



# AWS 練習問題

はじめに .....	2
1 IAM ユーザーで AWS ヘログイン .....	3
2 パブリックサブネットの作成 .....	4
3 EC2 を利用した Web ページの公開 .....	6
4 Oracle DB インスタンスの構築 .....	9
5 NoSQL テーブルでのクエリの実行 .....	10
6 Web アプリケーションのデプロイ .....	11
7 静的 WEB ページのホスティング .....	24
8 スケーリングによる EC2 の自動復旧 .....	25
9 CDN によるコンテンツ配信 .....	26

## はじめに

AWS では、インターネット経由でサーバー、ストレージ、データベース、ソフトウェアといった様々なサービスが提供されています。練習問題では、講義内で学んだことを生かし、さらに AWS に慣れていきましょう。

テキストやインターネットを使って調べても構いません。

なお、ハンズオンをする際には会社から付与されている AWS アカウントを使用してください。

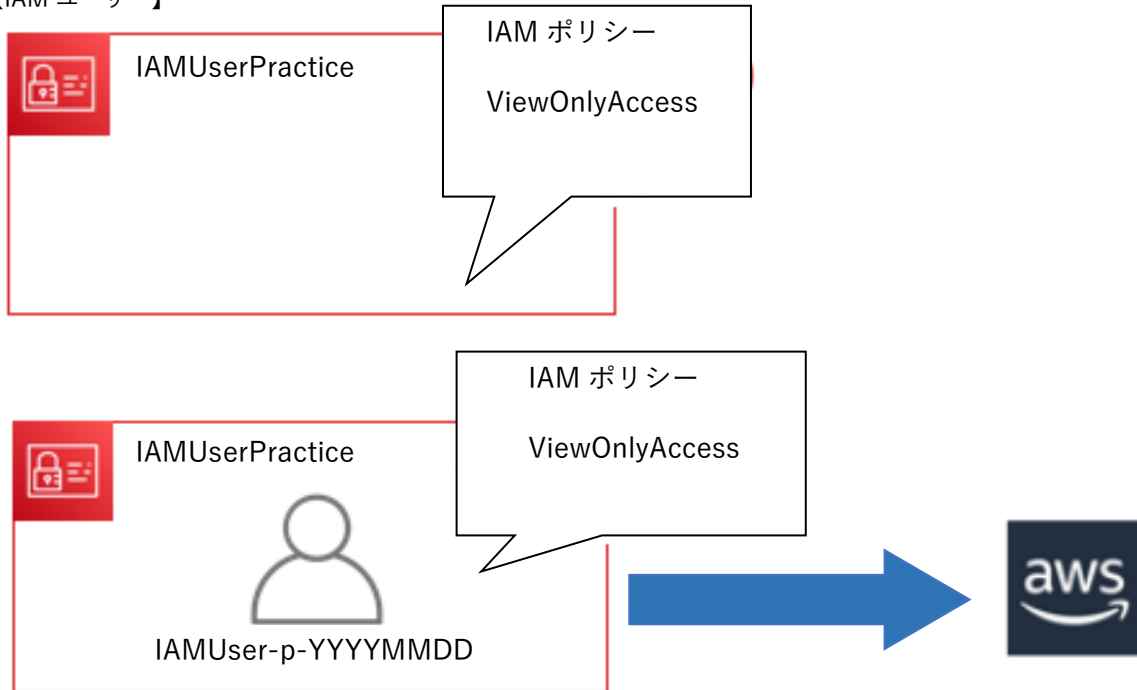
## 1 IAM ユーザーで AWS へログイン

以下の条件に沿って IAM ユーザーを作成してください。また、作成したユーザーで AWS マネジメントコンソールへサインインできることを確認してください。

○完成図

【IAM グループ】

【IAM ユーザー】



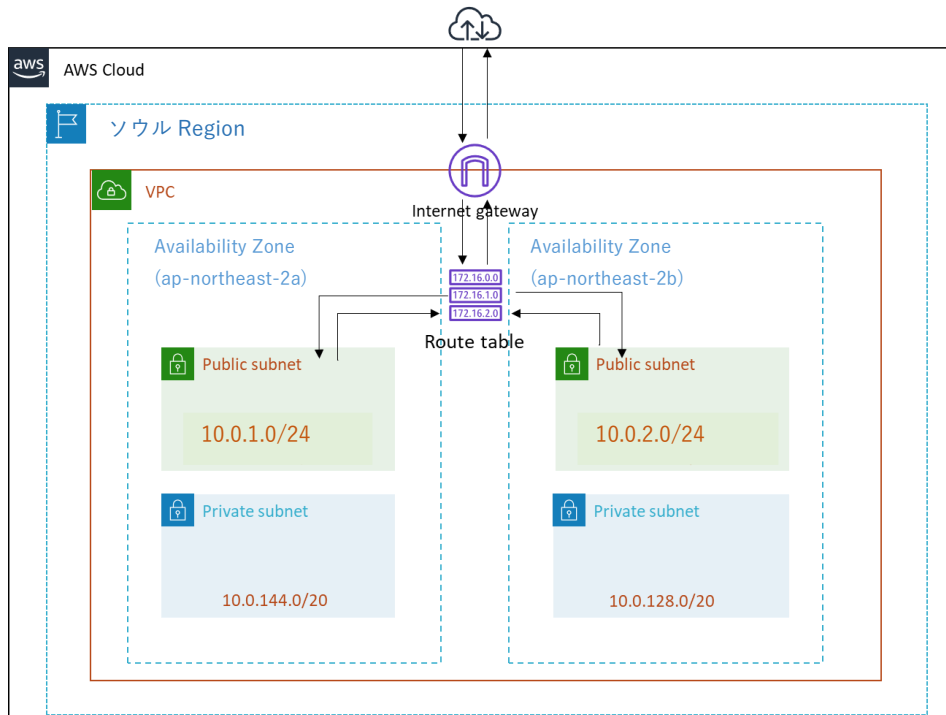
○条件

- ・ ユーザーグループ：IAMUserPractice
- ・ グループに設定するポリシー：ViewOnlyAccess
- ・ ユーザー名：IAMUserPracticeYYYYMMDD (YYYYMMDD は本日の日付)
- ・ アクセスの種類：パスワード・AWS マネジメントコンソールへのアクセス
- ・ コンソールのパスワード：自動生成パスワード
- ・ パスワードのリセットが必要：[ユーザーは次のサインインでパスワードを作成する必要があります]  
※チェックを入れる。

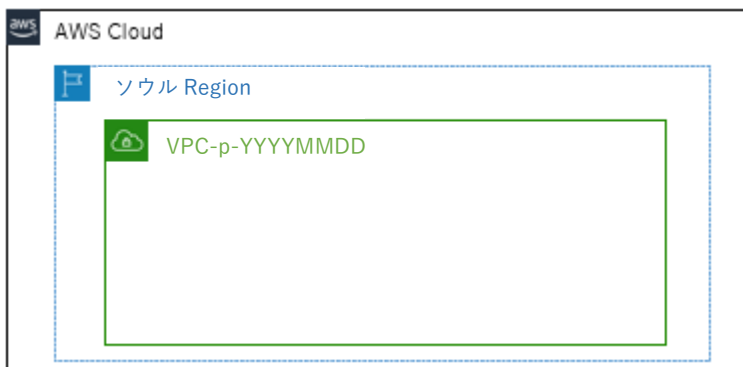
## 2 パブリックサブネットの作成

以下の条件に沿って VPC とパブリックサブネットを作成し、インターネットからアクセス可能な状態にしてください。  
作成した VPC、パブリックサブネットは、次の EC2 の練習問題終了後に削除してください。

○完成図



【VPC の設定】



=====

○条件

- ・ 作成するリソース：VPC など
- ・ 名前タグの自動生成：チェックする
- ・ 名前タグ：VPC-p-**YYYYMMDD** (**YYYYMMDD** は本日の日付)
- ・ IPv4 CIDR ブロック：10.0.0.0/16
- ・ IPv6 CIDR ブロック：IPv6CIDR ブロックなし
- ・ テナンシー：デフォルト
- ・ アベイラビリティゾーンの数：2
- ・ 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
- ・ NAT ゲートウェイ：なし
- ・ VPC エンドポイント：なし
- ・ DNS ホスト名を有効化：チェックする
- ・ DNS 解決を有効化：チェックする

○パブリックサブネットの作成方法

※↑VPC 作成後に実施すること

- ① サービス一覧から VPC を検索
- ② VPC ダッシュボードの画面左にあるメニューから [サブネット] をクリック
- ③ 作成されているパブリックサブネットを削除
- ④ 「サブネット作成」をクリックし、サブネット作成画面に移動をしたら以下の設定でサブネットを作成する。

<1 つめ>

- ・ VPC:自分で作成したもの(VPC-p-**YYYYMMDD**)
- ・ サブネット名：VPC-p-**YYYYMMDD**-subnet-public-ap-northeast-2a
- ・ アベイラビリティゾーン：アジアパシフィック(ソウル)/ap-northeast-2a
- ・ IPv4 CIDR ブロック：10.0.1.0/24

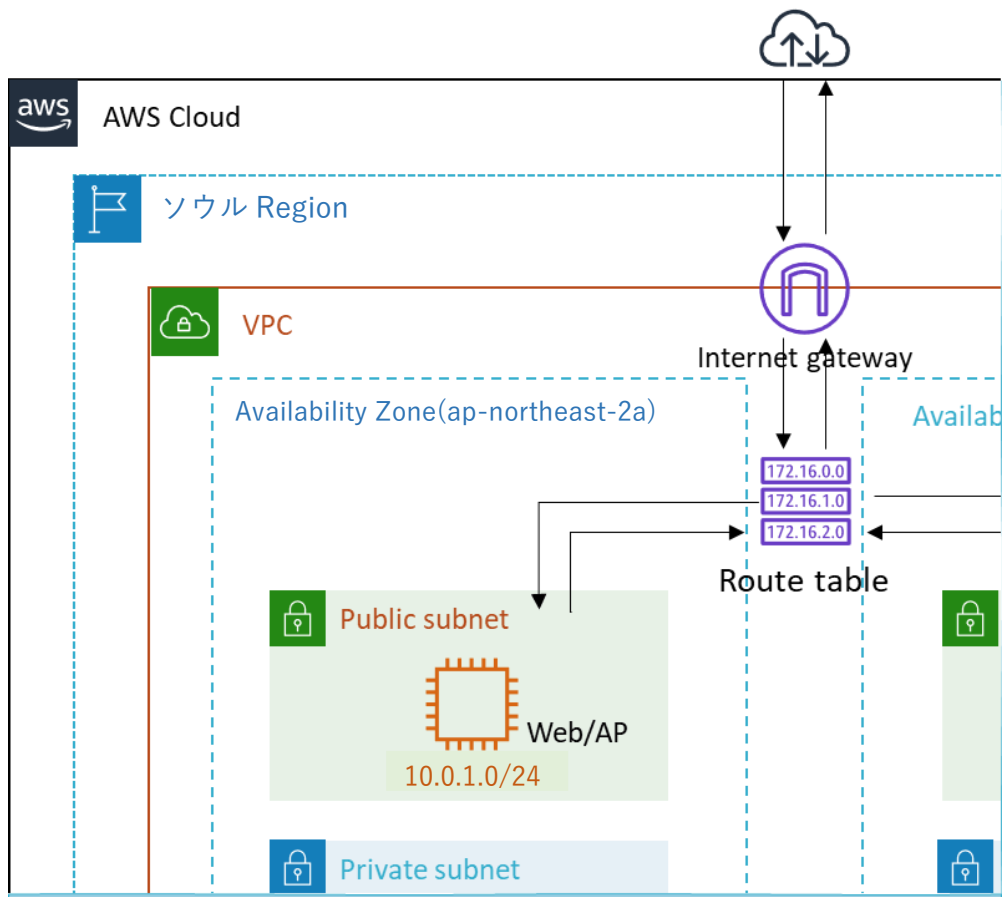
<2 つめ>

- ・ VPC:自分で作成したもの(VPC-p-**YYYYMMDD**)
- ・ サブネット名：VPC-p-**YYYYMMDD**-subnet-public-ap-northeast-2b  
(**YYYYMMDD** は本日の日付)
- ・ アベイラビリティゾーン：アジアパシフィック(ソウル)/ap-northeast-2b
- ・ IPv4 CIDR ブロック：10.0.2.0/24

### 3 EC2 を利用した Web ページの公開

以下の条件に沿って EC2 を作成し、Web ページをインターネット上に公開してください。  
公開する web ページは、「【配布】HTML,SQL,画像」フォルダの index.html を使用してください。  
なお、問題を解き終わり次第、作成した EC2 と VPC は削除してください。

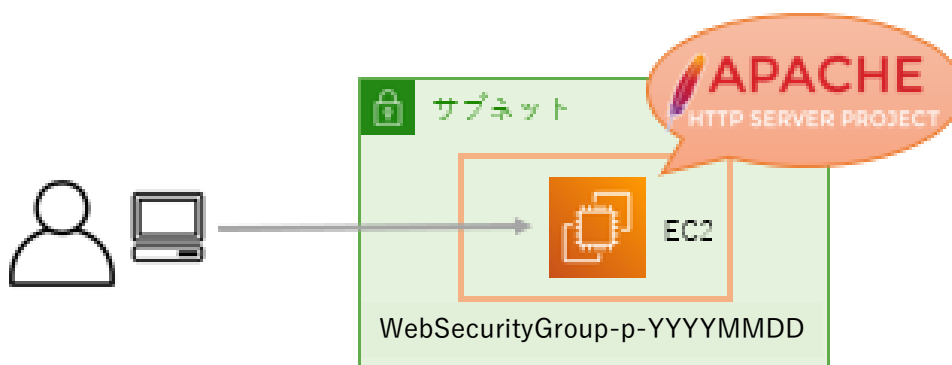
○完成図



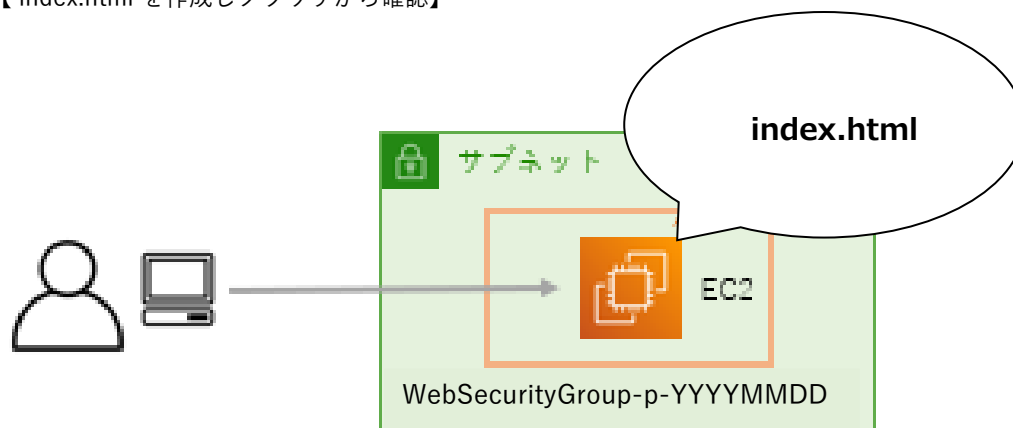
【EC2 とセキュリティグループ】



【EC2 へ Apache HTTP Server をインストール】



【index.html を作成しブラウザから確認】





=====

○条件

① EC2

- ・ 名前：EC2-p-YYYYMMDD
- ・ (YYYYMMDD は本日の日付)
- ・ クイックスタート：Red Hat
- ・ マシンイメージ：Red Hat Enterprise Linux8(HVM),SSD Volume Type
- ・ インスタンスタイプ：t2.micro
- ・ キーペア名：EC2Key-p-YYYYMMDD ※新しいキーペアの作成
- ・ キーペアのタイプ：RSA
- ・ プライベートキーファイル作成：.pem
- ・ VPC：VPC-p-YYYYMMDD ※[編集]をクリックし、先程作成した VPC を選択する。
- ・ サブネット：VPC-p-YYYYMMDD-subnet-public1-ap-northeast-2a  
※先程作成したパブリックサブネットを選択
- ・ パブリック IP の自動割り当て：有効化
- ・ ファイアウォール（セキュリティグループ）：WebSecurityGroup-p-YYYYMMDD
- ・ セキュリティグループルール 1：  
タイプ：ssh  
ソースタイプ：My IP  
※InstanceConnect を使用する場合は、ソースタイプ[カスタム]、  
ソース[3.112.23.0/29]を設定。
- ・ セキュリティグループルール 2：（Add security group rule ボタンをクリックして追加）  
タイプ：HTTP  
ソースタイプ：任意の場所
- ・ ストレージ：1x [10] GiB gp2

② SSH 接続

- ・ ユーザー名：ec2-user
- ・ 認証方式：[RSA/DSA/ECDSA/ED25519 鍵を使う]
- ・ 秘密鍵：KeyPair-p-YYYYMMDD (EC2 作成時にダウンロードしたキーペア)

③ 作成する index.html

別フォルダで配布した index.html をコピーして作成してください。

→配布フォルダ名：【配布用】HTML,SQL,画像,jar

ファイル名 : index.html

=====

## 4 Oracle DB インスタンスの構築

以下の条件に沿って、Oracle DB インスタンスを作成し、接続を行ってください。なおインスタンス作成は Amazon RDS を利用することとします。作成したインスタンスは次の web アプリケーションのデプロイが終わり次第、削除するようにしてください。

※1 日で問題を解き終わらない場合は、インスタンスは停止させておくこと。(料金が発生するため)

### ○条件

#### ① データベースの設定

データベース作成方法	: 標準作成
エンジンのタイプ	: Oracle
エディション	: Oracle Standard Edition Two
ライセンス	: license-included
バージョン	: デフォルトで選択されているバージョン
テンプレート	: 開発/テスト
DB インスタンス識別子	: OracleDB-p-Instance
マスターユーザー名	: oracleadmin
パスワード自動生成	: 有効
DB インスタンスクラス	: バースト可能クラス(t クラスを含む) [db.t3.small]
ストレージタイプ	: 汎用 (SSD)
ストレージ割り当て	: 20
ストレージの自動スケーリングを有効にする	: 有効
最大ストレージしきい値	: 1000
Virtual Private Cloud	: Default VPC (vpc-xxxxxx) (デフォルト VPC を選択)
サブネットグループ	: default-vpc-xxxxxx (デフォルトサブネットを選択)
パブリックアクセス可能	: あり
VPC セキュリティグループ	: 新規作成
新しい VPC セキュリティグループ名	: DBInstanceSecurityGroup-p-YYYYMMDD (YYYYMMDD は本日の日付)

#### ② SQL Developer で接続する時

Connection Name (接続名)	: Oracle-RDS
Username (ユーザー名)	: oracleadmin
Password (パスワード)	: データベース管理者のパスワード
Hostname (ホスト名)	: DB インスタンスのエンドポイント
Port (ポート)	: 1521
SID	: ORCL

## 5 NoSQL テーブルでのクエリの実行

以下の条件に沿って DynamoDB でテーブルを作成してください

また、作成したデータに追加・変更・削除を行ってください。

作成したテーブルは、最後に必ず削除するようにしてください。

=====

○条件

① 作成するテーブル

テーブル名	: Sports
パーティションキー	: event
ソートキー	: type

② 追加するデータ

パーティションキー(event)	: baseball
ソートキー(type)	: ball game

パーティションキー(event)	: wrestling
ソートキー(type)	: ball game

パーティションキー(event)	: swimming
ソートキー(type)	: water sports

パーティションキー(event)	: soccer
ソートキー(type)	: ball game

③ 検索するデータ

パーティションキー(event)	: baseball
ソートキー(type)	: ball game

④ 変更するデータ

パーティションキー(event)	: wrestling
ソートキー(type)	: fighting sports

⑤ 削除するデータ

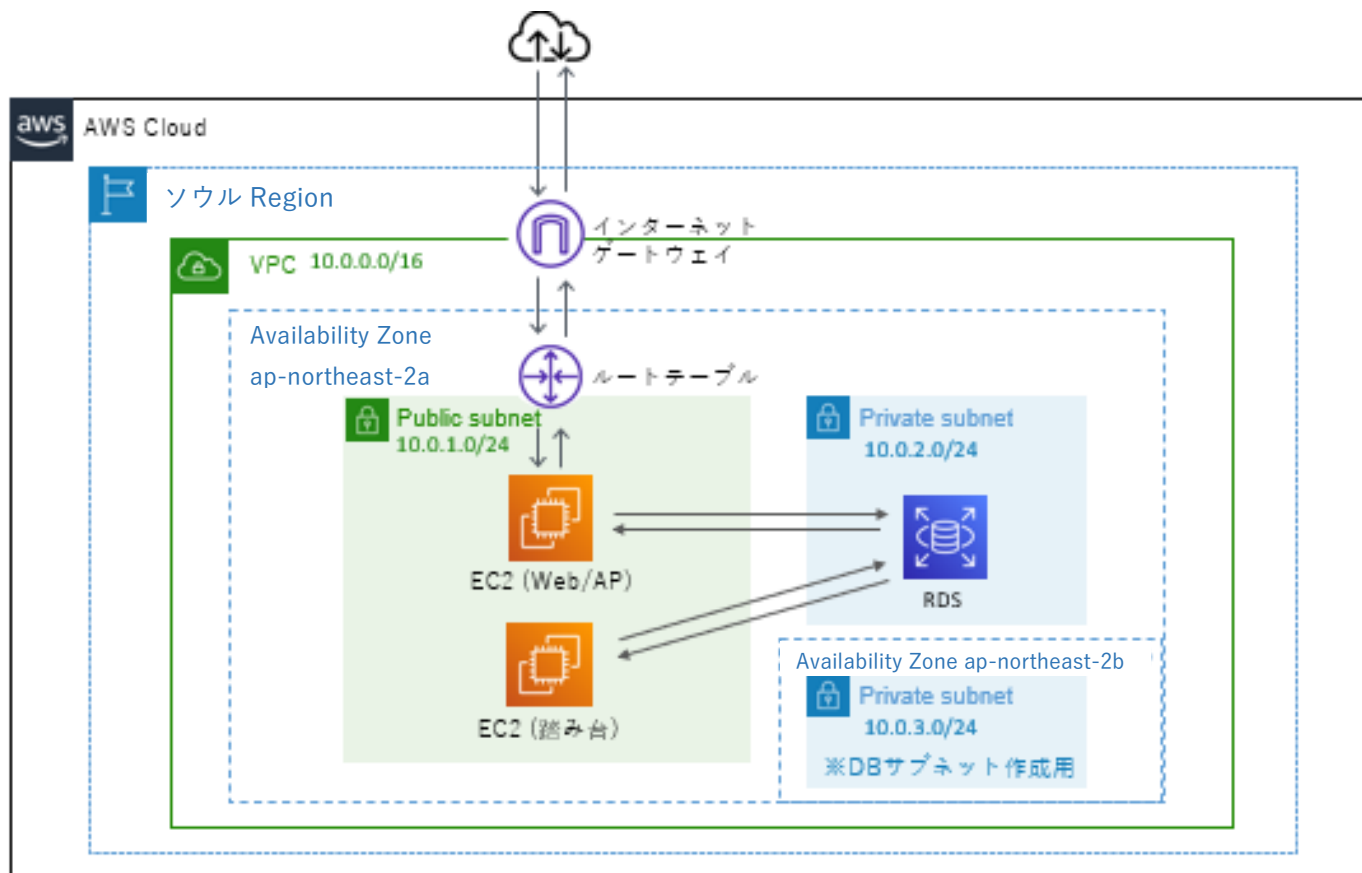
パーティションキー(event)	: baseball
ソートキー(type)	: ball game

=====

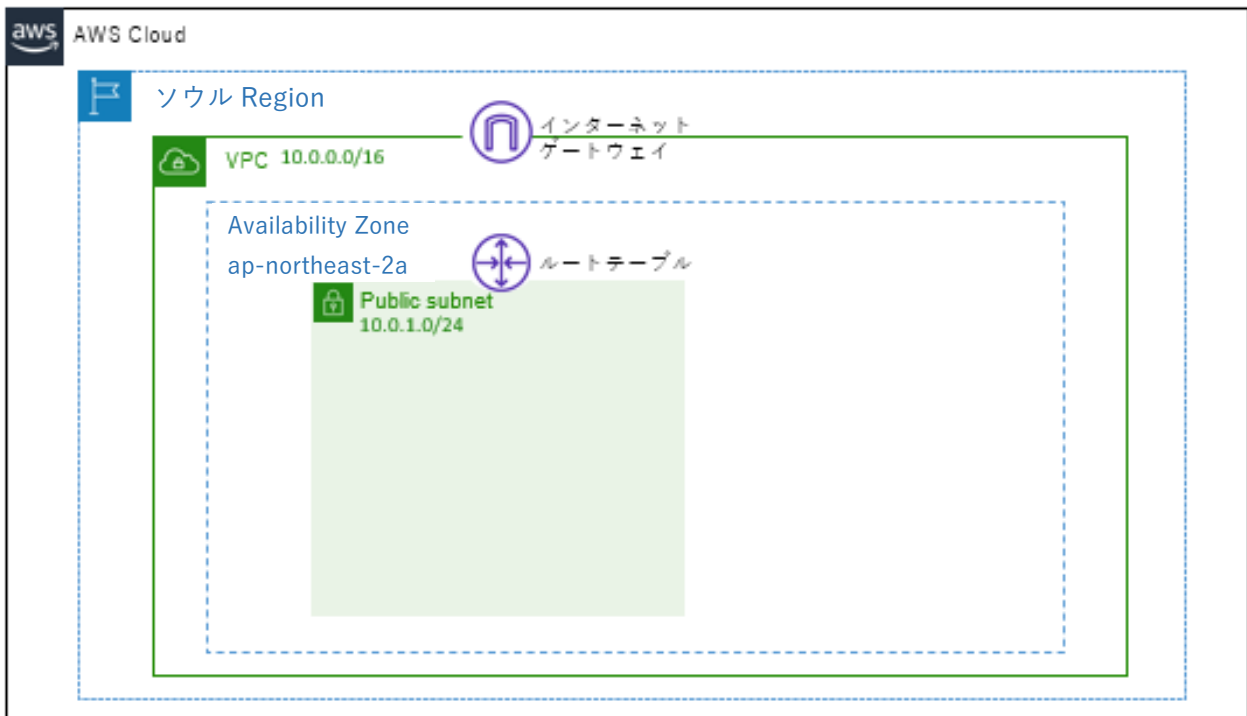
## 6 Web アプリケーションのデプロイ

以下の条件に沿って Web アプリケーション（学生検索システム）をインターネット上に公開しましょう。  
作成した AWS サービスは、デプロイ終了後すべて削除してください。

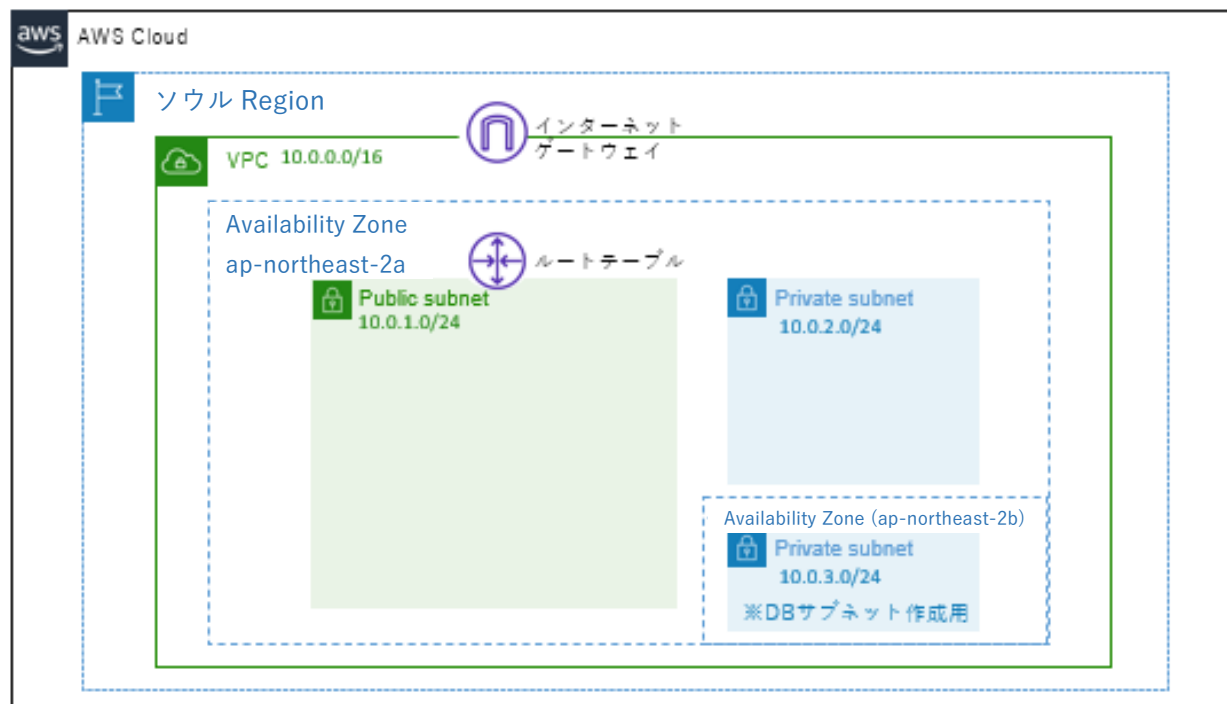
〇完成図



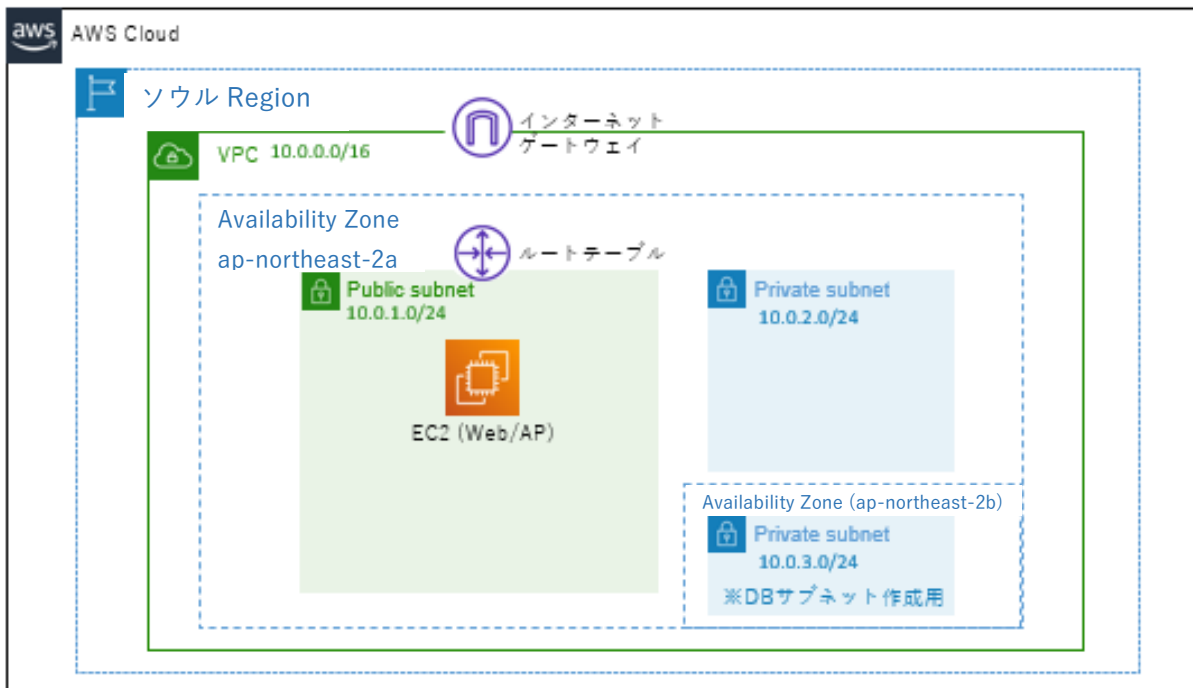
【VPC の作成】



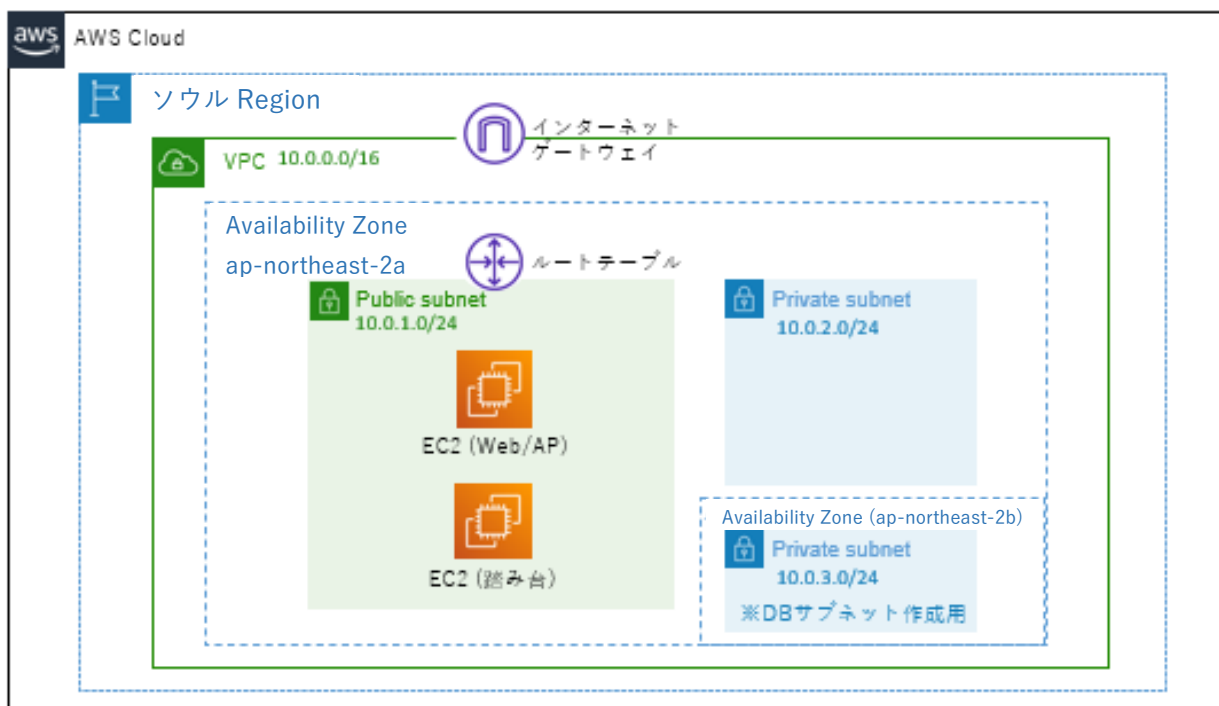
【プライベートサブネットの作成】



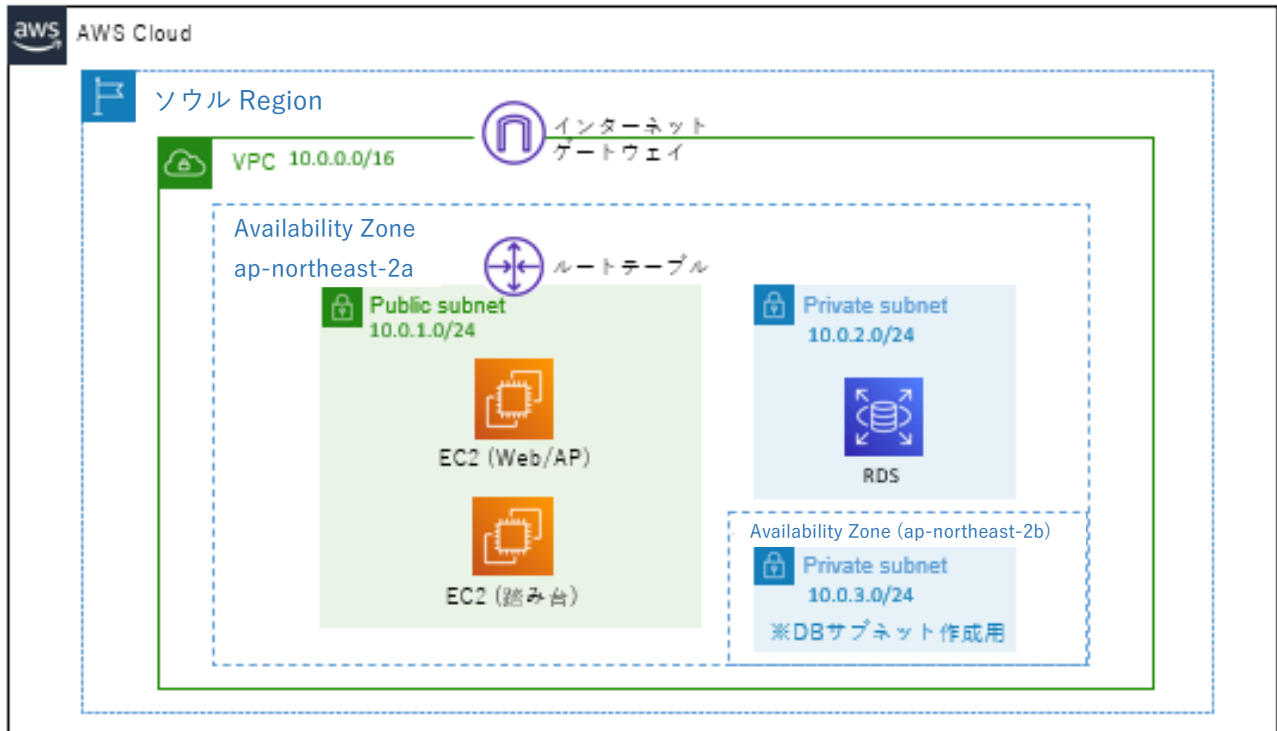
【EC2 インスタンス (Web/AP) を作成】



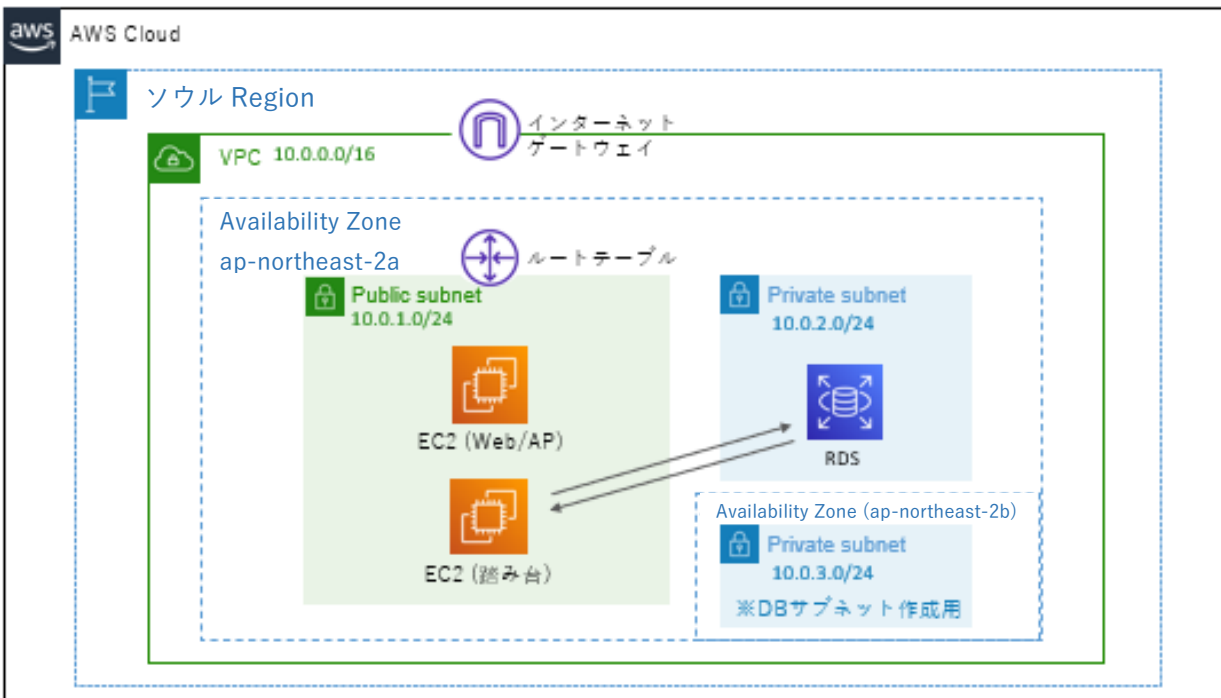
【EC2 インスタンス(踏み台)を作成】



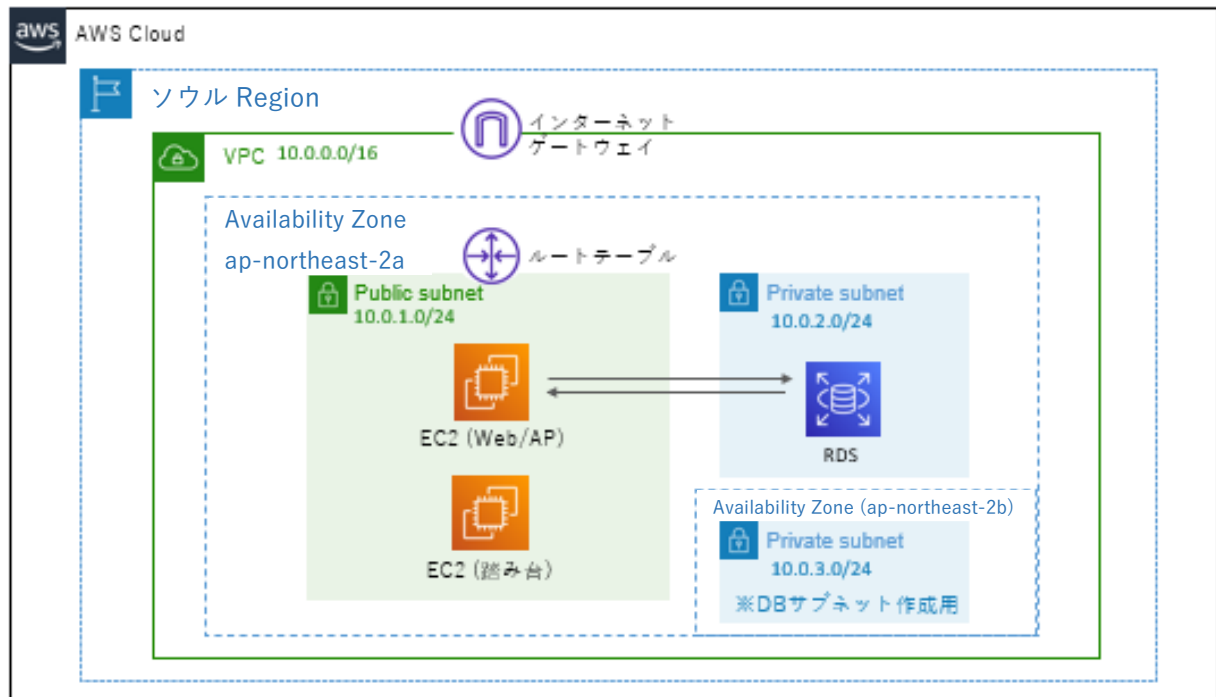
【RDS の作成】



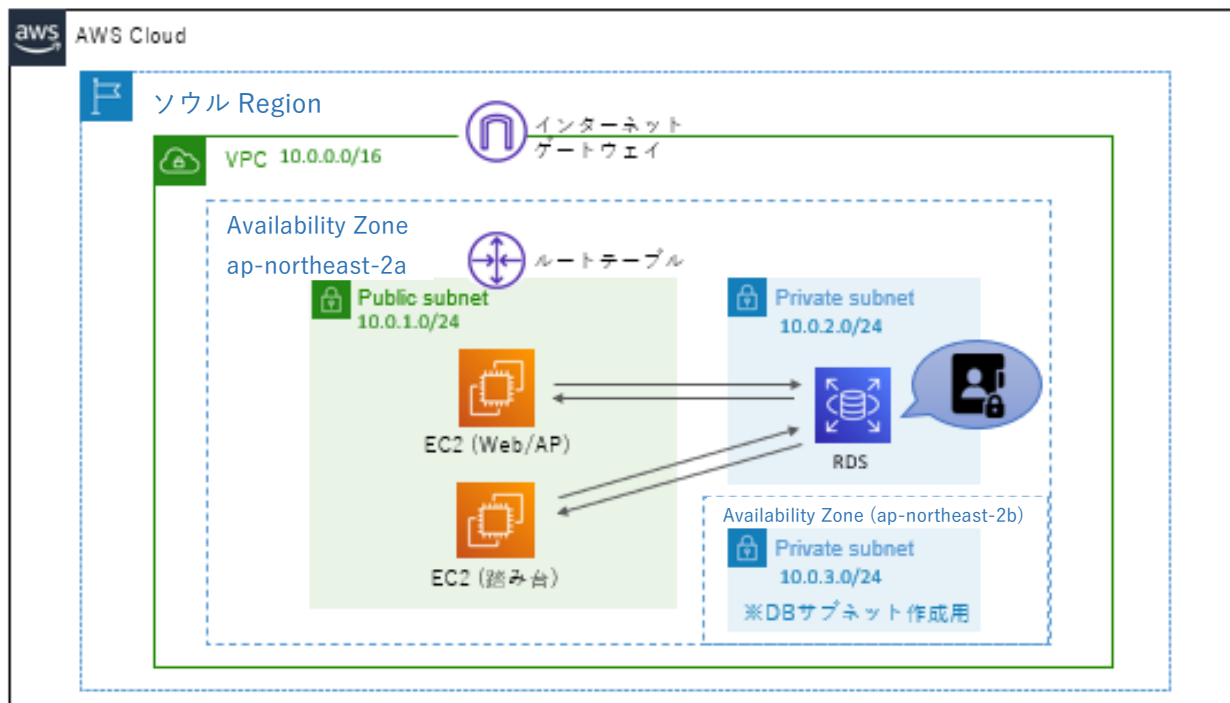
【EC2(踏み台)と RDS を接続】



【EC2 インスタンス(WEB/AP)と RDS を接続】

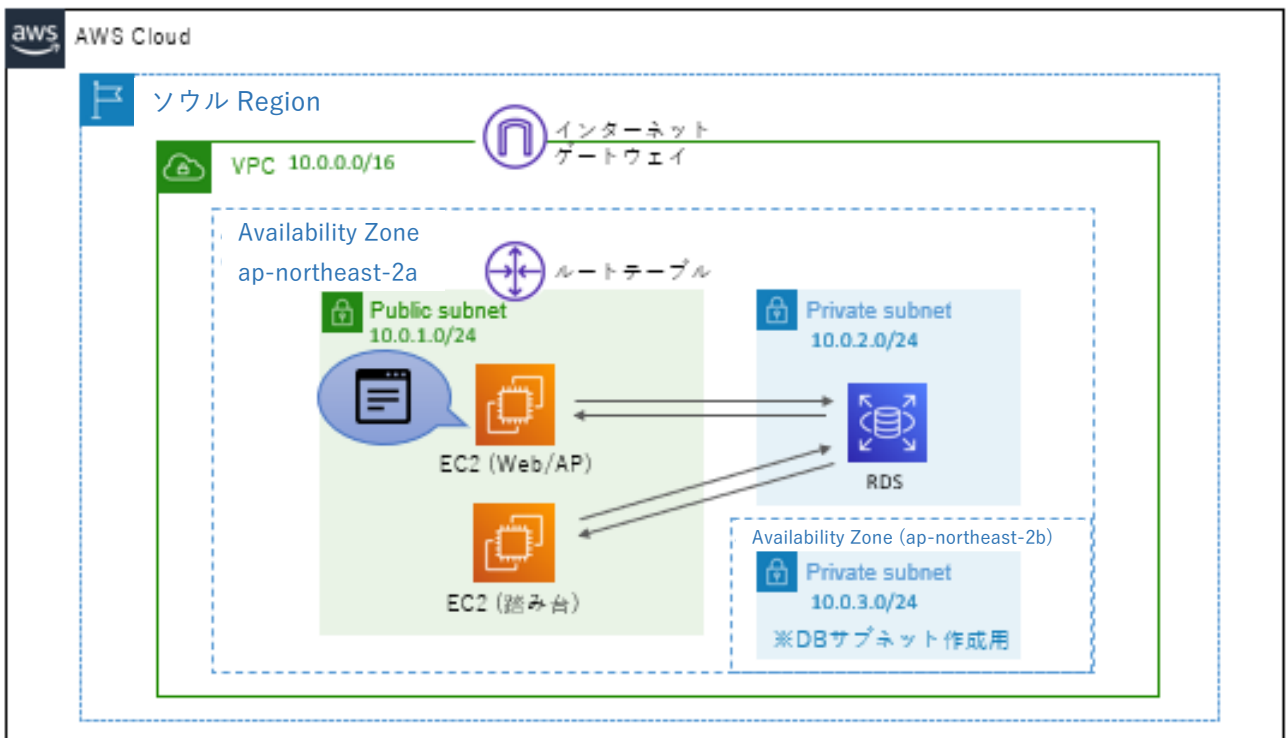


【アプリケーションのデプロイ(RDS)】

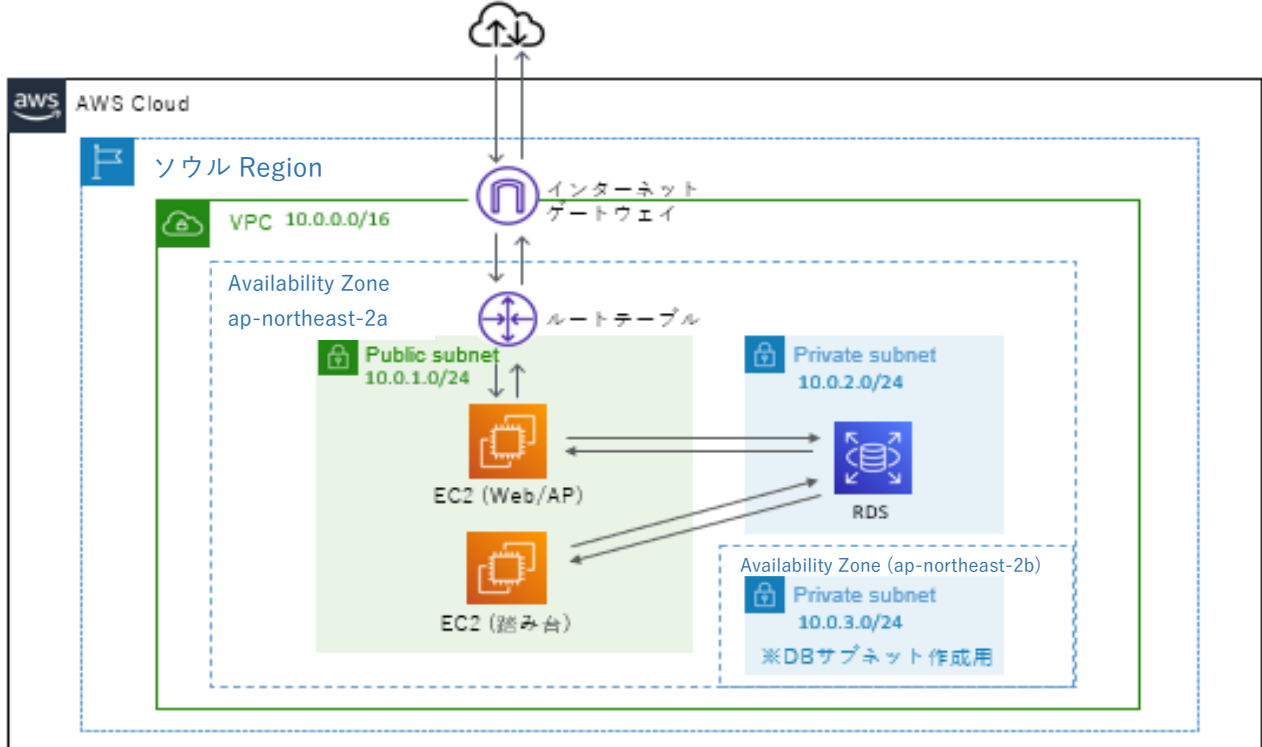




【アプリケーションのデプロイ (EC2)】



【ブラウザからアプリケーションにアクセス】



=====

○条件

① VPC

- ・ 名前タグ : DeployVPC-p-YYYYMMDD (YYYYMMDD は本日の日付)
- ・ IPv4 CIDR ブロック : 10.0.0.0/16

② パブリックサブネット

- ・ VPC : DeployVPC-p-YYYYMMDD (①の VPC)
- ・ サブネット名 : DeployPublicSubnet-p-YYYYMMDD
- ・ アベイラビリティーゾーン : ap-northeast-2a
- ・ IPv4 CIDR ブロック : 10.0.1.0/24

③ インターネットゲートウェイ

- ・ 名前タグ : DeployInternetGateway-p-YYYYMMDD

④ ルートテーブル

- ・ 名前タグ : DeployRouteTable-p-YYYYMMDD
- ・ VPC : DeployVPC-p-YYYYMMDD (①の VPC)
- ・ サブネットの関連付け
  - DeployPublicSubnet-p-YYYYMMDD (②で作成したサブネットを選択)
- ・ ルートの追加
  - 送信先 : 0.0.0.0/0
  - ターゲット : DeployInternetGateway-p-YYYYMMDD

⑤ プライベートサブネット

1 つめ

- ・ VPC : DeployVPC-p-YYYYMMDD (①の VPC)
- ・ サブネット名 : DeployPrivateSubnet2a-p-YYYYMMDD
- ・ アベイラビリティーゾーン : ap-northeast-2a
- ・ IPv4 CIDR ブロック : 10.0.2.0/24

2 つめ

- ・ VPC : DeployVPC-p-YYYYMMDD (①の VPC)
- ・ サブネット名 : DeployPrivateSubnet2b-p-YYYYMMDD
- ・ アベイラビリティーゾーン : ap-northeast-2b
- ・ IPv4 CIDR ブロック : 10.0.3.0/24

⑥ EC2(WEB/AP)

- ・インスタンス名：DeployWebInstance-p-YYYYMMDD  
(YYYYMMDD は本日の日付)
- ・Amazon マシンイメージ (AMI)：Amazon Linux 2 AMI (HVM), SSD Volume Type  
インスタンスタイプの選択：t2.micro
- ・キーペア：新しいキーペアの作成  
キーペア名：DeployKeyPair-p-YYYYMMDD
- ・ネットワーク：DeployVPC-p-YYYYMMDD (①の VPC)
- ・サブネット：DeployPublicSubnet-p-YYYYMMDD (②のサブネット)
- ・自動割り当てパブリック IP：[有効]
- ・セキュリティグループの割り当て：[新しいセキュリティグループを作成する]
- ・セキュリティグループ名：DeployWebSecurityGroup-p-YYYYMMDD
- ・説明：Security Group for YYYYMMDD Deploy Hands On Web Instance
- ・ルール：タイプ：SSH      ソース：[マイ IP]
- ・ストレージ設定：デフォルトのまま

⑦ EC2(踏み台)

- ・インスタンス名：DeployBastionInstance-p-YYYYMMDD
- ・Amazon マシンイメージ (AMI)：Amazon Linux 2 AMI (HVM), SSD Volume Type  
インスタンスタイプの選択：t2.micro
- ・キーペア：新しいキーペアの作成  
キーペア名：DeployKeyPair-p-Bastion-YYYYMMDD
- ・ネットワーク：DeployVPC-p-YYYYMMDD (①の VPC)
- ・サブネット：DeployPublicSubnet-p-YYYYMMDD (②のサブネット)
- ・自動割り当てパブリック IP：[有効]
- ・セキュリティグループの割り当て：[新しいセキュリティグループを作成する]
- ・セキュリティグループ名：DeployBastionSecurityGroup-p-YYYYMMDD
- ・説明：Security Group for YYYYMMDD Deploy Hands On Bastion Instance
- ・ルール：タイプ：SSH      ソース：[My IP]
- ・ストレージ設定：デフォルトのまま

⑧ RDS のサブネット

- ・名前：p-deploynsubnetgroup
- ・説明：DB Subnet Group for Deploy Hands On
- ・VPC：DeployVPC-p-YYYYMMDD
- ・アベイラビリティゾーン：[ap-northeast-2a] [ap-northeast-2b]
- ・サブネット：[10.0.2.0/24 のサブネット] [10.0.3.0/24 のサブネット]

⑨ セキュリティグループ(EC2 のダッシュボードから作成)

- ・セキュリティグループ名：DeployDBSecurityGroup-p-YYYYMMDD
- ・説明：Security Group for Deploy Hands On DB Instance p YYYYMMDD
- ・VPC：DeployVPC-p-YYYYMMDD

➤ インバウンドルール

① インバウンドルール 1

- ・ タイプ 1 : Oracle-RDS
- ・ リソースタイプ 1 : カスタム
- ・ ソース 1 : DeployWebSecurityGroup-p-**YYYYMMDD** のセキュリティグループ ID

② インバウンドルール 2

※[ルールを追加]をクリックし、インバウンドルール 2 を設定します。

- ・ タイプ 2 : Oracle-RDS
- ・ リソースタイプ 2 : カスタム
- ・ ソース 2 : DeployBastionSecurityGroup-p-**YYYYMMDD** のセキュリティグループ ID

⑩ RDS

➤ データベース作成方法を選択

- ・ データベースの作成方法 : [標準作成]

➤ エンジンのオプション

- ・ エンジンのタイプ : Oracle
- ・ エディション : Oracle Standard Edition Two
- ・ ライセンス : license-included
- ・ バージョン : Oracle 19.0.0.0.ru-2020-07.rur-2020-10.r1

※無い場合は一番新しいバージョンを選択

➤ テンプレート

- ・ テンプレート : 開発/テスト

➤ 設定

- ・ DB インスタンス識別子 : DeployOracleInstance-p-**YYYYMMDD**  
(**YYYYMMDD** は本日の日付)
- ・ マスターユーザー名 : oracleadmin
- ・ パスワードの自動生成 : 有効

➤ DB インスタンスサイズ

- ・ DB インスタンスクラス : バースト可能クラス(t クラスを含む)[db,t3,small]

➤ ストレージ

- ・ ストレージタイプ : 汎用(SSD)
- ・ ストレージ割り当て : 20
- ・ ストレージの自動スケーリング : 無効 (チェックを外す)
- ・ 最大ストレージしきい値 : 1000

➤ 可用性と耐久性

- ・ マルチ AZ 配置 : スタンバイインスタンスを作成しないでください

➤ 接続

- ・ VPC : DeployVPC-p-**YYYYMMDD**
- ・ サブネットグループ : p-deploysubnetgroup
- ・ パブリックアクセス可能 : なし
- ・ VPC セキュリティグループ : DeployDBSecurityGroup-p-**YYYYMMDD**
- ・ アベイラビリティゾーン : ap-northeast-2a

➤ データベース認証

- ・ データベース認証オプション : パスワード認証

➤ 追加設定

- ・ 最初のデータベース名 : oracledb
- ・ DB パラメータグループ : default.oracle-se2-19 (初期値)
- ・ オプショングループ : default.oracle-se2-19 (初期値)
- ・ 文字セット : AL32UTF8
- ・ 自動バックアップの有効化 : 無効 (チェックを外す)
- ・ バックアップの保持期間 : 7 日間
- ・ バックアップウィンドウ : 設定なし
- ・ スナップショットにタグをコピー : 有効
- ・ 暗号を有効化 : 無効 (チェックを外す)
- ・ Performance Insights を有効にする : 無効 (チェックを外す)
- ・ 拡張モニタリングの有効化 : 無効 (チェックを外す)
- ・ アラートログ : 無効
- ・ 監査ログ : 無効
- ・ リスナーログ : 無効
- ・ トレースログ : 無効
- ・ マイナーバージョン自動アップグレードの有効化 : 無効
- ・ メンテナンスウィンドウ : 設定なし
- ・ 削除保護の有効化 : 無効

⑩ EC2 インスタンス(踏み台)と RDS を接続

- ・名前: bastion\_ec2\_p
- ・ホスト: EC2 (踏み台) のパブリック IP アドレス
- ・ユーザー名: [ec2-user]
- ・キー・ファイルの使用: 有効 [DeployKeyPair-p-YYYYMMDD]※EC2 インスタンスの作成時に指定したキーペアファイルを[参照]ボタンから選択。
- ・ローカル: ポート転送の追加: 有効
- ・名前: ec2\_rds\_p
- ・ホスト: RDS のエンドポイント

➤ RDS との接続設定(SQL Developer)

- ・名前: rds\_connection
- ・ユーザー名: oracleadmin (RDS 作成時に設定した値)
- ・パスワード: RDS 作成時に生成されたマスターパスワード
- ・接続タイプ: SSH
- ・ポート転送: bastion\_ec2\_p
- ・SID: oracledb (RDS 作成時に設定した値)

⑪ EC2 インスタンス(WEB/AP)と RDS を接続

テキスト[CLD01]クラウド基礎 2-2\_講義資料\_1.2pdf の P34 を参照しましょう。

練習問題の作業はテキストの P38 まで全く同じです。

P39 以降は以下のコマンド(①～⑧)を入力しましょう。

- ① `sudo yum update`
- ② `sudo yum -y localinstall jdk-16.0.2_linux-x64_bin.`
- ③ `sudo alternatives --config java` (※)

※③のコマンドについて

[alternatives]コマンドは、複数の JDK をインストールしている場合にバージョンを切り替えるためのコマンドです。

実行後に以下のように表示が行われるため [1] を入力して実行します。

```

3.112.52.98 - ec2-user@ip-10-0-20-127 ~ - VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
Installed:
  jdk-16.0.2.x86_64 2000:16.0.2-ga

Complete!
[ec2-user@ip-10-0-20-127 ~]$ sudo alternatives --config java

There is 1 program that provides 'java'.

  Selection    Command
  -----
*+ 1           /usr/java/jdk-16.0.2/bin/java

Enter to keep the current selection[+], or type selection number: 1
[ec2-user@ip-10-0-20-127 ~]$

```

- ④ `java -version`
- ⑤ `sudo rpm -Uvh oracle-instantclient19.3-basic-19.3.0.0.0-1.x86_64.rpm`
- ⑥ `sudo rpm -Uvh oracle-instantclient19.3-sqlplus-19.3.0.0.0-1.x86_64.rpm`
- ⑦ `sqlplus64`
- ⑧ `oracleadmin@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=[RDS のエンドポイント])(PORT=1521))(CONNECT_DATA=(SID=[DB 名])))`

★①～⑧のコマンドを入力し、以下の画像と同じ `sql` と表示されれば OK★

パスワードの入力後、SQL の入力可能な状態になれば RDS への接続は成功です。  
※ SQL の入力を終了する場合は `[exit]` とコマンドを実行します。

```
sql>
```

⑫ デプロイ

➤ SQL の登録

別フォルダで配布した SQL コマンドを実行し、SQL Developer にてデータの登録を行ってください。

- ・配布フォルダ名：SQL
- ・ファイル名       ：デプロイ用 SQL

➤EC2(WEB/AP)のセキュリティグループに追加

- ・セキュリティグループ名： DeployWebSecurityGroup-p-**YYYYMMDD**
- ・タイプ：カスタム TCP
- ・ポート範囲：7778
- ・ソース：[カスタム]  
              [0.0.0.0/0]

➤ 編集する jar ファイル

配置場所は以下のフォルダの中

- ・配布フォルダ名：【配布用】HTML,SQL,画像,jar
- ・ファイル名       ：student.jar

➤ EC2(WEB/AP)状に転送した後のコマンド

java -jar student.jar

⑬ 最終画面

以下の画面が表示されればデプロイ成功です。

生徒一覧

生徒検索：

学生ID検索：

生徒番号	名前	性別
1011	山田太郎	男
1012	高橋健一	男
1013	鈴木花子	女
1014	佐々木良子	女
1015	皆藤啓介	男
1016	源頼子	女
1017	端美路幸之助	男
1018	犬塚咲	女
1019	粕川平助	男

東京IT専門学校



## 7 静的 WEB ページのホスティング

以下の条件に沿って S3 を使用し、HTML ファイルを公開してください。

作成した S3 は、最後に必ず削除するようにしてください。

=====

○条件

① 作成する HTML ファイル

別フォルダで配布した index.html をコピーして作成してください。

→配布フォルダ名：【配布用】HTML,SQL,画像,jar

ファイル名 : index.html(※)

※本練習問題の第 4 章 EC2 を利用した Web ページの公開で使った HTML ファイルです。

② パケット

・パケット名：hosting-static-webpage-p-YYYYMMDD  
(YYYYMMDD は本日の日付)

・リージョン：アジアパシフィック(ソウル) ap-northeast-2

③ S3 のホスティング設定

・静的ウェブサイトホスティング：有効にする

・ホスティングタイプ：静的ウェブサイトをホストする

・インデックسدキュメント：index.html

・エラードキュメント：error.html

=====

## 8 スケーリングによる EC2 の自動復旧

以下の条件に沿って Auto Scaling グループの作成を行い、グループ内の EC2 が削除された場合に自動復旧させてください。  
作成した Auto Scaling グループは、最後に必ず削除してください。

=====

○条件

① 起動テンプレート作成

- ・起動テンプレート名：LaunchTemplate-p-**YYYYMMDD**  
(**YYYYMMDD** は本日の日付)
- ・Amazon マシンイメージ：Red Hat Enterprise Linux8(HVM),SSD Volume Type
- ・インスタンスタイプ：t2.micro

② キーペア作成

- ・名前：AutoScaling-p-**YYYYMMDD**
- ・キーペアのタイプ：[RSA]
- ・ファイル形式：pem

③ ネットワーク設定

- ・名前：Virtual Private Cloud (VPC)
- ・セキュリティグループ：VPC のデフォルトセキュリティグループ

④ Auto Scaling グループ作成

- ・Auto Scaling グループ名：AutoScalingGroup-p-**YYYYMMDD**
  - ・起動テンプレート：LaunchTemplate-p-**YYYYMMDD** (①のテンプレートを選択)
  - ・[設定の構成] 画面で、[インスタンスの購入オプション] と [ネットワーク] 欄で次のように設定を行い、[次へ] をクリック。
    - ① インスタンスの購入オプション：起動テンプレートに準拠する
    - ② VPC：デフォルトの VPC
    - ③ サブネット：デフォルトのサブネット W
  - ・[[詳細オプションを設定] 画面：[ロードバランサーがありません]  
そのほかの項目はデフォルト設定のまま。
  - ・[グループサイズとスケーリングポリシーを設定する] 画面：[グループサイズ] がいずれも 1 を選択。
- =====

## 9 CDN によるコンテンツ配信

以下の条件に沿って CloudFront を使用し、S3 バケットをオリジンとしたコンテンツ配信を行ってください。作成した S3 バケットは、最後に必ず削除してください。

=====

○条件

① S3 バケット作成

- ・バケット名：cloudfront-content-distribution-p-**YYYYMMDD**  
(**YYYYMMDD** は本日の日付)
- ・リージョン：[アジアパシフィック(ソウル) ap-northeast-2]
- ・パブリックアクセスをすべてブロック：[チェックを外す]
- ・現在の設定により、このバケットとバケット内のオブジェクトがパブリックになる可能性があることを承認します：[チェックをつける]
- ・その他の入力項目はデフォルト設定のままにしておく。

② S3 アップロードする画像

配布した別フォルダより、画像をアップロードしてください。

→配布フォルダ名：【配布用】HTML,SQL,画像,jar

ファイル名 : flower.jpg

③ 作成する HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>My CloudFront Test</title>
</head>
<body>
<p>My distribution content goes here.</p>
<p></p>
</body>
</html>
```

=====