれまでの設定内容を確認し、右下の[起動]をクリックします。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更に戻ることができます。[作成] をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

▼ AMI の詳細 AMI の編集

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0701e21c502689c31

無料利用枠の対象
Amazon Linux 2 には 5 年間のサポートが含まれます。Amazon EC2、systemd 219、GCC 7.3、Glibc 2.26、Binutils 2.29.1 で最適なパフォーマンスを発揮できるように調整された Linux カーネル 4.14、および、追加の最新のソフトウェアパッケージを提供します。
ルートデバイスタイプ: ebs 仮想化タイプ: hvm

▼ インスタンスタイプ インスタンスタイプの編集

インスタンスタイプ	ECU	vCPU	メモリ (GiB)	インスタンス ストレージ (GB)	EBS 最適化利用	ネットワークパフォーマンス
t2.micro	-	1	1	EBS のみ	-	Low to Moderate

▼ セキュリティグループ セキュリティグループの編集

セキュリティグループ名: launch-wizard-1
説明: launch-wizard-1 created 2021-10-25T20:03:40.709+09:00

タイプ	プロトコル	ポート範囲	ソース	説明
HTTP	TCP	80	::/0	

キャンセル 戻る **起動**

- ⑨ 表示されたダイアログでは、“キーペアなしで続行”を選択します。下部の「私は、キーペアが無い場合、・・・」のチェックボックスにチェックし、画面右下の[インスタンスの作成]をクリックします。

既存のキーペアを選択するか、新しいキーペアを作成します。

キーペアは、AWS が保存するパブリックキーとユーザーが保存するプライベートキーファイルで構成されます。組み合わせることで、インスタンスに安全に接続できます。Windows AMI の場合、プライベートキーファイルは、インスタンスへのログインに使用されるパスワードを取得するために必要です。Linux AMI の場合、プライベートキーファイルを使用してインスタンスに SSH で安全に接続できます。Amazon EC2 は ED25519 および RSA キーペアタイプをサポートしています。

注: 選択したキーペアは、このインスタンスに対して権限がある一連のキーに追加されます。「パブリック AMI から既存のキーペアを削除する」の詳細情報をご覧ください。

キーペアなしで続行

私は、キーペアがない場合、EC2 Instance Connect を使用するか、AMI に組み込まれているパスワードを知っている場合にのみ、このインスタンスに接続できることを認識しています。EC2 Instance Connect は、Amazon Linux 2 および Ubuntu でのみサポートされています。詳細はこちらをご覧ください。

キャンセル **インスタンスの作成**

- 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。

作成ステータス

✔ インスタンスは現在作成中です
次のインスタンスの作成が開始されました: i-07e83e159e8496086 [作成ログの表示](#)

ℹ 予想請求額の通知を受け取る
請求アラートの作成 AWS 請求書の予想請求額が設定した金額を超えた場合 (つまり、無料利用枠を超えた場合)、メール通知を受け取ります。

インスタンスへの接続方法

インスタンスは作成中です。実行中状態になり、使用する準備ができるまでに数分かかることがあります。新しいインスタンスの使用時間は、すぐに始まり、インスタンスを停止または終了するまで続きます。

[インスタンスの表示] をクリックして、インスタンスのステータスを監視します。インスタンスが一度実行中状態になれば、[インスタンス] 画面からインスタンスに接続できます。インスタンスへの接続方法を [詳細はこちら](#)。

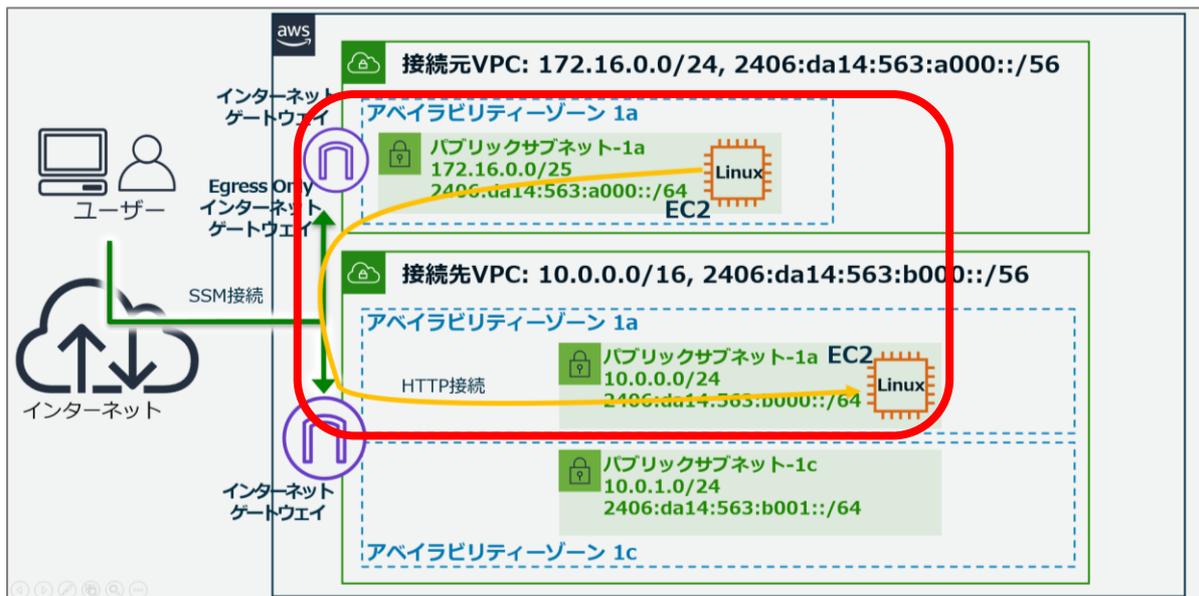
▼ ここには、作業を始めるのに役立つリソースがあります

- [Linux インスタンスへの接続方法](#)
- [Amazon EC2: ユーザーガイド](#)
- [AWS 無料利用枠の詳細](#)
- [Amazon EC2: ディスカッションフォーラム](#)

インスタンスの作成中、次のことも行うことができます

4.4. Web サーバへの接続テスト

前の章で起動した接続元 VPC 環境の Linux OS から、起動した Web サーバへ接続テストを行います。



- ① 起動したインスタンス名の左にあるラジオボタンにチェックを入れます。画面下に表示される詳細項目から、“IPv6 アドレス”の左にあるコピーボタンをクリックします。

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with 'インスタンス (1/2) 情報', '接続', 'インスタンスの状態', 'アクション', and 'インスタンスを起動'. Below this is a search bar and a table of instances. The table has columns for Name, インスタンス ID, インスタンスの状態, インスタンスタイプ, ステータスチェック, アラームの状態, and アベイラビリティゾーン. The instance 'IPv6-Web-Server-1a' is selected, and its details are shown in a modal window below. The modal window has tabs for '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', and 'タグ'. The '詳細' tab is active, showing 'インスタンス概要 情報'. The 'IPv6 アドレス' field is highlighted with a red box, and a tooltip indicates 'IPv6 アドレスがコピーされました'.

Name	インスタンス ID	インスタンスの状態	イン...	イン...	ステータスチェ...	アラームの状態	アベイラビリティ
bastion linux server	i-0fb3c9fda0b076c60	実行中	t2.micro	2/2 のチェックに	アラーム...	+	ap-northeast-1
IPv6-Web-Server-1a	i-0e9bf3336867c92f0	実行中	t2.micro	2/2 のチェックに	アラーム...	+	ap-northeast-1

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	プライベート IPv4 アドレス
i-0e9bf3336867c92f0 (IPv6-Web-Server-1a)	-	10.0.0.148
IPv6 アドレス	インスタンスの状態	パブリック IPv4 DNS
2001:0000:0000:0000:0000:0000:0000:0000	実行中	-
プライベート IP アドレス	インスタンスタイプ	Elastic IP アドレス
ip-10-0-0-148.ap-northeast-1.compute.internal	t2.micro	-

- ② 前の章で作成した、VPC 上で起動した Linux OS の SSM 画面に戻り、curl コマンドを利用し、コピーした IPv6 アドレスへアクセスします。（SSM の画面がタイムアウトしている場合、一度、ブラウザのタブをクローズ後、再度接続してください。）SSM コンソールでは、右クリックから“貼り付け”を選択可能です。curl コマンドで IPv6 を記載する場合、アドレスを角カッ

コロン“:”で囲みます。コマンド実行結果に赤字で示す Nginx のテストページを示す記載があれば、アクセス成功です。

ここで利用した IPv6 アドレスは、後の作業で利用するため、テキストエディタなどにメモしてください。

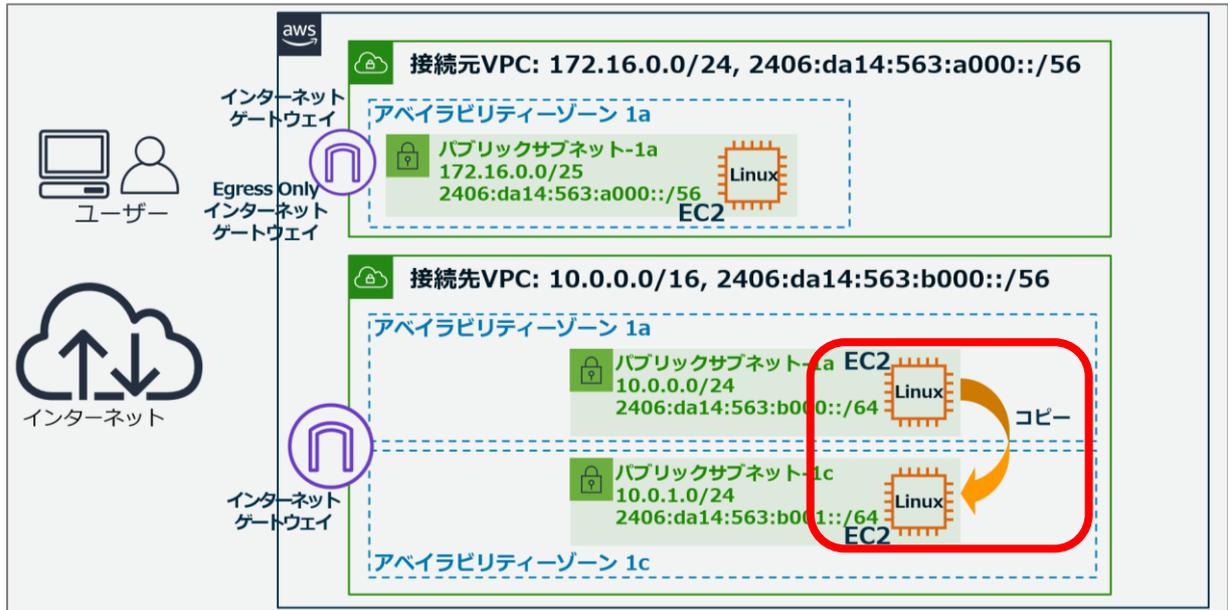
curl -v6 http://[2406:da14:a42:5400:152c:2857:82e:e01b]/ | head

```
sh-4.2$ curl -v6 http://[2406:da14:d55:e500:5e10:dfa5:883e:e43] | head
* Trying 2406:da14:d55:e500:5e10:dfa5:883e:e43:80...
<<中略>>
> User-Agent: curl/7.76.1
> Accept: */*
<<中略>>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<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">
      /**]
        body {
          background-color: #fff;</pre></div><div data-bbox="191 519 794 537" data-label="Section-Header"><h3>参考 : IPv6 アドレスの場合、なぜ角カッコで囲む必要があるの？</h3></div><div data-bbox="210 560 880 667" data-label="Text"><p>URL の書式では、単一のコロン“:”を「IP アドレスとポート番号の区切り」として指定されています。このため、IPv6 アドレスで 16 ビットの区切り文字として利用しているコロン“:”と混乱してしまいます。この状況を回避するため、IPv6 アドレスを扱う際には、ホスト部分を明確にするために、角カッコ“[~]”で囲んで入力する必要があります。</p></div><div data-bbox="693 907 783 948" data-label="Page-Footer"><img alt="aws logo" data-bbox="693 907 783 948"/></div><div data-bbox="804 912 837 928" data-label="Page-Footer"><p>40</p></div>
```

4.5. 冗長化用 Web サーバ作成

先ほど起動した Web サーバをコピーし、冗長化用の Web サーバをもう一つのサブネットに起動します。



- ① EC2 ダッシュボードにて“IPv6-Web-Server-1a”が選択された状態で、右上の[アクション] > [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。



- ② “ステップ 7”からウィザードが開始されます。画面上部の[3. インスタンスの設定]をクリックします。

1. AMIの選択 2. インスタンスタイプの選択 **3. インスタンスの設定** 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更に戻ることができます。[作成]をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

▼ AMIの詳細 [AMIの編集](#)

amzn2-ami-hvm-2.0.20210219.0-x86_64-gp2 - ami-09d28faae2e9e7138
Amazon Linux 2 AMI 2.0.20210219.0 x86_64 HVM gp2
ルートデバイスタイプ: ebs 仮想化タイプ: hvm

▼ インスタンスタイプ [インスタンスタイプの編集](#)

インスタンスタイプ	ECU	vCPU	メモリ (GiB)	インスタンスストレージ (GB)	EBS 最適化利用	ネットワークパフォーマンス
t2.micro	-	1	1	EBSのみ	-	Low to Moderate

▼ セキュリティグループ [セキュリティグループの編集](#)

- ③ “ステップ 3”の画面で、サブネットなどの設定を変更します。その他の項目は、表示されたままとします。画面上部の[5. タグの追加]をクリックします

設定項目	パラメータ
サブネット	“IPv6-Web-Subnet-1c” の名前がついた Subnet ID
自動割り当てパブリック IP	有効
IPv6 IP の自動割り当て	有効

ステップ 3: インスタンスの詳細の設定

購入のオプション スポットインスタンスのリクエスト

ネットワーク

サブネット

自動割り当てパブリック IP

IPv6 IP の自動割り当て

- ④ 予め Name タグが設定されているため、値を“IPv6-Web-Server-1c”に変更します。続けて、画面下の[確認と作成]をクリックします。

ステップ 5: タグの追加

タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、キーに「Name」、値に「Webserver」を使用してタグを定義することができます。

タグのコピーは、ボリューム、インスタンス、またはその両方に適用できます。

タグは、すべてのインスタンスとボリュームに適用されます。Amazon EC2 リソースのタグ付けに関する [詳細はこちら](#)。

キー (最大 128 文字)	値 (最大 256 文字)	インスタンス <input type="checkbox"/>	ボリューム <input type="checkbox"/>	ネットワークインターフェイス <input type="checkbox"/>
Name	IPv6-Web-Server-1c	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

別のタグを追加 (最大 50 個のタグ)

キャンセル 戻る **確認と作成** 次のステップ: セキュリティグループの設定

- ⑤ “ステップ 7”の画面にて右下の[起動]をクリックします。



1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更に戻ることができます。[作成] をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

▼ AMI の詳細 AMI の編集

 **amzn2-ami-hvm-2.0.20211001.1-x86_64-gp2 - ami-0701e21c502689c31**
Amazon Linux 2 AMI 2.0.20211001.1 x86_64 HVM gp2
ルートデバイスタイプ: ebs 仮想化タイプ: hvm

▼ インスタンスタイプ インスタンスタイプの編集

インスタンスタイプ	ECU	vCPU	メモリ (GiB)	インスタンスストレージ (GB)	EBS 最適化利用	ネットワークパフォーマンス
t2.micro	-	1	1	EBS のみ	-	Low to Moderate

キャンセル 戻る **起動**

- ⑥ 確認ダイアログでは、“キーペアなしで続行”を選択し、下部のラジオボタンにチェックを入れ、右下の[インスタンスの作成]をクリックします。



既存のキーペアを選択するか、新しいキーペアを作成します。

キーペアは、AWS が保存するパブリックキーとユーザーが保存するプライベートキーファイルで構成されます。組み合わせて使用することで、インスタンスに安全に接続できます。Windows AMI の場合、プライベートキーファイルは、インスタンスへのログインに使用されるパスワードを取得するために必要です。Linux AMI の場合、プライベートキーファイルを使用してインスタンスに SSH で安全に接続できます。Amazon EC2 は ED25519 および RSA キーペアタイプをサポートしています。

注: 選択したキーペアは、このインスタンスに対して権限がある一連のキーに追加されます。「パブリック AMI から既存のキーペアを削除する」の詳細情報をご覧ください。

キーペアなしで続行

私は、キーペアがない場合、EC2 Instance Connect を使用するか、AMI に組み込まれているパスワードを知っている場合のみ、このインスタンスに接続できることを認識しています。EC2 Instance Connect は、Amazon Linux 2 および Ubuntu でのみサポートされています。詳細はこちらをご覧ください。

キャンセル **インスタンスの作成**

- ⑦ 以下の画面に推移し、正常に EC2 インスタンスが作成中であることが確認できます。右下の[インスタンスの表示]をクリックしてください。

作成ステータス

✔ **インスタンスは現在作成中です**
 次のインスタンスの作成が開始されました: i-0a611f77e4f888bed [作成ログの表示](#)

i **予想請求額の通知を受け取る**
 請求アラートの作成 AWS 請求書の予想請求額が設定した金額を超えた場合 (つまり、無料利用枠を超えた場合)、メール通知を受け取りません。

インスタンスへの接続方法

インスタンスは作成中です。実行中状態になり、使用する準備ができるまでに数分かかることがあります。新しいインスタンスの使用時間は、すぐに始まり、インスタンスを停止または終了するまで継続します。

[インスタンスの表示] をクリックして、インスタンスのステータスを監視します。インスタンスが一度実行中状態になれば、[インスタンス] 画面からインスタンスに接続できます。インスタンスへの接続方法を [詳細はこちら](#)。

▼ ここには、作業を始めるのに役立つリソースがあります

- [Linux インスタンスへの接続方法](#)
- [Amazon EC2: ユーザーガイド](#)
- [AWS 無料利用枠の詳細](#)
- [Amazon EC2: ディスカッションフォーラム](#)

インスタンスの作成中、次のことも行うことができます

- [ステータスチェックアラームの作成](#) これらのインスタンスがステータスチェックに合格しなかったときは通知が送信されます。(追加料金が適用される場合があります)
- [追加の EBS ボリュームを作成してアタッチする](#) (追加料金が適用される場合があります)
- [セキュリティグループの管理](#)

インスタンスの表示

- ⑧ 先ほど作成した Web サーバと同等の EC2 インスタンスが、アベイラビリティゾーン、ap-northeast-1c に複製されていることを確認します。

インスタンス (3) 情報 接続 インスタンスの状態 ▼ アクション ▼ インスタンスを起動 ▼

🔍 | インスタンスをフィルタリング < 1 > ⚙️

インスタンスの状態: running ✕ フィルターをクリア

<input type="checkbox"/>	Name ▲	インスタンス ID	インスタン... ▼	イン... ▼	ステータスチェ...	アラームの状態	アベイラビリティ...
<input type="checkbox"/>	bastion linux server	i-0fb3c9fda0b076c60	✔️ 実行中 🔍	t2.micro	✔️ 2/2 のチェックに	アラーム... +	ap-northeast-1a
<input type="checkbox"/>	IPv6-Web-Server-1a	i-0d79a94b148afeea1	✔️ 実行中 🔍	t2.micro	✔️ 2/2 のチェックに	アラーム... +	ap-northeast-1a
<input type="checkbox"/>	IPv6-Web-Server-1c	i-014fbb0ff186f563f	✔️ 実行中 🔍	t2.micro	🕒 初期化しています	アラーム... +	ap-northeast-1c

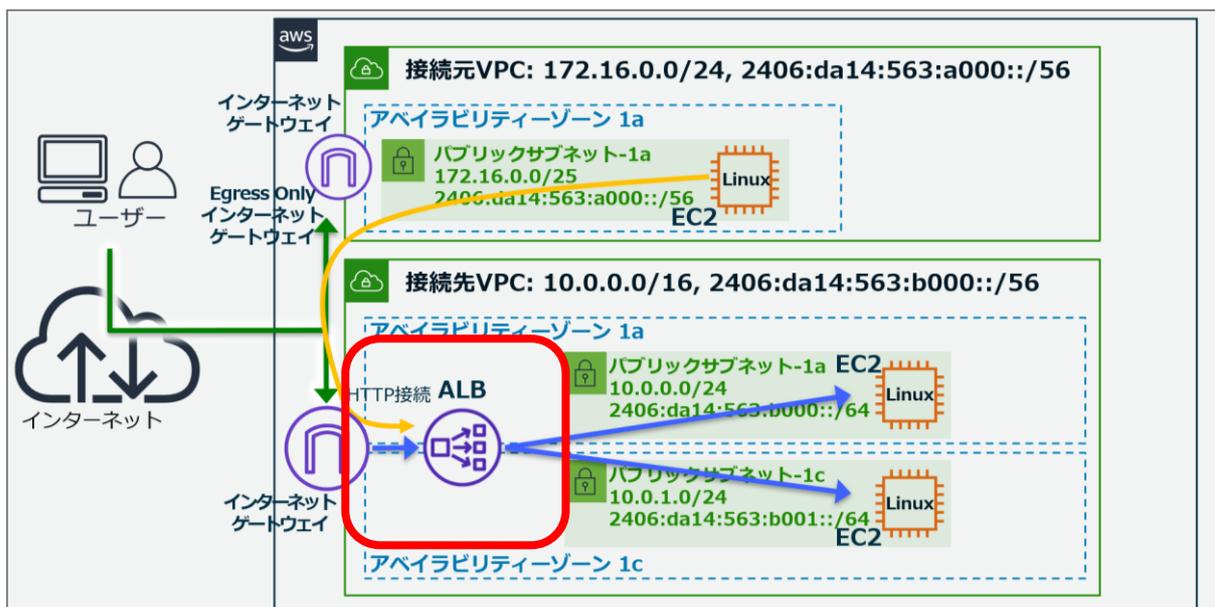
- ⑨ 複製した Web サーバの左にあるチェックボックスを選択し、下部ペインで表示される“IPv6 アドレス”のコピーボタンをクリックします。このアドレスは後の作業で必要となるので、テキストエディタなどにメモしてください。

The screenshot shows the AWS Management Console. At the top, there's a search bar and filters. A table lists instances, with 'IPv6-Web-Server-1c' highlighted in red. Below, the 'インスタンス: i-014fbb0ff186f563f (IPv6-Web-Server-1c)' details are shown. The 'IPv6 アドレス' field is highlighted with a red box, and a tooltip indicates 'IPv6 アドレスがコピーされました' (IPv6 address copied successfully).

Name	インスタンス ID	インスタン...	イン...	ステータスチェ...	アラームの状態	アベイラビリテ
bastion linux server	i-0fb3c9fda0b076c60	実行中	t2.micro	2/2 のチェックに	アラーム...	ap-northeast-1
IPv6-Web-Server-1a	i-0d79a94b148afeea1	実行中	t2.micro	2/2 のチェックに	アラーム...	ap-northeast-1
IPv6-Web-Server-1c	i-014fbb0ff186f563f	実行中	t2.micro	2/2 のチェックに	アラーム...	ap-northeast-1

4.6. 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。



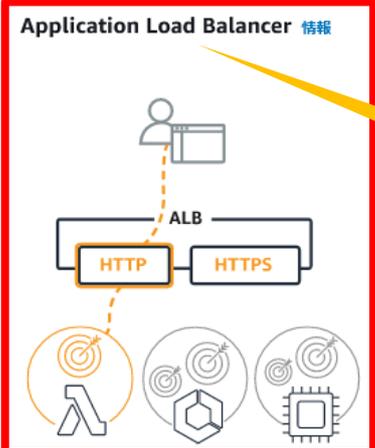
② “Application Load Balancer”を示す図の下にある[作成]をクリックします。

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](#)

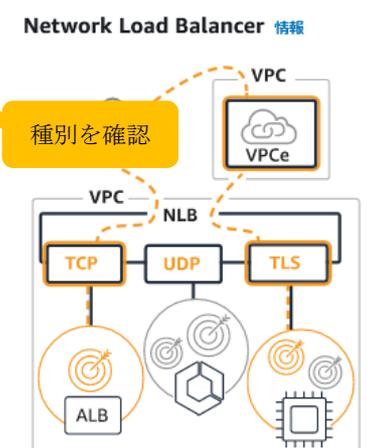
Application Load Balancer 情報



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。

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

種別を確認

クリック

▶ 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
この IP 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-Web-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ます。

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-Web-VPC
vpc-0ad9f362d8a8011ba
IPv4: 10.0.0.0/16 IPv6: 2406:da14:a42:5400::/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。ロードバランサーの作成後にサブネットを削除することはできませんが、追加のサブネットは追加できます。

ap-northeast-1a

サブネット
subnet-021d1982871feaf87 IPv6-Web-Subnet-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:a42:5400::/64 から割り当て済み

ap-northeast-1c

サブネット
subnet-02dcb69b83facf274 IPv6-Web-Subnet-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:a42:5401::/64 から割り当て済み

- ⑤ 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。

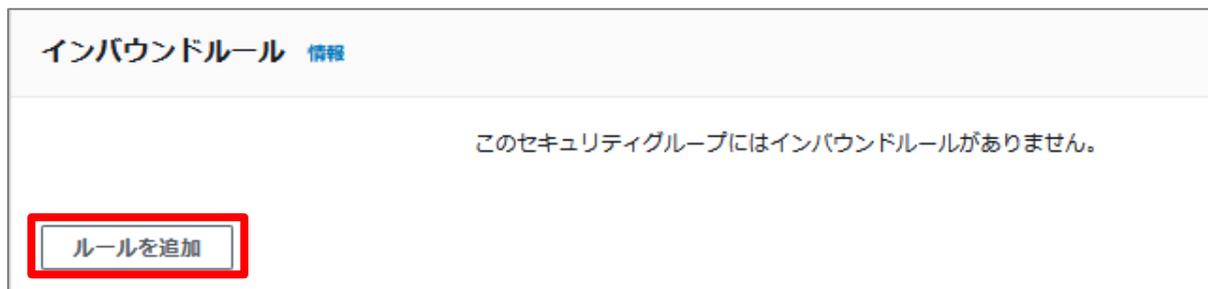


- ⑥ ブラウザで別タブが開き、“セキュリティグループを作成”へ遷移します。“基本的な詳細”では、以下を入力します。

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	AllowHTTP	レ
VPC	“IPv6-Web-VPC”の名前が付いた VPC ID を選択	レ



- ⑦ 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。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。



ルール(1)

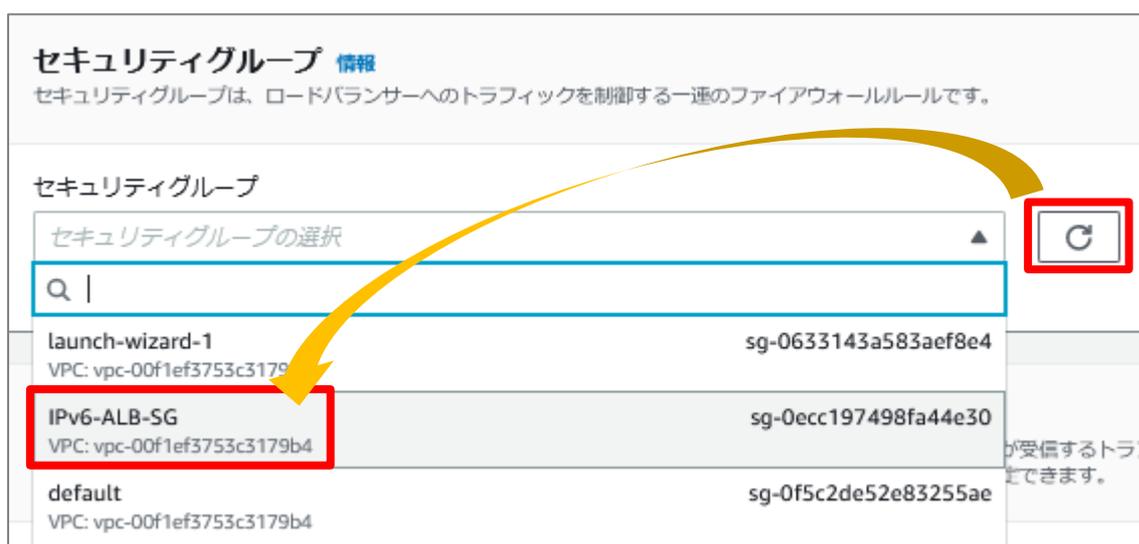
設定項目	パラメータ	要修正
タイプ	HTTP	レ
ソース(左側メニュー)	Anywhere-IPv4	レ
ソース(右側入力欄)	0.0.0.0/0	

ルール(2)

設定項目	パラメータ	要修正
タイプ	HTTP	レ
ソース(左側メニュー)	Anywhere-IPv6	レ
ソース(右側入力欄)	::/0	



- ⑧ 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。



- ⑨ 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、設定したプロトコルとポートを使用して、接続リクエストをチェックするプロセスです。リスナーが受信するトラフィックは、仕様に従ってルーティングされます。ロードバランサーの作成後に、リスナーごとに複数のルールと複数の証明書を指定できます。

▼ リスナー HTTP:80 削除

プロトコル	ポート	デフォルトアクション 情報
HTTP	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](#)

リスナーの追加

- ⑩ 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-Web-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。

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。

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 ページで、この VPC から IP アドレスを登録できます。

IPv6-Web-VPC
vpc-0ad9f362d8a8011ba
IPv4: 10.0.0.0/16 IPv6: 2406:da14:a42:5400::/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。

- ⑪ 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**

- ⑫ “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て行うことができます。

ネットワーク
 IPv6-Web-VPC
 vpc-0ad9f362d8a8011ba
 IPv6: 2406:da14:a42:5400::/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)

Q 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:a42:5401:2d66:8bd5:a948:1f7b	i-014fbb0ff186f563f	IPv6-Web-Server-1c	running	launch-wizard-1	ap-northeast-1c	subnet-02dcb69b83fac274
<input checked="" type="checkbox"/>	2406:da14:a42:5400:d239:532a:e433:696	i-0d79a94b148afeea1	IPv6-Web-Server-1a	running	launch-wizard-1	ap-northeast-1a	subnet-021d1982871feaf87

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。

 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

- ⑬ 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:a42:5400:d239:532a:e433:696	80	ap-northeast-1a
X	保留中	2406:da14:a42:5401:2d66:8bd5:a948:1f7b	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**

- ⑭ 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**

Q ターゲットグループを検索またはフィルタリング < 1 > ⚙

<input type="checkbox"/>	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
<input type="checkbox"/>	ipv6-handson-tg	arn:aws:elasticloadbalancing:a...	80	HTTP	IP	-

- ⑮ “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、設定したプロトコルとポートを使用して、接続リクエストをチェックするプロセスです。リスナーが受信するトラフィックは、仕様によってルーティングされます。ロードバランサーの作成後に、リスナーごとに複数のルールと複数の証明書を指定できます。

▼ 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 更新

ターゲット: 検索

ipv6-handson-tg (ターゲットの種類: IP, IPv6) HTTP

リスナーの追加

- ⑯ 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-alb
• インターネット向け
• Dualstack

セキュリティグループ 編集
• launch-wizard-1
sg-07b1b3317e1b2371c

ネットワークマッピング 編集
VPC vpc-0ad9f362d8a8011ba
IPv6-Web-VPC
• ap-northeast-1a
subnet-021d1982871feaf87
IPv6-Web-Subnet-1a
• ap-northeast-1c
subnet-02dcb69b83facf274
IPv6-Web-Subnet-1c

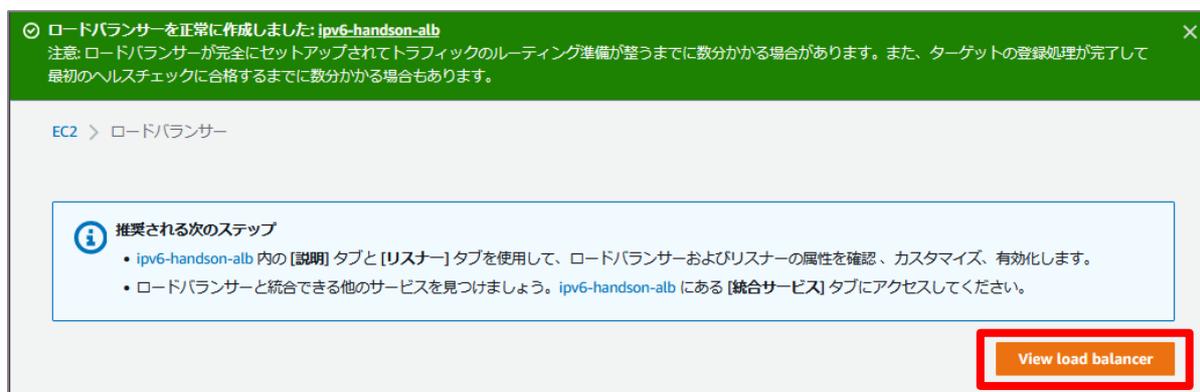
リスナーとルーティング 編集
• HTTP:80 次をデフォルトにする:
ipv6-handson-tg

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。

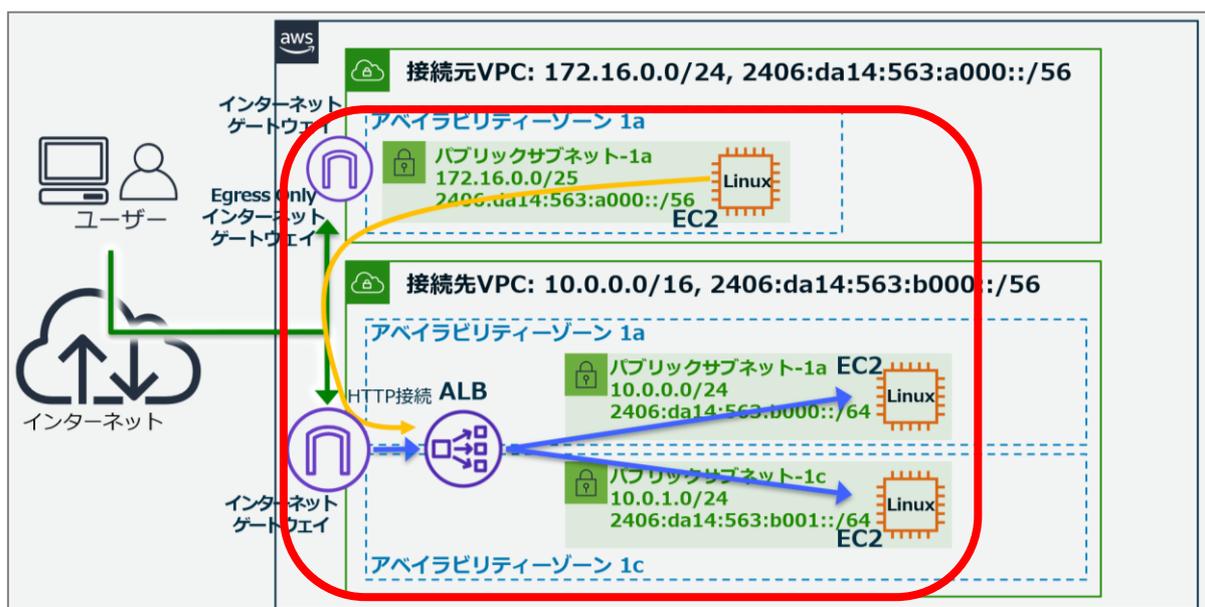
キャンセル **ロードバランサーの作成**

- ⑰ 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。



4.7. ALB への接続確認

接続元 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。画面の説明タブ内に記載の“DNS 名”の右にあるコピーボタンをクリックし、クリップボードに FQDN をコピーします。

なお、この DNS 名“ipv6-handson-alb-xxxxxx.elb.amazonaws.com”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。

The screenshot shows the AWS Management Console interface for an Elastic Load Balancing (ALB) instance. The instance name is 'ipv6-handson-alb'. The DNS name is 'ipv6-handson-alb-365479048.ap-northeast-1.elb.amazonaws.com', which is highlighted with a red box. The ARN is 'arn:aws:elasticloadbalancing:ap-northeast-1:725526135330:loadbalancer/app/ipv6-handson-alb/47fd2a5f7ed97e21'. The status is 'active' and the VPC ID is 'vpc-0916c8b86d6c'.

名前	DNS 名	状態	VPC ID
ipv6-handson-alb	ipv6-handson-alb-365479048...	active	vpc-0916c8b86d6c

ロードバランサー: ipv6-handson-alb

説明 リスナー モニタリング 統合サービス タグ

基本的な設定

名前	ipv6-handson-alb
ARN	arn:aws:elasticloadbalancing:ap-northeast-1:725526135330:loadbalancer/app/ipv6-handson-alb/47fd2a5f7ed97e21
DNS 名	ipv6-handson-alb-365479048.ap-northeast-1.elb.amazonaws.com (A レコードまたは AAAA レコード)

- ② 接続元 VPC 内の Linux OS へ SSM 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-781766639.ap-northeast-1.elb.amazonaws.com/ | head
```

```
sh-4.2$ curl -v6 http://ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com/ |head
* Connected to ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com
(2406:da14:d55:e500:31c3:b5ee:9c2e:32e9) port 80 (#0)
> GET / HTTP/1.1
> Host: ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com
> User-Agent: curl/7.76.1
> 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;
sh-4.2$
```

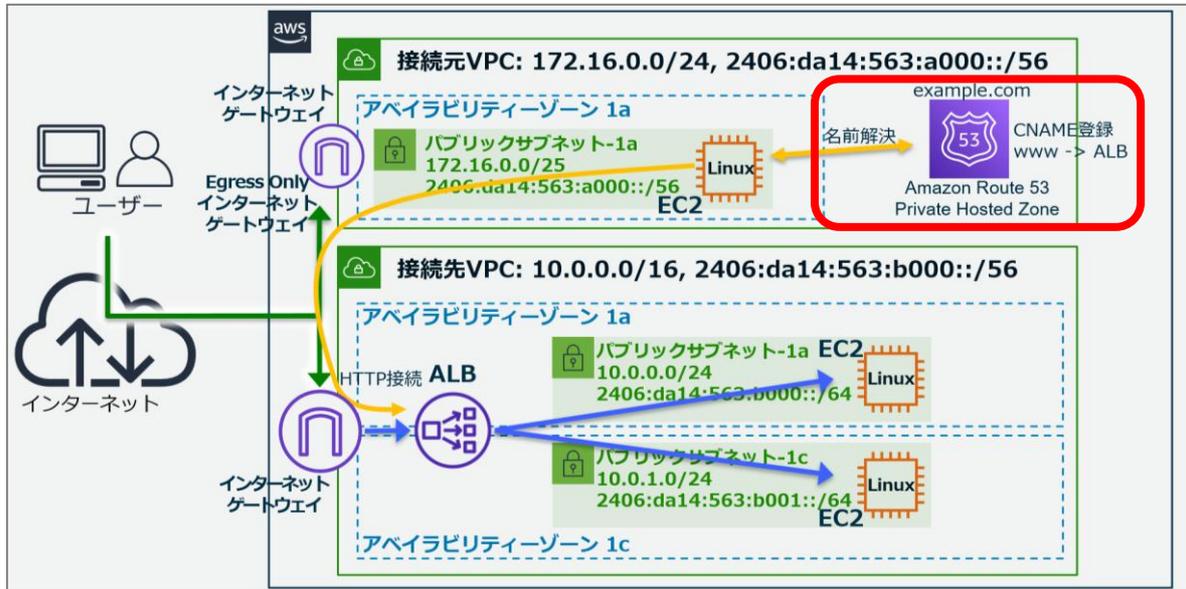
- ③ 前の手順では ALB の DNS 名を利用して、Web サーバへ接続できることを確認できました。アクセス元も IPv6 を使っていることを確認するため、アクセス元 IP アドレスを表示可能な URL へアクセスします。先ほどの手順で利用した URL の最後に[/index2.php]を加えて再度アクセスします。以下のように接続元 Linux OS にアサインされている IPv6 アドレスが表示されていれば、ALB に対し IPv6 を使ってアクセスしている状態になります。

```
curl -v6 http://ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com/index2.php
```

```
sh-4.2$ curl -v6 http://ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com/index2.php
* Trying 2406:da14:d55:e501:7f52:ac8e:ec78:b083:80...
* Connected to ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com
(2406:da14:d55:e501:7f52:ac8e:ec78:b083) port 80 (#0)
<<省略>>
<html><h1>HTTP_X_FORWARDED_FOR: 2406:da14:12b:4e00:9189:7f29:d689:5f79</html></h1>
* Connection #0 to host ipv6-handson-alb-1996582727.ap-northeast-1.elb.amazonaws.com left intact
sh-4.2$
```

5. DNS で IPv6 を FQDN に登録する

この章では、作成した ALB に対し、接続元 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接続元 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

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-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
この値で、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が示します。

- 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 などのプログラム的な方法を用いる必要があります。

① 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。

リージョン 情報 アジアパシフィック (東京) [ap-nor... ▼

VPC ID 情報 vpc-089949a378d5ae6c4 X

VPC を追加

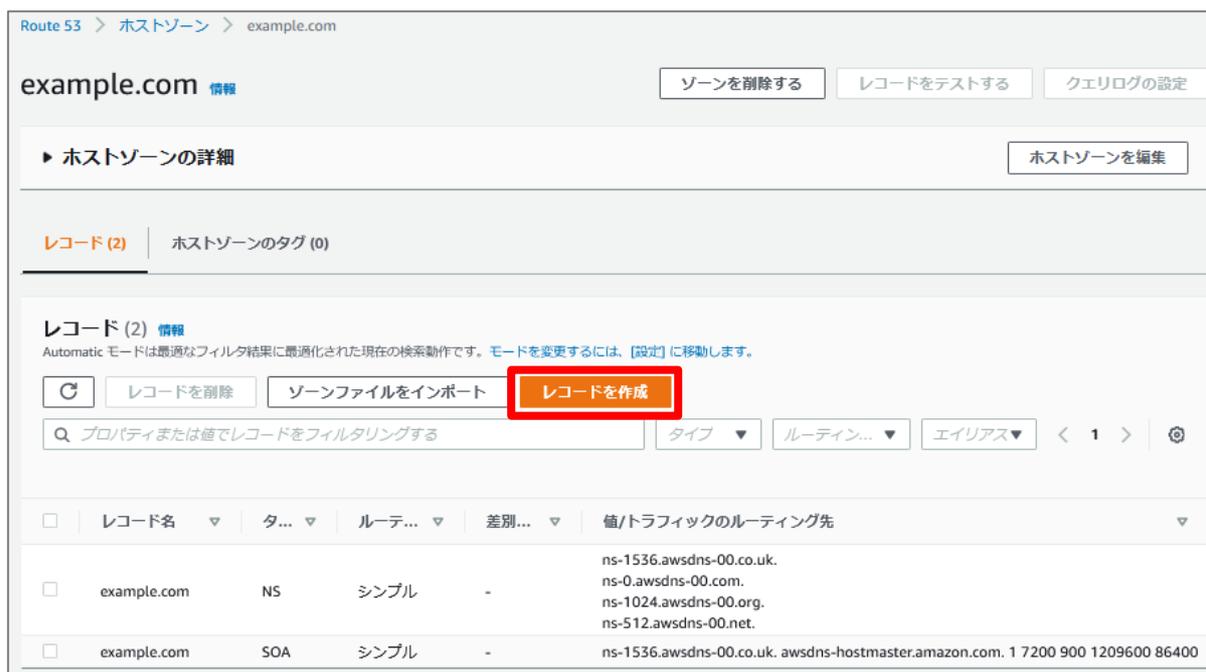
VPC を削除

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。



The screenshot shows the Amazon Route 53 console interface for a host zone named 'example.com'. The 'Records' tab is active, showing a list of records. The 'Create Record' button is highlighted with a red box. The table below shows the existing records:

レコード名	タイプ	ルーティング	優先度	値/トラフィックのルーティング先
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.
example.com	SOA	シンプル	-	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

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。

- 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-xxxxxxxx.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

レコードのクイック作成 情報 ウィザードに切り替える 別のレコードを追加

▼ 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 値 情報 エイリアス

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

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

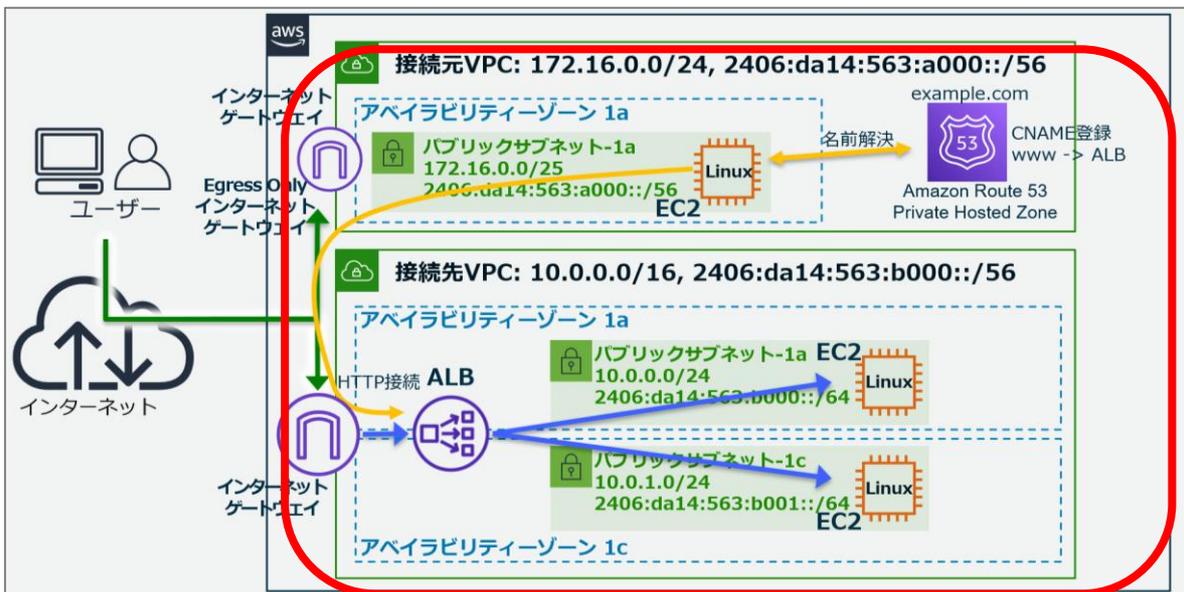
推奨値: 60~172800 (2 日間)

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。



5.2. Linux OS から名前解決の確認

接続元 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

```
sh-4.2$ dig www.example.com. +short
ip6v6-handson-alb-781766639.ap-northeast-1.elb.amazonaws.com.
52.68.116.48
18.179.98.174
```

- 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”をつけ、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

```
sh-4.2$ dig www.example.com. aaaa +short
ip6v6-handson-alb-781766639.ap-northeast-1.elb.amazonaws.com.
2406:da14:a42:5401:79bb:4fde:35a3:148e
2406:da14:a42:5400:78a1:f881:aaa4:e9d9
sh-4.2$
```

- 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

```
sh-4.2$ curl -v6 http://www.example.com/ | head
<<省略>>
* Connected to www.example.com (2406:da14:d55:e500:31c3:b5ee:9c2e:32e9) port 80 (#0)
> GET / HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.76.1
<<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;
```

- 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

```
sh-4.2$ curl -v6 http://www.example.com/index2.php
* Trying 2406:da14:d55:e501:7f52:ac8e:ec78:b083:80...
* Connected to www.example.com (2406:da14:d55:e501:7f52:ac8e:ec78:b083) port 80 (#0)
<<省略>>
<html><h1>HTTP_X_FORWARDED_FOR: 2406:da14:12b:4e00:9189:7f29:d689:5f79</html></h1>
* Connection #0 to host www.example.com left intact
sh-4.2$
```

以上、本ハンズオンの設定作業はすべて終了です。

6. 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。

6.1. 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)
- ALB 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)
- 接続元 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)

7. ドキュメント情報

7.1. 更新履歴

Ver #	Date	Author	Revision Description
0.1	2021.10.25	Nobuaki Kikuchi	初版
1.0	2021.11.17	Nobuaki Kikuchi	リリース予定