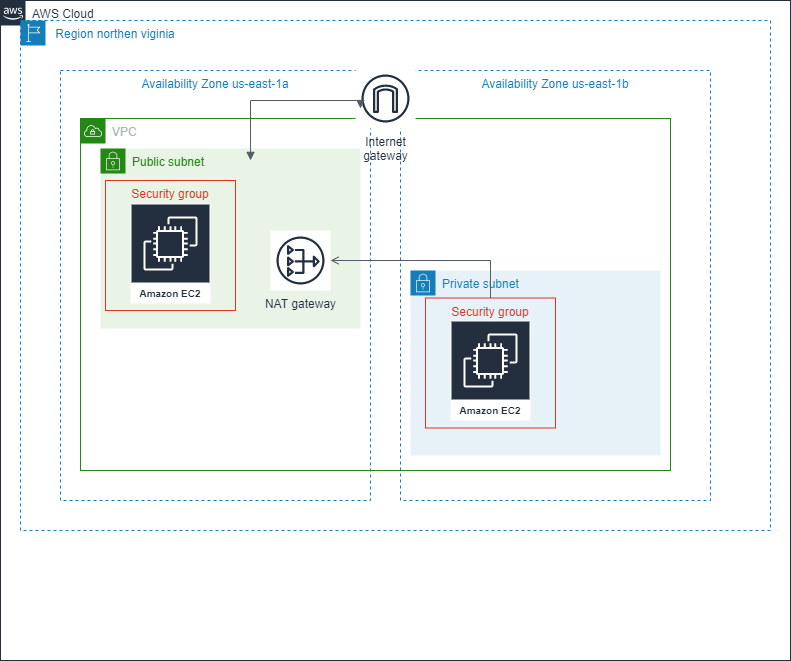
VPC構築（EC2も）

サンドボックスラボのStart Labをクリックした後、しばらくしてAWSをクリックしAWSコンソールを開く

　ラボの制限時間3時間

**■VPCの構築**

最終的な完成図



VPC設定1

1. VPCを作成しなさい．  
   VPCのみ  
   名前タグ：クラス名\_VPC  
   IPv4CIDRブロック：10.0.0.0/16
2. サブネットを作成しなさい．  
   ①パブリックサブネット  
   VPC：１.で作成したVPC（クラス名\_VPC）  
   サブネット名：クラス名\_PubA  
   アベイラビリティーゾーン：us-east-1a  
   IPv4 CIDR ブロック:10.0.0.0/24  
   新しいサブネットを追加をクリックし、さらにサブネット設定を追加  
   ②プライベートサブネット  
   VPC：１.で作成したVPC（クラス名\_VPC）  
   サブネット名：クラス名\_PriB  
   アベイラビリティーゾーン：us-east-1b  
   IPv4 CIDR ブロック:10.0.1.0/24  
   サブネットが作成されたら、サブネットクラス名\_PubAに設定を追加する（続く）  
   アクション→サブネットの設定を編集→  
    IP アドレスの自動割り当て設定→パブリックIPv4アドレスの自動割り当てを有効化にチェック  
    チェック後、をクリック
3. インターネットゲートウェイを作成し、VPCにアタッチする  
   名前：クラス名\_VPC\_IGW VPCを作成したら、クラス名\_VPCにアタッチ
4. ルートテーブルの作成し、サブネットの関連付け、ルートの追加を行う  
   名前：クラス名\_Pub\_rtb  
   VPC：クラス名\_VPC  
   作成後、作成したルートテーブルの詳細の「サブネットの関連付け」と「ルート」を編集  
   サブネットの関連付け：クラス名\_PubAにチェック  
   ルート：  
    送信先→0.0.0.0/0   
    ターゲット：インターネットゲートウェイ(最終的にクラス名\_VPC\_IGW)を選択

■EC2の設置  
①EC2（ec2\_PubA）

* 名前とタグ  
  名前：ec2\_PubA
* アプリケーションおよび OS イメージ (Amazon マシンイメージ)  
  Amazon マシンイメージ (AMI)：Amazon Linux 2 AMI(HVM) 64ビット(無料枠/デフォルト設定のまま）
* インスタンスタイプ   
  インスタンスタイプt2.micro（無料枠/デフォルト設定のまま）
* キーペアログイン  
  新しいキーペアの作成→キーペア名：クラス名日付（例：クラス名0906）  
   →キーペアのタイプ：RSA（デフォルト設定のまま）  
   →プライベートキーファイル形式：.pem（デフォルト設定のまま）
* ネットワーク設定  
  VPC: クラス名\_VPC  
  サブネット：クラス名\_PubA  
  パブリック IP の自動割り当て：有効化（デフォルト）  
  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：クラス名0906\_EC2  
  セキュリティグループルールを追加(SSHのルールは残す)し、  
  タイプ：HTTP、ソースタイプ：任意の場所、ソース：0.0.0.0/0  
  タイプ：すべてのICMPv4、ソースタイプ：カスタム、ソース：10.0.0.0/16
* ストレージを設定  
  デフォルト設定のまま
* 高度な詳細  
  デフォルト設定のまま

②EC2（ec2\_PriB）

* 名前とタグ  
  名前：ec2\_PriB
* アプリケーションおよび OS イメージ (Amazon マシンイメージ)  
  Amazon マシンイメージ (AMI)：Amazon Linux 2 AMI(HVM) 64ビット(無料枠/デフォルト設定のまま）
* インスタンスタイプ   
  インスタンスタイプt2.micro（無料枠/デフォルト設定のまま）
* キーペアログイン  
  クラス名0906.pemを選択
* ネットワーク設定  
  VPC: クラス名\_VPC  
  サブネット：クラス名\_PriB  
  パブリック IP の自動割り当て：無効化（デフォルト）  
  既存のセキュリティグループを選択する：  
   セキュリティグループ名：クラス名0906\_EC2
* ストレージを設定  
  デフォルト設定のまま
* 高度な詳細  
  デフォルト設定のまま

■確認1

EC2インスタンス（ec2\_PubA）の状態を確認する。

パブリックIPv4 アドレス・・・[3.208.27.182]  
プライベート IPv4 アドレス・・・[10.0.0.144]

EC2インスタンス（ec2\_PriB）の状態を確認する。  
 プライベート IPv4 アドレス・・・[10.0.1.171]

EC2インスタンスに接続し、接続確認を行う

①Tera Termを使用して、作成したEC2インスタンス（ec2\_PubA）に接続

　　IPアドレス：パブリックIPv4アドレス

　　ユーザ名：ec2-user パスワード：なし

　　秘密鍵：クラス名日付.pem(キーペアで作成したもの、例：クラス名0906) (次ページへ)

②接続したら、インターネットへの接続確認を行う 例：ping 8.8.8.8  
また、EC2インスタンス（ec2\_PriB）のプライベート IPv4 アドレスにpingする

[接続できた/できない] できない場合はセキュリティグループの設定確認

③EC2インスタンス（ec2\_PubA）からEC2インスタンス（ec2\_PriB）へSSH接続する  
ec2\_PubAへpemファイルをコピーする  
アクセス権の変更：chmod 400 クラス名日付.pem  
ec2\_PriBへSSH接続： ssh -i クラス名日付.pem ec2-user@プライベートIPv4アドレス(ec2\_PriB)

④接続したら、インターネットへの接続確認を行う 例：ping 8.8.8.8

接続できない→外部へのアクセスするための設定がないから

VPC設定２（NATゲートウェイの追加）

1. NATゲートウェイの作成  
   名前：クラス名\_VPC\_NAT   
   サブネット：クラス名\_PubA  
   接続タイプ：パブリック(デフォルト)  
   Elastic IP 割り当てID：割り当てるものがない場合、Elastic IPを割り当てをクリックしてから割り当てする
2. ルートテーブルの作成し、サブネットの関連付け、ルートの追加を行う  
   名前：クラス名\_Pri\_rtb  
   VPC：クラス名\_VPC  
   作成後、作成したルートテーブルの詳細の「サブネットの関連付け」と「ルート」を編集  
   サブネットの関連付け：クラス名\_PriBにチェック  
   ルート：  
    送信先→0.0.0.0/0   
    ターゲット：NATゲートウェイ(最終的にクラス名\_VPC\_NAT)を選択

■確認２

ec2\_PriBから、インターネットへの接続確認を行う 例：ping 8.8.8.8

接続できない場合は、NATの設定関連の見直し

完了できたら、ec2\_PriB→8.8.8.8へのping の状態をスクリーンショットとして、↓の空白に貼り付ける

ec2\_PriBからlogoutしておく End Labしておく

