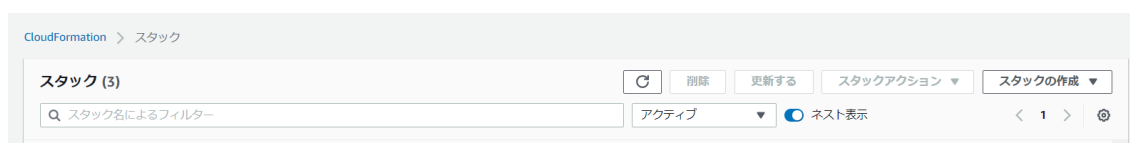
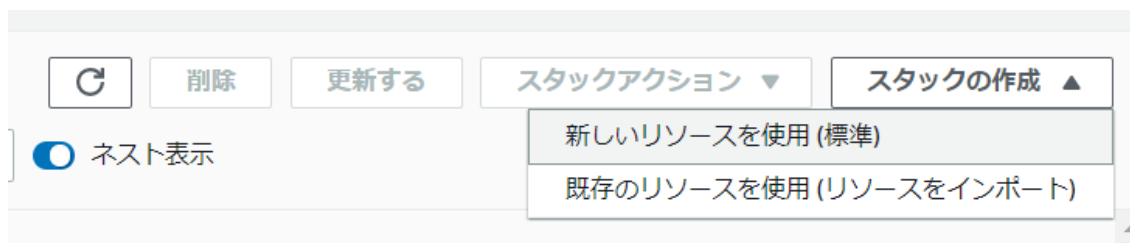


1. CloudFormation による環境の構築

- 1.1. 作業はすべてバージニア北部で行います。CloudFormation のマネジメントコンソール上で「スタックの作成」ボタンをおします



- 1.2. 「新しいリソースを仕様（標準）を押します」



- 1.3. 以下の URL からファイルをダウンロードしどこか適当なところに保存します

<https://github.com/harunobukameda/VPC-Reachability-Analyzer/blob/main/vpcreachabilityanalyzertestenvironment.yaml>

- 1.4. 「テンプレートファイルのアップロード」から上記でダウンロードしたテンプレートをアップロードし、「次へ」ボタンをおします。中身に興味がある方は、「次へ」を押す前に「デザイナーで表示」ボタンをおして今から作成しようとしている環境のダイアグラムを見てみてください。

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、スタックに含む 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 

ファイルが選択されていません

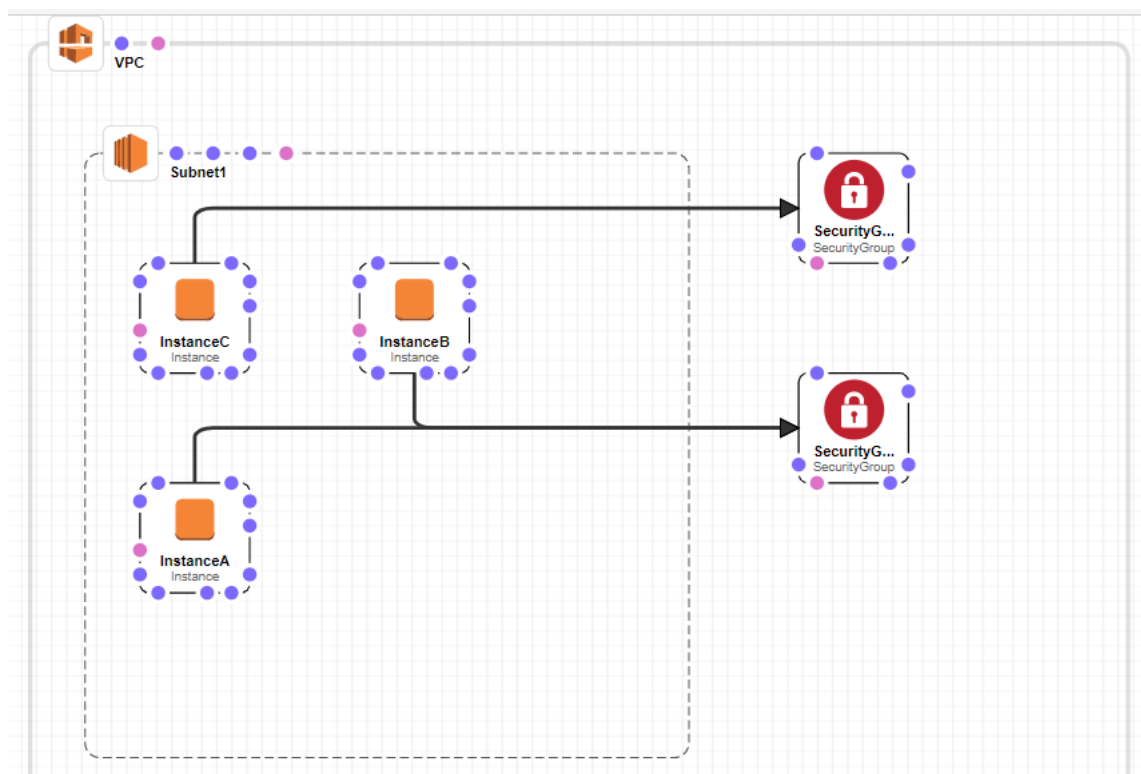
JSON または YAML 形式のファイル

S3 URL: テンプレートファイルをアップロードすると生成されます。

デザイナーで表示

キャンセル

次へ



この環境では、1つの VPC、3つの EC2 インスタンス、2つのセキュリティグループを作ります。インスタンス A と B は相互に通信できますが、インスタンス C にアタッチされたセキュリティグループでは着信トラフィックが許可されないため、それら 2 つのインスタンスはインスタンス C と通信できません。

- 1.5. スタックの適当な名前を付けて「次へ」を押します。次の画面は全てデフォルトのまま「次へ」を再度押します。

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。

パラメータなし

テンプレートで定義されているパラメータはありません

キャンセル 戻る 次へ

- 1.6. 最後の確認画面で「スタックの作成」を押します。以下の様に構築中の画面になりますので少し待ちます。

vpctest 削除 更新する スタックアクション ▼ スタックの作成 ▼

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 変更セット

イベント (1)

🔄⚙️

タイムスタンプ	論理 ID	ステータス	状況の理由
2021-03-12 14:41:12 UTC+0900	vpctest	🔄 CREATE_IN_PROGRESS	User Initiated

- 1.7. 以下のようになれば構築が完了です。

CloudFormation > スタック > vpctest

📦 スタック (12)

🔄

アクティブ ▼ 🔘 ネスト表示

< 1 >

vpctest

2021-03-12 14:41:12 UTC+0900

✅ CREATE_COMPLETE

🔘

CloudFormation のリソースタブで出力された VPCID、3 つのインスタンス ID を控えておきます。

- 1.8. EC2 マネージメントコンソールで、それぞれ[A][B][C]という名前を生成された 3 つのインスタンスに指定します。(A,B と C は技術統制が異なるので、必ず上記手順と同じ指定となるようにしてください)

インスタンス (9) 情報							
Q インスタンスをフィルタリング							
<input type="checkbox"/>	Name ▼	インスタンス ID	インスタンス... ▼	インスタン... ▼	ステータスチエ...	アラームの...	アベイラビリテ...
<input type="checkbox"/>	C	i-02812a82eb9097605	✔ 実行中	t3.nano	✔ 2/2 のチェックに	アラーム... +	us-east-1c
<input type="checkbox"/>	B	i-0541964ae1f73950b	✔ 実行中	t3.nano	✔ 2/2 のチェックに	アラーム... +	us-east-1c
<input type="checkbox"/>	A	i-0aa6245c147fc416e	✔ 実行中	t3.nano	✔ 2/2 のチェックに	アラーム... +	us-east-1c

2. VPC Reachability Analyzer の確認

- 2.1. VPC マネージメントコンソールの左ペインで、[Reachability Analyzer]をクリックします



- 2.2. 「パスの作成と分析」をおします



- 2.3. 適当な名前を入力し、送信元タイプで「Instances」、送信元で「A」、を選択しま

す。同様に 2 番目の送信元タイプで「Instances」、送信先で「B」、を選択します。
ポート、プロトコルはそのまま「パスの分析と作成」を押します

パス設定

名前タグ - オプション

「Name」キーと、ユーザーが指定する値でタグを作成します。

送信元タイプ

Instances

送信元

i-Oaa6245c147fc416e

送信元 IP アドレス - オプション

192.0.2.1

送信先タイプ

Instances

送信先

送信先

送信先 IP アドレス - オプション

192.0.2.1

送信先ポート - オプション

数値を入力してください

0 から 65535 までの数値にする必要があります

プロトコル

適切なプロトコルを使用

TCP

2.4. しばらく待つと、分析結果が表示されます

nip-0bc760d55a63150cb

概要 情報

アクション ▼パスの分析

パス ID

nip-0bc760d55a63150cb

最終分析日

Fri Mar 12 2021 14:52:44 GMT+0900 (日本標準時)

到達のステータス

🟢 到達可能

前回の分析ステータス

🟢 成功しました

送信元

i-Oaa6245c147fc416e

送信先

i-0541964ae1f73950b

送信先ポート

-

プロトコル

TCP

分析 タグ

分析 (1/1) 情報

🔄 分析を削除

🔍 パス分析をフィルタリング

< 1 > ⚙️

分析 ID ▼

分析の実行日 ▼

到達のステータス ▼

中間コンポーネント... ▼

状態 ▼

🔵 nia-0f1fcd7914c1eb007

Fri Mar 12 2021 14:52:...

🟢 到達可能

-

🟢 成功しました

A から B は疎通できていることが確認できます。

2.5. 新しく A から C への分析を同様の手順で作成します。しばらく待つと「到達不可能」として分析が完了します

nip-0b57f6757f34b2ac1

概要 情報

アクション バスの分析

バス ID nip-0b57f6757f34b2ac1	最終分析日 Fri Mar 12 2021 14:54:48 GMT+0900 (日本標準時)	可達性ステータス 到達不可能	前回の分析ステータス 成功しました
送信元 i-0aa6245c147fc416e	送信先 i-02812a82eb9097605	送信先ポート -	プロトコル TCP

分析 タグ

分析 (1/1) 情報

分析をフィルタリング

分析を削除

分析 ID	分析の実行日	可達性ステータス	中間コンポーネント...	状態
nia-0d2f21f32955a58e4	Fri Mar 12 2021 14:54:...	到達不可能	-	成功しました

- 2.6. 以下のように、とあるセキュリティグループが ingress ルール（インバウンド通信）を許可していないため、不達となったことがわかります。

分析エクスプローラー 情報

送信先は到達不能です。詳細については、以下の説明を参照してください。 Give us feedback

Explanations

次のセキュリティグループの ingress ルールはいずれも適用されません: sg-031629fc11dbab506。

▼ Details

```
{
  "direction": "ingress",
  "explanationCode": "ENI_SG_RULES_MISMATCH",
  "networkInterface": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:network-interface/eni-0e03f230fda467f31",
    "id": "eni-0e03f230fda467f31"
  },
  "securityGroups": [
    {
      "arn": "arn:aws:ec2:us-east-1:294963776963:security-group/sg-031629fc11dbab506",
      "id": "sg-031629fc11dbab506"
    }
  ],
  "subnet": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:subnet/subnet-0e168275c122427a4",
    "id": "subnet-0e168275c122427a4"
  },
  "vpc": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:vpc/vpc-0a0a84436fc8cdb4a",
    "id": "vpc-0a0a84436fc8cdb4a"
  }
}
```

- 2.7. セキュリティグループをクリックします。「インバウンドルールを編集」を押します

sg-031629fc11dbab506 - VPCAnalyzer-SecurityGroup2-10P9BYWVUUBUV

アクション

詳細

セキュリティグループ名 VPCAnalyzer-SecurityGroup2-10P9BYWVUUBUV	セキュリティグループ ID sg-031629fc11dbab506	説明 Allow all egress traffic	VPC ID vpc-0a0a84436fc8cdb4a
所有者 294963776963	インバウンドルールカウント 0 アクセス許可エントリ	アウトバウンドルールカウント 1 アクセス許可エントリ	

インバウンドルール

アウトバウンドルール

タグ

インバウンドルール (0)

インバウンドルールを編集

タイプ	プロトコル	ポート範囲	ソース	説明 - オプション
ルールが見つかりません				
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。				

2.8. 以下のように TCP をインバウンドで設定し、「ルールを保存」を押します

インバウンドルールを編集 [情報](#)

インバウンドルールは、インスタンスに到達できる送信トラフィックをコントロールします。

インバウンドルール [情報](#)

タイプ [情報](#)

すべての TCP ▼

プロトコル [情報](#)

TCP

ポート範囲 [情報](#)

0 - 65535

ソース [情報](#)

カスタム ▼

説明 - オプション [情報](#)

Q

0.0.0.0/0 ✕

削除

ルールを追加

⚠ 注意: 既存のルールを編集すると、編集したルールが削除されて、新しい詳細を含む新しいルールが作成されます。これにより、そのルールに依存するトラフィックは、新しいルールが作成されるまで非常に短時間切断されます。

キャンセル

変更をプレビュー

ルールを保存

2.9. 先ほどの分析に戻り、「パスの分析」を押して、再実行します

VPC > 到達可能性アナライザー > nip-0b57f6757f34b2ac1

nip-0b57f6757f34b2ac1

概要 [情報](#)

アクション ▼

パスの分析

パス ID nip-0b57f6757f34b2ac1	最終分析日 Fri Mar 12 2021 14:54:48 GMT+0900 (日本標準時)	到達のステータス 🚫 到達不可能	前回の分析ステータス ✅ 成功しました
送信元 i-0aa6245c147fc416e	送信先 i-02812a82eb9097605	送信先ポート -	プロトコル TCP

2.10. オプションで通るべきパスを指定する画面がでてきますが、空欄のまま「確認」をおします

パスの分析

✕

パス ID
nip-0b57f6757f34b2ac1

送信元
i-0aa6245c147fc416e

送信先
i-02812a82eb9097605

下の **確認** ボタンをクリックして、このパスを分析することを確認してください。

中間コンポーネントフィルタを含める - オプション

Enter Amazon Resource Name

キャンセル

確認

2.11. 次は到達可能として分析が成功します

VPC > 到達可能性アナライザー > nip-0b57f6757f34b2ac1

nip-0b57f6757f34b2ac1

概要 情報

アクション ▼ バスの分析

バス ID nip-0b57f6757f34b2ac1	最終分析日 Fri Mar 12 2021 15:00:58 GMT+0900 (日本標準時)	到達のステータス 🟢 到達可能	前回の分析ステータス 🟢 成功しました
送信元 i-0aa6245c147fc416e	送信先 i-02812a82eb9097605	送信先ポート -	プロトコル TCP

分析

タグ

分析 (1/2) 情報

🔄 分析を削除

< 1 > ⚙️

分析 ID	分析の実行日	到達のステータス	中間コンポーネント...	状態
🔵 nia-0bd7828a49ab92009	Fri Mar 12 2021 15:00:...	🟢 到達可能	-	🟢 成功しました
🔴 nia-0d2f21f32955a58e4	Fri Mar 12 2021 14:54:...	🔴 到達不可能	-	🟢 成功しました

3. 異なる VPC への疎通 VPC Peering

3.1. これからの手順は複数の VPC を利用するため、非常に ID 表示などがややこしく、以下のパラメータをまずメモ帳などにメモったのち作業を始めましょう。

- ・今日作成した VPCID
- ・今日作成した VPCID の IPv4 CIDR
- ・A インスタンスの ID
- ・バージニア北部リージョンのデフォルト VPCID
- ・上記デフォルト VPC の IPv4 CIDR

3.2. デフォルト VPC が存在していない場合、別の VPC でも作業可能です。まず VPC 内に EC2 を 1 個作成します。OS、インスタンスタイプ、サブネットはなんでもかまいません。ログインしませんので、pem キーの保存などは不要です。EC2 の id を上記メモに追記します。

3.3. 作成されたインスタンスに[handson-target]と名前を付けておきます

3.4. 先ほどと同じ手順で VPC Reachability Analyzer の分析を、送信元「A」、送信先「handson-target」で作成し、実行します。以下のエラーが出力されます。

🔴 送信先は到達不能です。詳細については、以下の説明を参照してください。

Give us feedback

Explanations

クエリの送信元が VPC vpc-09cc3c6190f6821b6 にあり、クエリの送信先が VPC vpc-5e977a3a にあります。これらの VPC はピアリング接続されていないため、可能なパスはありません。

▼ Details

```
{
  "destinationVpc": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:vpc/vpc-5e977a3a",
    "id": "vpc-5e977a3a"
  },
  "explanationCode": "DISCONNECTED_VPCS",
  "sourceVpc": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:vpc/vpc-09cc3c6190f6821b6",
    "id": "vpc-09cc3c6190f6821b6"
  }
}
```


- 3.5. VPC マネージメントコンソールの左ペインから「ピアリング接続」を選び、「ピアリング接続の作成」ボタンをおします



- 3.6. ネームタグに「vpcpeeringtest」と名付け、リクエストに今日作成した VPC の ID を設定します

ピアリング接続の作成

ピアリング接続ネームタグ ⓘ

ピアリング接続するローカル VPC を選択

VPC (リクエスト)* ⓘ

CIDR	CIDR	ステータス	ステータスの理由
	172.0.0.0/16	● associated	

- 3.7. もうひとつの VPC を選択、の画面でデフォルト VPC の ID を選び、「ピアリング接続の作成」ボタンをおします

ピアリング接続するもうひとつの VPC を選択

アカウント ☒ 自分のアカウント
☐ 別のアカウント

リージョン ☒ このリージョン (us-east-1)
☐ 別のリージョン

VPC (アクセプタ)* ⓘ

CIDR	CIDR	ステータス	ステータスの理由
	172.31.0.0/16	● associated	

- 3.8. 「承諾の保留中」ステータスの接続がありますので、そちらを選んでアクションから「リクエストの承諾」を選びます。ダイアログが出てきますので「はい、承諾する」をおします

VPC ピアリング接続リクエストの承諾

×

この VPC ピアリング接続リクエスト (pcx-0e40a01fb6ff25e00) を承諾してよろしいですか?

リクエストアカウント ID	294963776963 (このアカウント)	アクセプタアカウント ID	294963776963 (このアカウント)
リクエスト VPC ID	vpc-09cc3c6190f6821b6	アクセプタ VPC ID	vpc-5e977a3a
リクエスト VPC リージョン	us-east-1	アクセプタ VPC リージョン	us-east-1
リクエスト VPC CIDR	172.0.0.0/16	アクセプタ VPC CIDR	-

キャンセル

はい、承諾する

次の画面ではお互いの VPC の相互通信を成立させるために必要なルートテーブルの設定喚起が出ていますが、一旦画面を閉じます。メモ帳に作成され Peering の ID を記載しておきます。(pcx-xxxxxxx…)

3.9. 再度 VPC Reachability Analyzer の画面で新しい分析を作成します。今度は送信元をインスタンス A、送信先を「VPC Peering test」とします。以下の通り今日作成した VPC からデフォルト VPC へ通信をルーティングさせる設定がないため、失敗します

Explanations

ルートテーブル `rtb-0b52582630f26461d` には `pcx-0e40a01fb6ff25e00` への適切なルートがありません。 [rtb-0b52582630f26461d](#) を参照してください。
▼ Details

```
{
  "destination": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:vpc-peering-connection/pcx-0e40a01fb6ff25e00",
    "id": "pcx-0e40a01fb6ff25e00"
  },
  "explanationCode": "NO_ROUTE_TO_DESTINATION",
  "routeTable": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:route-table/rtb-0b52582630f26461d",
    "id": "rtb-0b52582630f26461d"
  },
  "vpc": {
    "arn": "arn:aws:ec2:us-east-1:294963776963:vpc/vpc-09cc3c6190f6821b6",
    "id": "vpc-09cc3c6190f6821b6"
  }
}
```

3.10. VPC のマネージメントコンソールから今日作成した VPC の詳細画面へいき、ルートテーブル ID を特定します。（上記エラーで表示されているルートテーブル ID を右クリックでも遷移します） クリックするとルートテーブル設定画面へ遷移します

vpc-09cc3c6190f6821b6

アクション ▼

詳細 情報

VPC ID
vpc-09cc3c6190f6821b6

デナンスー
Default

デフォルト VPC
いいえ

所有者 ID
294963776963

状態
Available

DHCP オプションセット
dopt-c0c23fa5

IPv4 CIDR
172.0.0.0/16

DNS ホスト名
有効

メインルートテーブル
rtb-0b52582630f26461d

IPv6 プール
-

DNS 解決
有効

メインネットワーク ACL
acl-0679948d69a9f5a43

IPv6 CIDR (ネットワークボーダークラウドグループ)
-

3.11. アクションから、「ルートの編集」をえらびます

ルートテーブル > ルートの編集

ルートの編集

送信先	ターゲット	ステータス	伝播済み
172.0.0.0/16	local	active	いいえ

ルートの追加

* 必須

キャンセル ルートの保存

3.12. 現時点での設定では、今日作成した VPC 内部のみが設定されていますので、先ほど Peering したデフォルト VPC へのルートを追加します。ターゲットにはデフォルト VPC の ID ではなく、Peering Connection を選びます。以下のような画面になったら「ルートの保存」を押します。ここで入力されている IP アドレスブロックは、先ほどメモをしたデフォルト VPC のブロックです

ルートの編集

送信先	ターゲット	ステータス	伝播済み
172.0.0.0/16	local	active	いいえ
172.31.0.0/16	pcx-0e40a01fb6ff25e00		いいえ

[ルートの追加](#)

* 必須

[キャンセル](#) [ルートの保存](#)

- 3.13. 再度先ほどの分析を実行すると以下のように VPC Peering までの疎通の確認ができました。ルートテーブルの反映は少し時間がかかるため、エラーとなる場合は、1 分程度たってから再度分析を実行してみてください



- 3.14. 再度新しい分析を作成して、送信元と A,送信先を handson-target として、分析を実行します。以下のエラーが出力されます。作成した VPC からデフォルト VPC へのルートはできましたが、デフォルト VPC から今日作成した VPC へのルートがないためです。同様の手順でルートテーブルを設定し、分析を実行すると以下のように成功します。

分析エクスプローラー 情報

☐ リバースパスを表示



- 3.15. VP マネージメントコンソール、左ペインからピアリング接続の設定を、アクションボタンのドロップダウンから削除して、再度分析を行ってください。以下のように到達不可能となります。タイミングで異なるエラーが出る場合がありますが、到達不可能となっていればこの手順では正解です

分析エクスペローラ情報

送信先は到達不能です。詳細については、以下の説明を参照してください。

Give us feedback

Explanations

- ルートのテーブル `rtb-0b52582630f26461d` の `172.31.0.0/16` から `vpcPeeringConnectionId` へのルートが機能不全状態 `none` になっており、使用できません。

Details
- コンポーネント `pcx-0e40a01fb6ff25e00` が機能不全状態 `deleted` になっています。

Details

3.16. 2つのルートテーブルから、VPC Peering 向けのルートを削除します。
それぞれ以下が正しい状態です。

ルートの編集

表示

すべてのルート

送信先	ターゲット	ステータス	伝播済み
172.0.0.0/16	local	active	いいえ

表示

すべてのルート

送信先	ターゲット	ステータス	伝播済み
172.31.0.0/16	local	active	いいえ
0.0.0.0/0	igw-9a104cff	active	いいえ

4. おつかれさまでした
削除は、
CloudFormation スタック
デフォルト VPC の EC2 インスタンス
を行ってください。