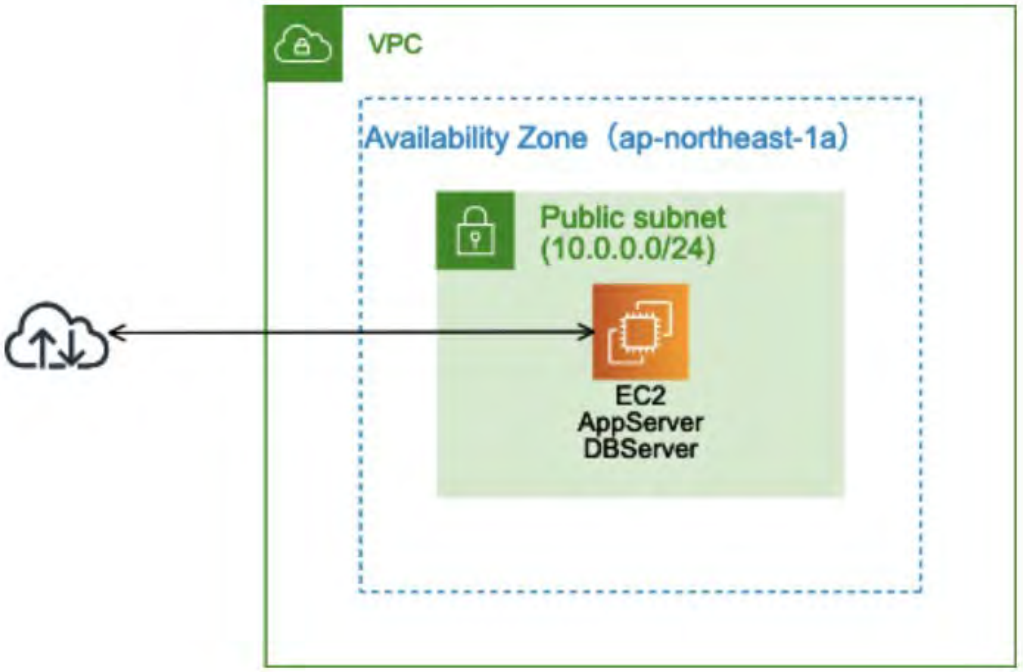
AWSハンズオン

参考サイト
https://aws.amazon.com/jp/getting-started/projects/scalable-wordpress-website/

[フェーズ1]
~サーバー 1 台構成で Redmine 環境を構築~



▼フェーズ 1-1: コンソールへのログインと VPC（ネットワーク）の作成

ステップ 1-1-1: AWS マネジメントコンソールにログインする

1. アカウント、ユーザー名、パスワード等を入力して、AWSマネジメントコンソールにログインします。

ログイン方法は利用するアカウント種類によって異なります。

IAM アカウントを御利用の場合

- 会社等で、IAM アカウントをあらかじめ準備されているケース
- 事前にログイン情報が記載された csv ファイル (user1.csv等) を確認してください。
- そのファイルに、ログイン用の URL、User Name、パスワードが記載されていますので、それに従ってログインしてください。

AWS のルートアカウント（個人アカウント）をご利用の場合

- https://console.aws.amazon.com にブラウザでアクセスしてください。
- アカウントの E メールアドレスとパスワードでログインしてください。

1. 左上部の「ホームに戻るボタン」をクリックします。
2. すべてのサービスを表示 をクリックします。

ステップ **1-1-2**: リージョンを変更する

1. 「リージョン」をクリックします。
2. 「アジアパシフィック (東京) 」を選択します。

ステップ 1-1-3: VPC 管理ページを開く

1. クリック画面上部の検索窓から「**VPC**」と入力します。
2. 「**VPC**」をクリックします。

ステップ 1-1-4: VPC の作成ウィザードを開始する

1. 「**VPC** ウィザードの起動」をクリックします。

ステップ **1-1-5**: VPC 作成ウィザード

1. 「**1** 個のパブリックサブネットを持つ **VPC**」をクリックします。
2. 「選択」をクリックします。

1. 「**10.0.0.0/16**」であることを確認します。
2. 「**handson-**自分のユーザー名」と入力します。
   例) handson-user1
3. 「**10.0.0.0/24**」であることを確認します。
4. 「**ap-northeast-1a**」であることを確認します。

VPC が作成されました。

1. 「**OK**」をクリックします。

以下の図の緑枠である「**VPC**」を作成しました。
これでサーバーを配置できるネットワークを作ったことになります。

ステップ **1-1-6**: VPC のフィルタリング設定

VPCでフィルタリングします。先ほど作成したVPCはすぐにはプルダウンメニューに含まれないため、一度画面をリロードする必要があります。

1. 画面をリロードする
2. 画面左上の「**VPC** でフィルタリング」のプルダウンメニューから先ほど作成した **VPC** を選択してください。
   ※他VPC と間違わないように注意してください。

ステップ **1-1-7**: 作成された VPC の確認

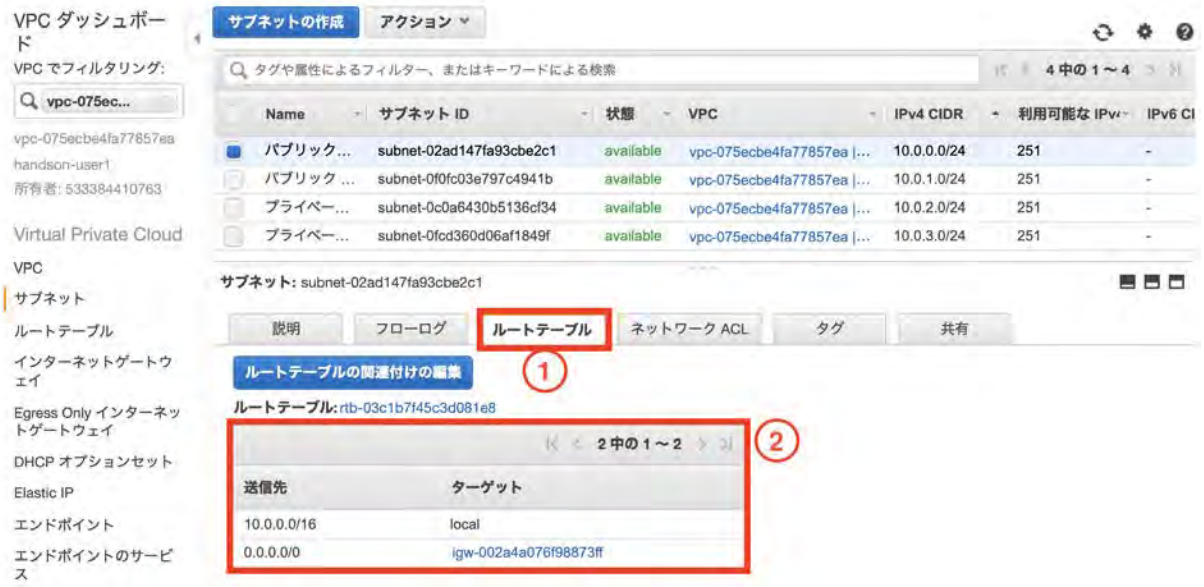
1. 「**VPC**」をクリックします。
2. 先ほど作成した **VPC** が存在するか（正しく絞り込めているか）を確認します。

3. 「**10.0.0.0/16**」であることを確認します。

ステップ **1-1-8**: ウィザードで作成されたサブネットを確認

1. 「サブネット」をクリックします。
2. サブネットを選択します。
3. 「**10.0.0.0/24**」であることを確認します。
4. 「**ap-northeast-1a**」であることを確認します。

ステップ 1-1-9: 作成されたサブネットの Route Table を確認



VPC のネットワークアドレス 10.0.0.0/16 のターゲットが local に、デフォルトルートの 0.0.0.0/0 のターゲットがインターネットゲートウェイ (igw-XXXX)になっており、インターネットと通信できる設定になっています。

1. 「ルートテーブル」をクリックします。
2. 内容を確認します。

確認したサブネットは図の緑色の領域のことです。


VPC
Availability Zone（ap-northeast-1a）
Public subnet
(10.0.0.0/24)

▼ フェーズ 1-2: サブネットを追加作成

ステップ **1-2-1**: サブネットを 3 つ追加作成

1. 「サブネットの作成」をクリックします。
2. VPC IDはフェーズ**1-1-5**で作成したものを選択してください。

1. 以下の表と図を参考にサブネットを3つ作成してください。（上スクリーンショットは表の1つ目のサブネット作成例）
2. 「新しいサブネットを追加」をクリックすることで2つ目、3つ目のサブネットも同時に作成することができます。

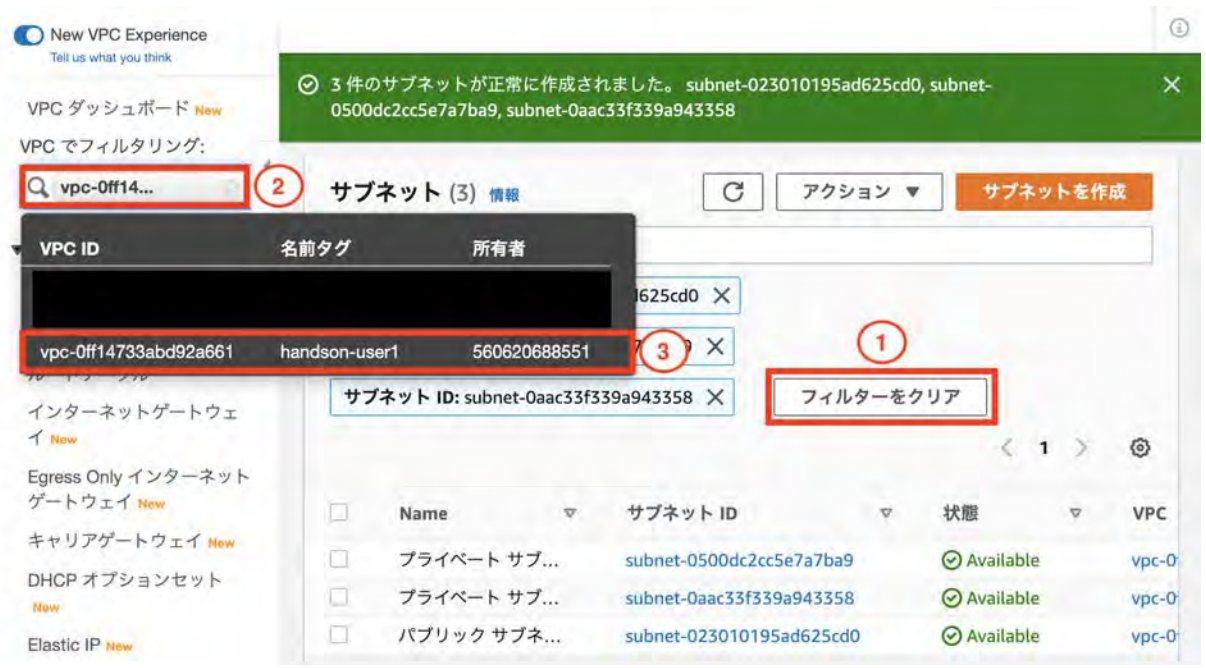
	サブネット名 （①）	アベイラビリティゾーン(②)	CIDRブロック (③)
1つ目	パブリックサブネットc	ap-northeast-1c	10.0.1.0/24
2つ目	プライベートサブネットa	ap-northeast-1a	10.0.2.0/24

3つ目	**プライベートサブネットc**	**ap-northeast-1c**	**10.0.3.0/24**

1. 3つとも入力後「サブネットを作成」ボタンをクリックします。

図の赤枠の部分を作成しました。

ステップ 1-2-2: 全てのサブネットを確認



サブネットが作成できたら、作成した3つのサブネットだけが表示されるフィルターがかかっているため、一度クリアしてVPCでフィルタリングをします。

1. 「フィルターをクリア」ボタンをクリックする。
2. 「**vpc**でフィルタリング」からフェーズ**1-1-5**で作成した**vpc**を選択してください。

ウィザードで作成したサブネットと追加したサブネットを確認します。
パブリックサブネットが2、プライベートサブネットが2,
ap-northeast-1aアベイラビリティゾーンが2、**ap-northeast-1c**アベイラビリティゾーンが2
作成しているいことを確認します。

ステップ **1-2-3**: パブリックサブネットのルートテーブルを変更

1. サブネットの検索窓で「**10.0.1.0**」と入力し、フィルタをかけます。

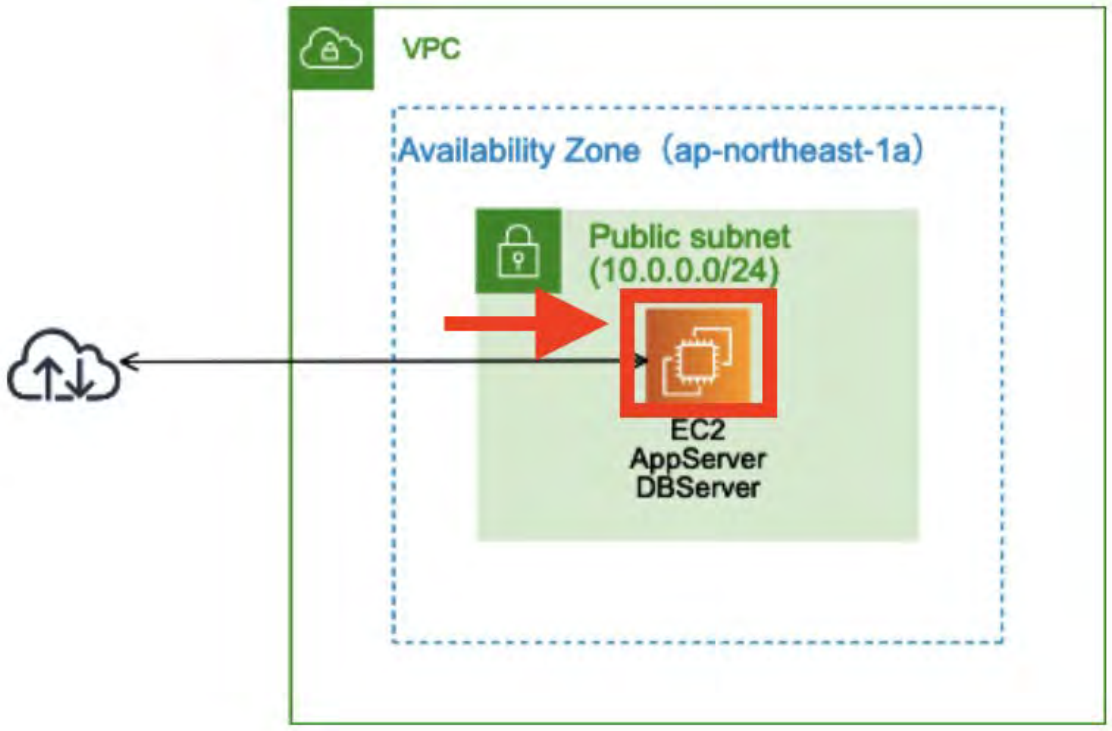
追加したサブネット「**10.0.1.0**」を実際にインターネットと通信できるように、ルートテーブルの割り当てを変更します。

1. 「**10.0.1.0/24**」のサブネットをクリックします。
2. 「ルートテーブル」をクリックします。
3. 「ルートテーブルの関連付けの編集」を クリックします。

1. これまでと異なるものを選択してくだい。(メインルートテーブルではない方)
   ※この VPC にはルートテーブルが 2 つしかありません
2. 「**0.0.0.0/0**」が表示されていることを確認します。
3. 「保存」をクリックします。

▼フェーズ 1-3: Amazon EC2 インスタンスの作成

図のオレンジの部分を作成します。
インスタンスとはAWSクラウドにある仮想サーバーのことです。



ステップ 1-3-1: EC2 管理ページを開く

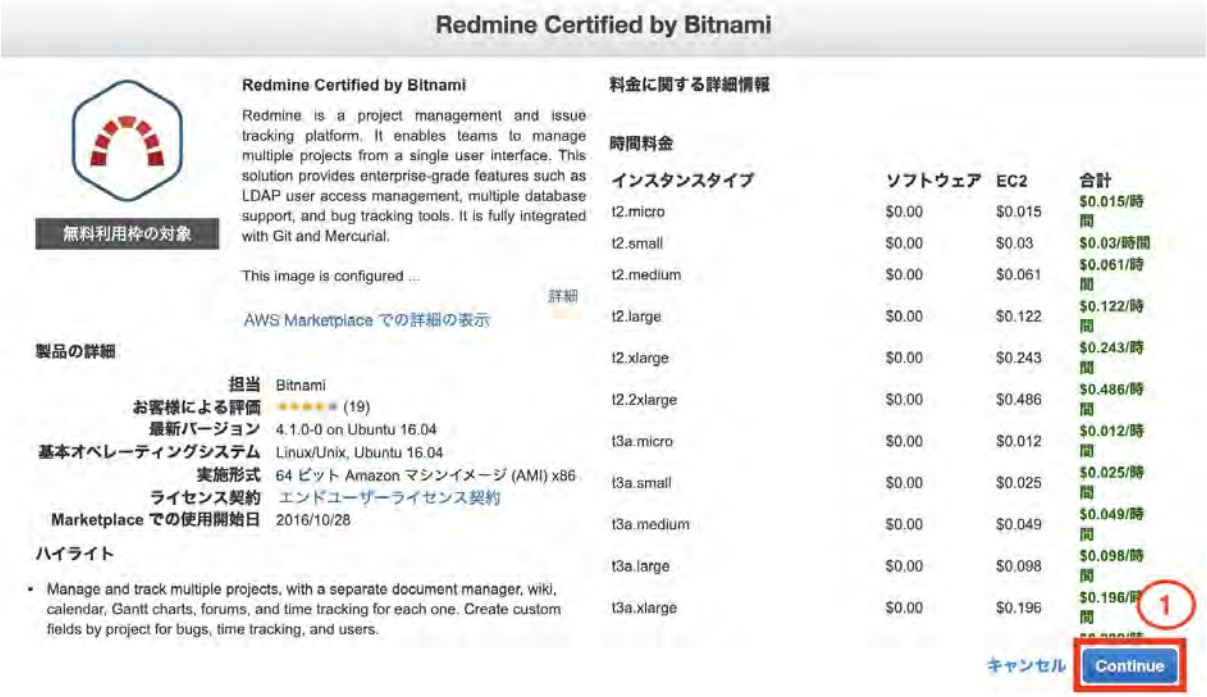
1. 「**EC2**」を入力します。
2. 「**EC2**」をクリックします。

ステップ **1-3-2**: EC2 インスタンスの作成

Web サーバーの作成を行います。

1. 「インスタンス」をクリックします。
2. 「インスタンスを起動」をクリックします。

1. 「AWS Marketplace」をクリックします。
2. 「redmine」と入力しエンターを押す。
3. 「Redmine Certified by Bitnami」を選択します。



1. 「Continue」をクリックします。

1. 「**t3.micro**」を選択します。
2. 「次のステップ: インスタンスの詳細の設定」をクリックします。

インスタンスの詳細設定を行います。VPC を選択するところでは、フェーズ**1-1-5**で作成した **VPC** を選択してください。

1. フェーズ**1-1-5**で作成した **VPC** を選択します。
2. 「**10.0.0.0/24 | パブリックサブネット | ap-northeast-1a**」を選択します。
   ※ プライベートサブネットと間違えないこと
3. 「有効」を選択します。
4. 「次のステップ: ストレージの追加」をクリックします。

ストレージは変更せずに、次に進みます。

1. 「次のステップ: タグの追加」をクリックします。

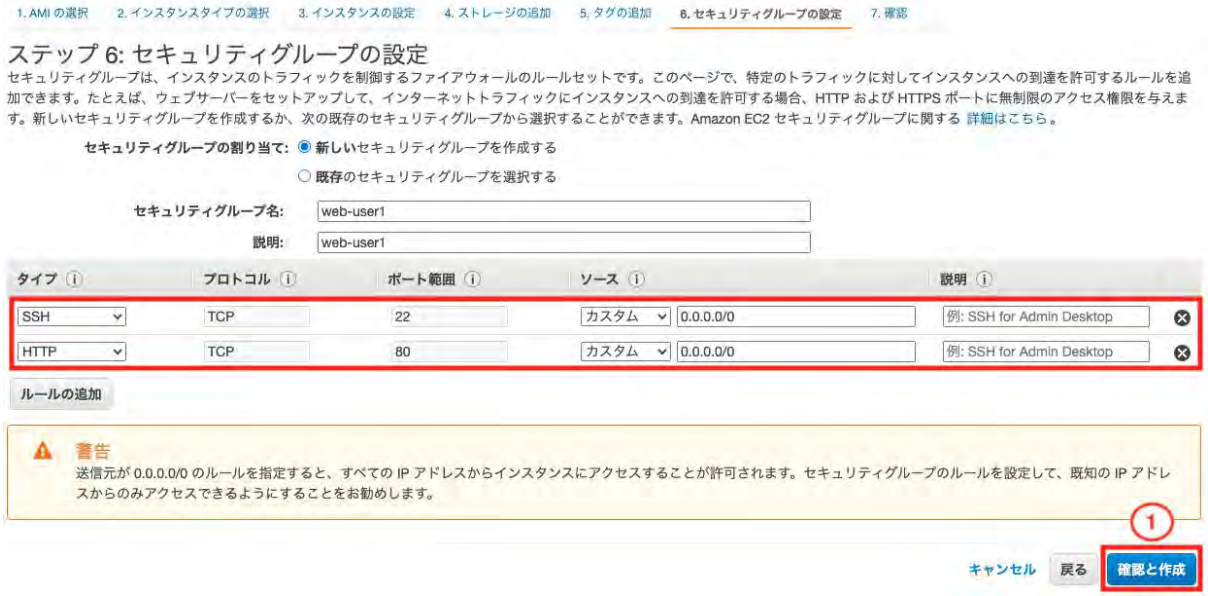
インスタンスを区別できるようにタグに名前を設定します。(webserver#1-user1 等ユーザー名を付けます。)

1. 「タグの追加」をクリックします。

2. キーに「**Name**」と入力します。
3. 「webserver#1- ユーザー名」とします。
   例）[webserver#1-user1]
4. 「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。

「新しいセキュリティグループを作成する」を選択します。

1. 「新しいセキュリティグループを作成する」を選択します。
2. セキュリティグループ名は **web-**ユーザー名としてください。説明にも同じ値を入力します。
   例) web-user1
3. タイプが「**HTTPS**」のルールを削除する。



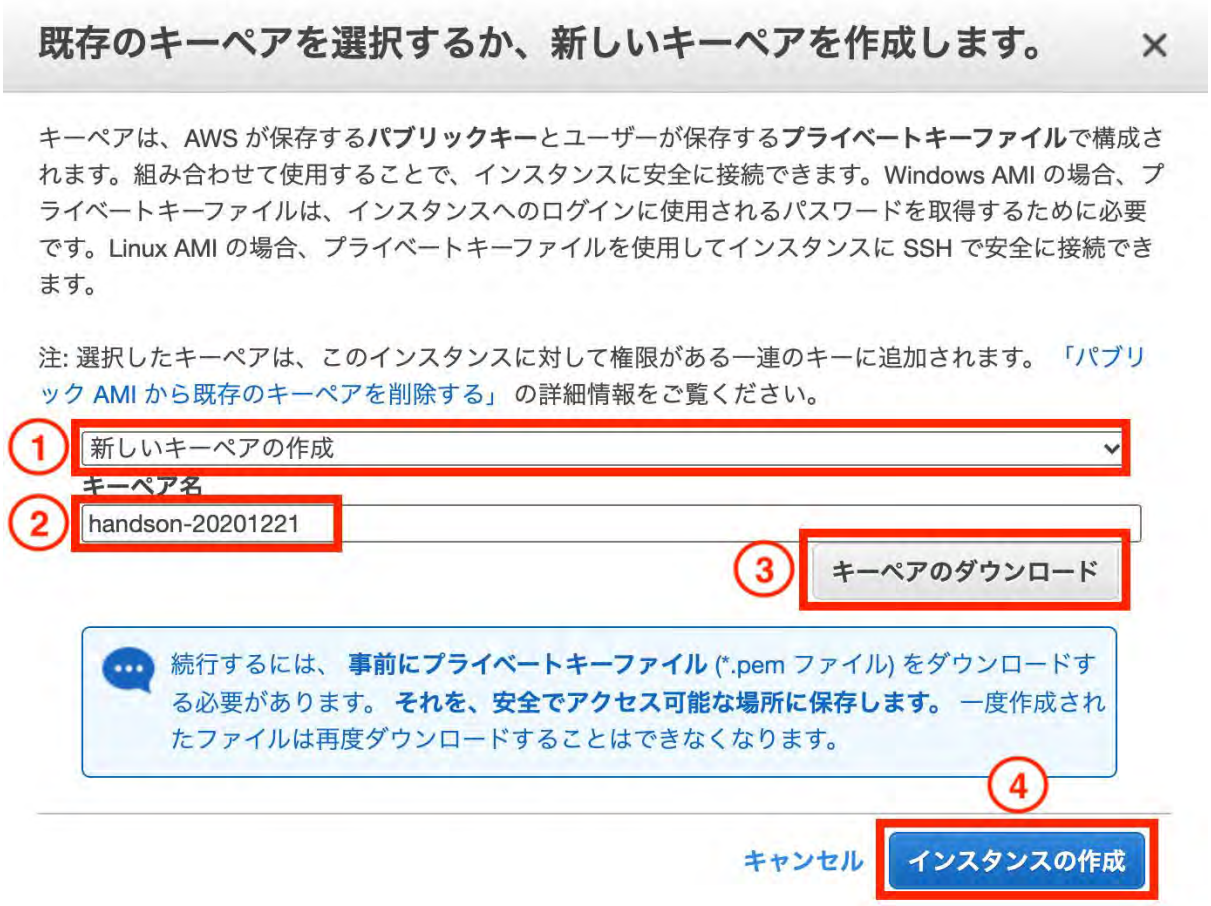
ルールタイプがsshとhttpの2つ設定されていることを確認し、確認と作成ボタンをクリックします。「新しいセキュリティグループを作成する」を選択します。

1. 「確認と作成」をクリックする。

画面を下にスクロールさせて設定内容を確認してから作成します。

1. 「起動」をクリックします。

ステップ 1-3-3: キーペアを選択する



新しいキーペアを作成します。

1. 「新しいキーペアの作成」を選択します。
2. 「**handson-**今日の日付」と入力します。
   例) handson-2021xxxx
3. 「キーペアのダウンロード」をクリックして、ファイルをダウンロードします。ダウンロードしたファイルは今後の作業で使うため大事に保管します。
4. 「インスタンスの作成」をクリックします。

EC2 インスタンスが作成されました。

1. 「インスタンスの表示」をクリックします。

ステップ 1-3-4: 作成した EC2 インスタンスを確認

ユーザー名等で絞込を行うと便利です。インスタンス作成完了には数分かかります。

1. ユーザー名を入れてリターンを押すことで表示を絞り込むことができます。
   例）user1

▼フェーズ 1-4: Elastic IP（固定 IP）の割り当て

「サービス」→「**ec2**」の画面を表示します。

ステップ **1-4-1**: Elastic IP (EIP) を取得

1. 「**Elastic IP**」をクリックします。
2. 「**Elastic IP** アドレスの割り当て」をクリックします。

1. 「割り当て」をクリックします。

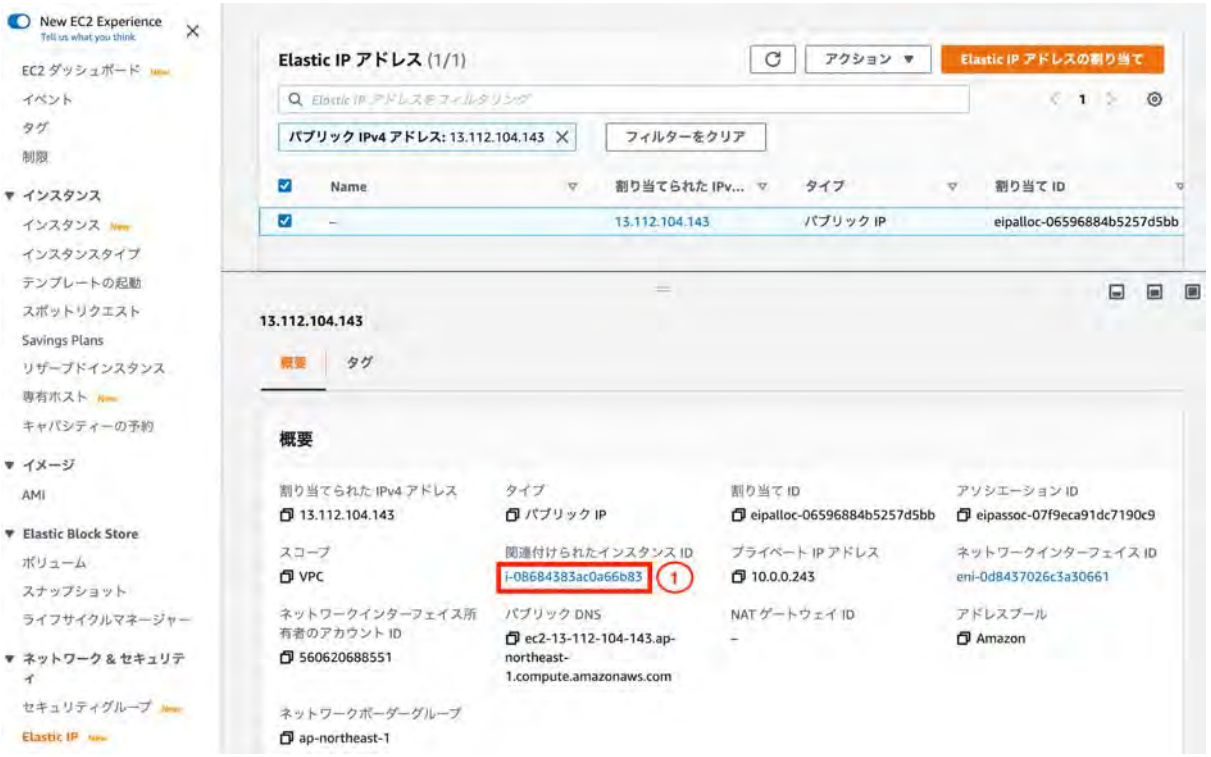
ステップ **1-4-2**: Elastic IP (EIP) をインスタンスに紐付け

先ほど割り当てられたEIPをインスタンスに関連付けます。

1. 「この**Elastic IP**アドレスを関連付ける」をクリックします。

取得した EIP を EC2 インスタンスに紐付けます。フェーズ**1-3** で作成した EC2 インスタンスを選択してください。

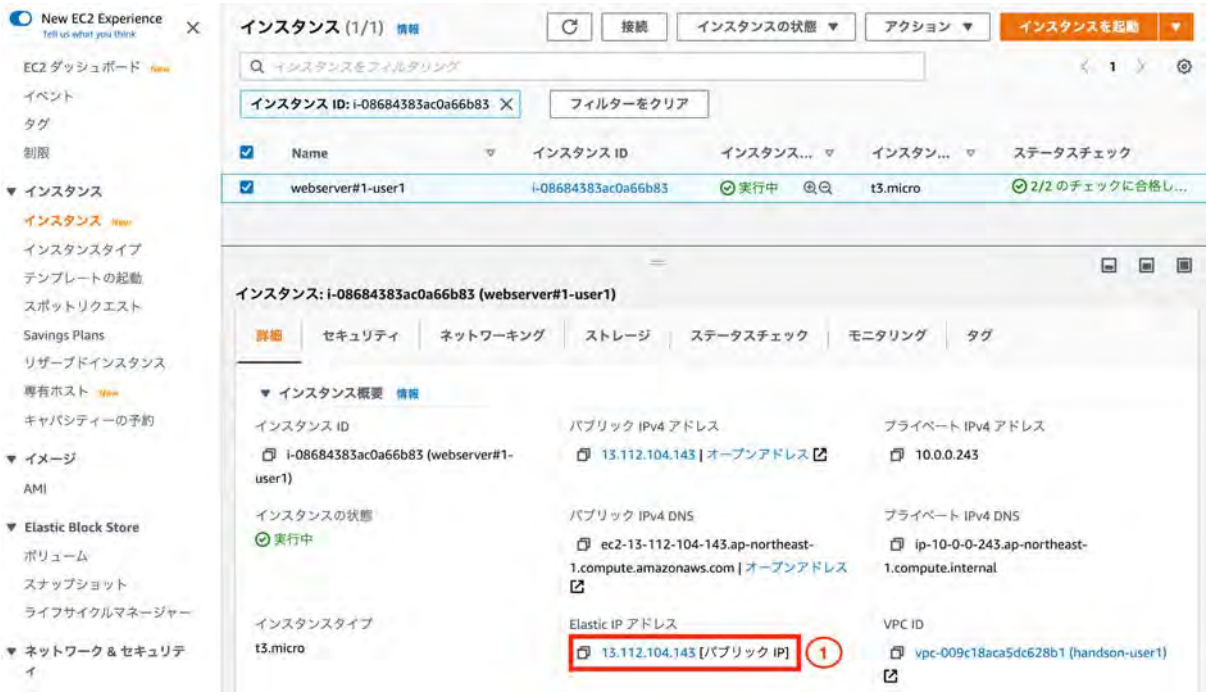
1. クリックすると候補が表示されます 自分の名前（例. user1）等を入力しフェーズ**1-3** で作成した EC2 インスタンスを選択してください。
   例）[webserver#1-user1]等
2. 「関連付ける」をクリックします。



正しくEC2 インスタンス紐付けられていることを確認します。EIP は後で使用するため、メモしておきます。

1. 関連付けられたインスタンスIDをクリックし、フェーズ**1-3** で作成した EC2 インスタンス画面に移動することを確認します。

ステップ **1-4-3**: Elastic IPアドレスをメモ



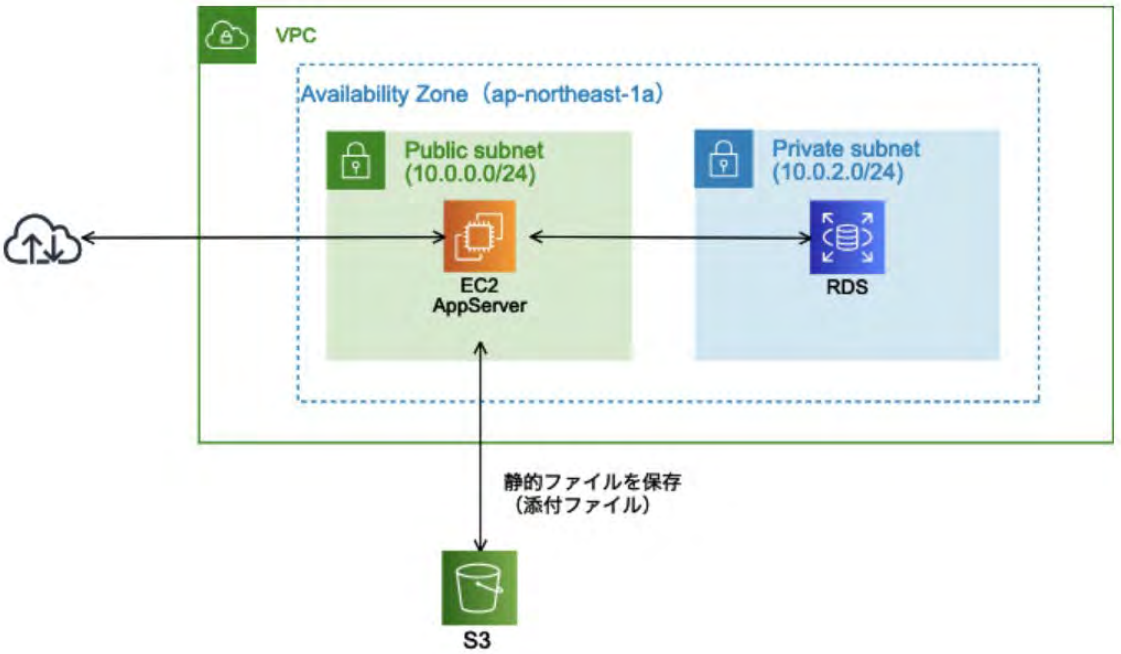
1. 作成したインスタンスを選択して、「**Elastic IP**アドレス」をメモする。

ステップ **1-4-4**: Redmineにアクセス

先ほどメモした EIP にアクセスし、redmineが表示されることを確認します。

1. ブラウザで**http://<Elastic IPアドレス>/** にアクセスします。
2. redmineが表示されることを確認します。

[フェーズ 2]
~拡張性を向上しつつDB 運用負荷を軽減する構成を構築~



▼フェーズ 2-1: Amazon RDS のセキュリティグループを作成

ステップ **2-1-1**: DB 用セキュリティグループを作成

1. 「**ec2**」を入力します。

2. 「**EC2**」をクリックします。

1. 「セキュリティグループ」をクリックします。
2. 「セキュリティグループの作成」をクリックします。

1. 「**db-**ユーザー名」を入力します。 例）db-user1

2. 「**RDS for MySQL**」など説明を入力します。
3. フェーズ**1-1-5** で作成した**VPC** を選択してください。 例）handson-user1 を選択
4. 「ルールの追加」をクリックします。
   ※ インバウンドルールであることを確認

1. 「**MySQL/Aurora**」を選択します。
2. 「カスタム」を選択します。
3. 「**Web**」と入力して候補を表示させます。
   Web と入力しても補完されない場合には、該当するセキュリティグループの ID (sg-xxxxxx) を入力します。
4. 「候補」をクリックします。

5. 「セキュリティグループを作成」をクリックします。

▼フェーズ 2-2: DB サブネットグループを作成

ステップ2-2-1: Amazon RDS 管理ページを開く

1. 「**rds**」を入力します。
2. 「**RDS**」をクリックします。

ステップ 2-2-2: DB サブネットグループを作成

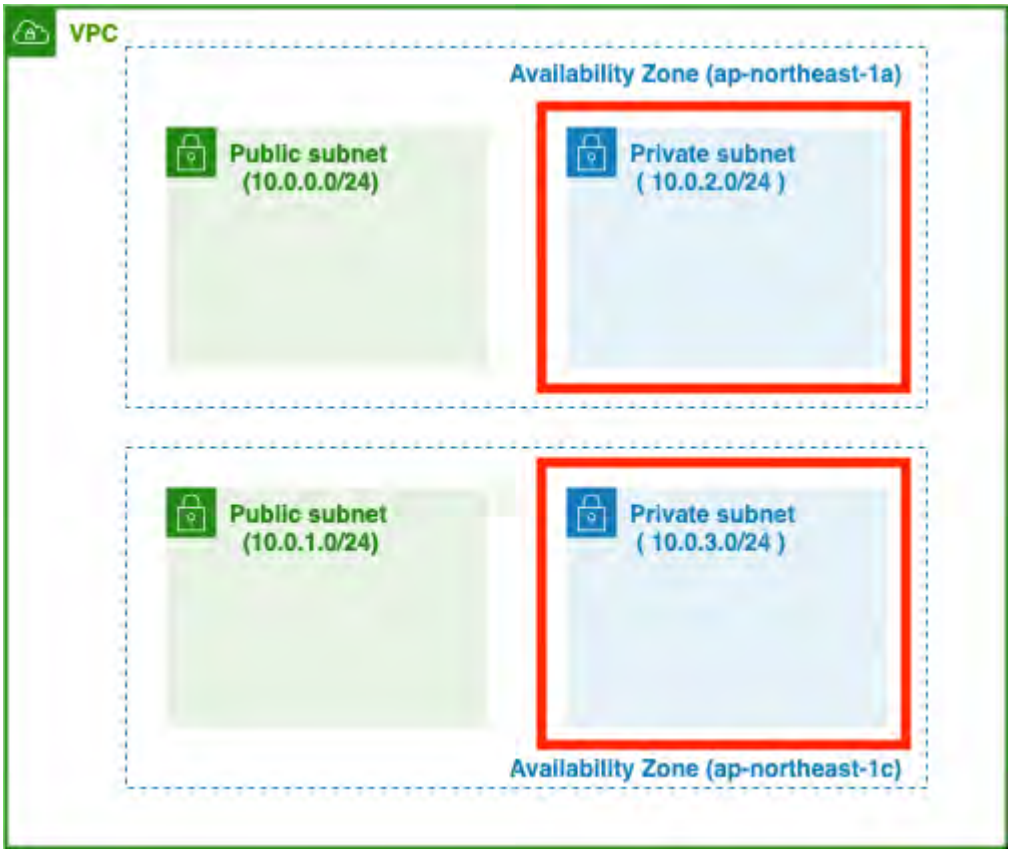
プライベートサブネット内に DB サブネットグループを作成します。

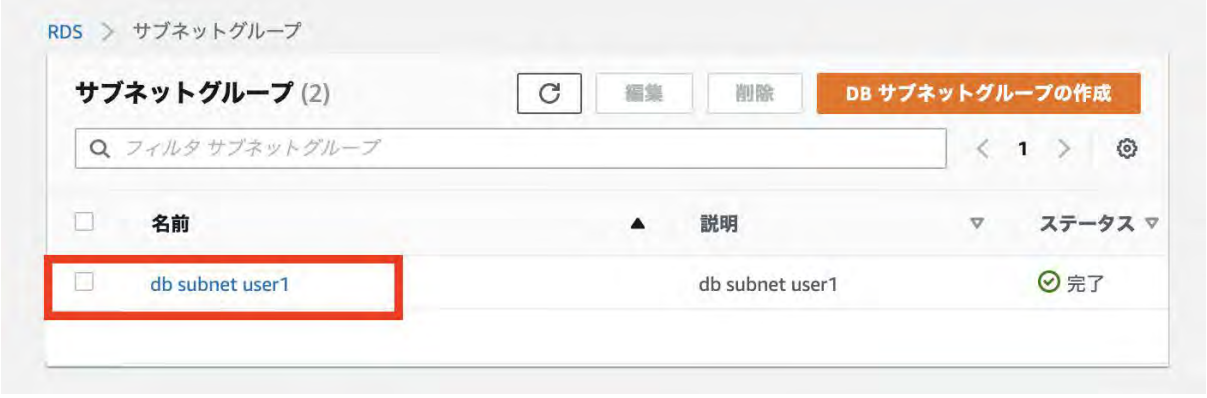
1. 「サブネットグループ」をクリックします。
2. 「**DB** サブネットグループの作成」をクリックします。

1. 「**db subnet** ユーザー名」を入力します。 例）db subnet user1
2. 「**RDS for MySQL**」などと入力します。
3. フェーズ**1-1-5** で作成した **VPC**を選択します。 例）[handson-user1]

ap-northeast-1a のプライベートサブネット (10.0.2.0/24) とap-northeast-1cのプライベートサブネット (10.0.3.0/24)を追加します。

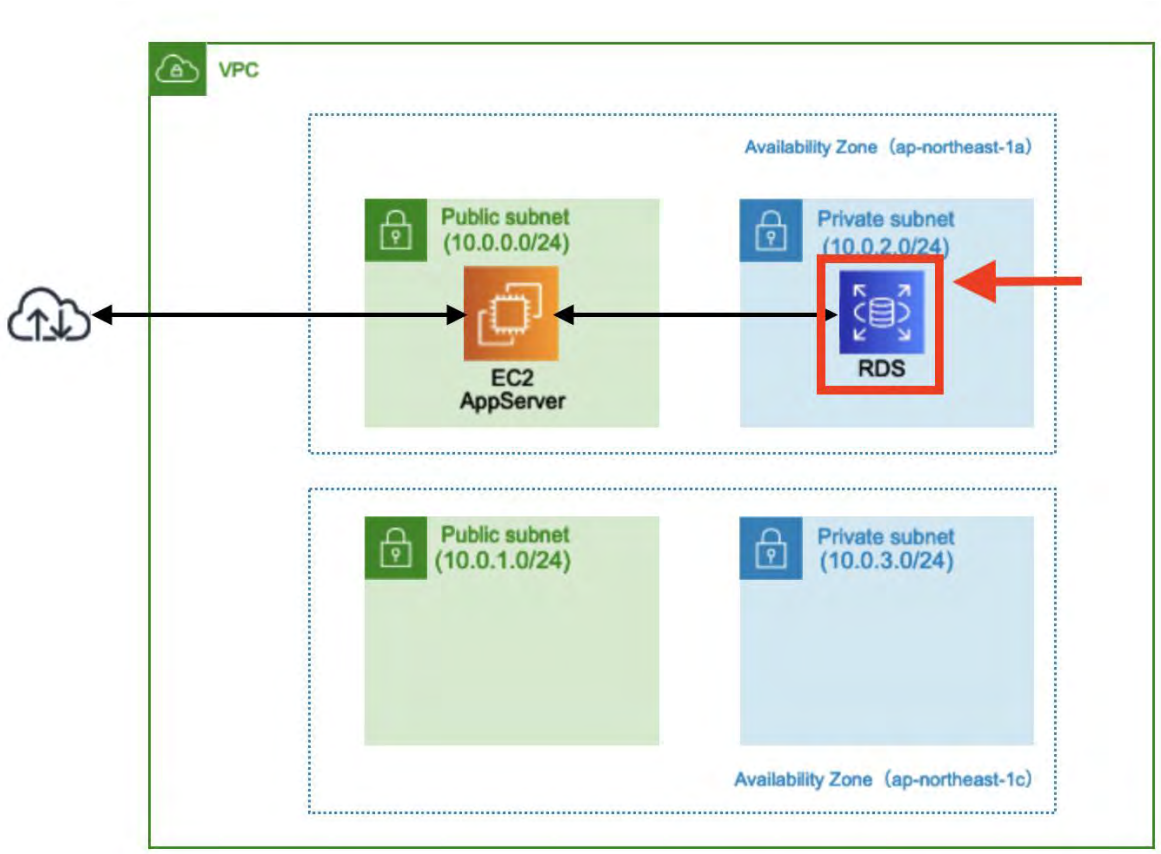
1. 「**ap-northeast-1a**」、「**ap-northeast-1c**」を選択します。
2. 「プライベートサブネット**(10.0.2.0/24)**」、「プライベートサブネット**(10.0.3.0/24)**」を選択します。
3. 「作成」をクリックします。



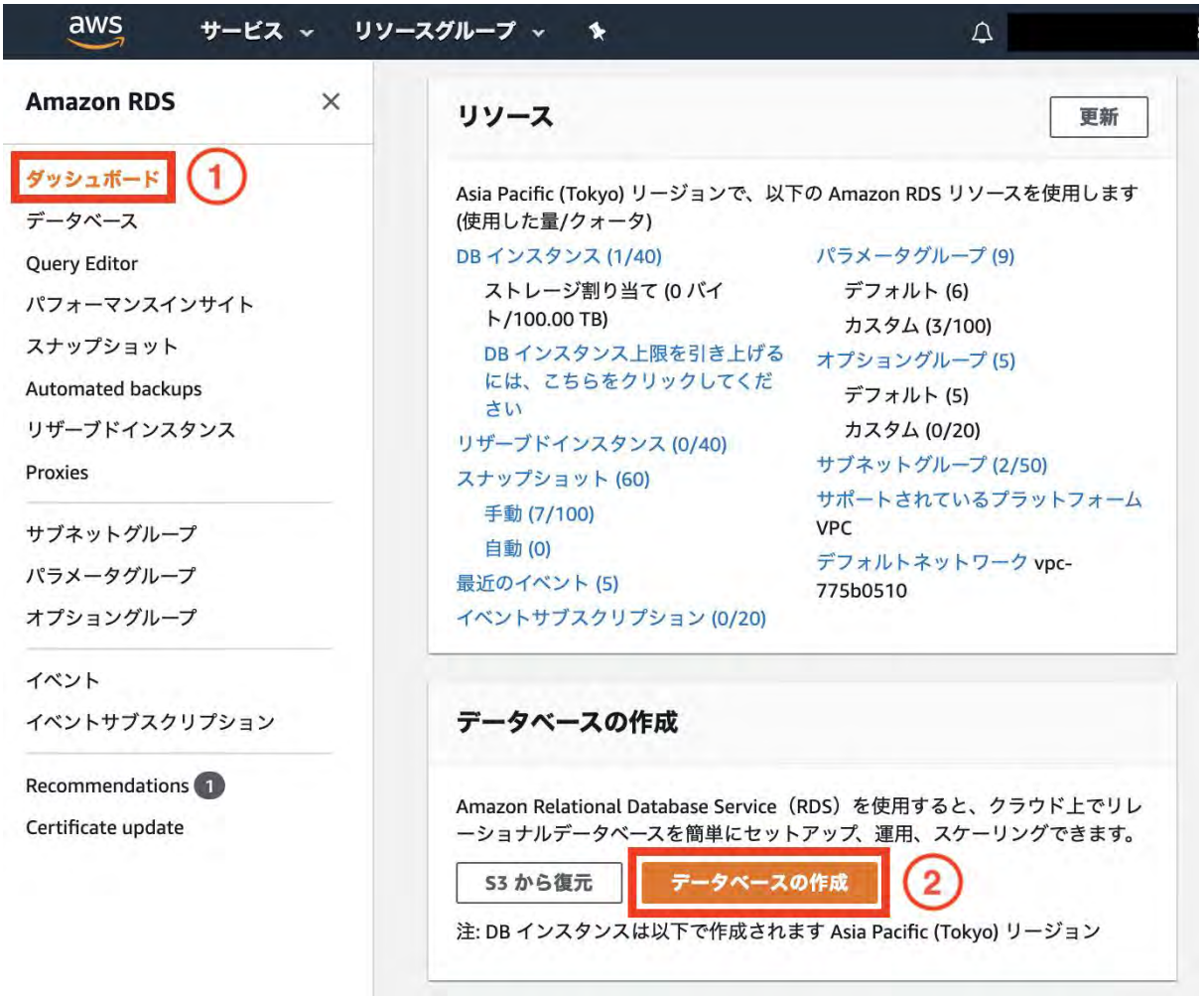


DBサブネットが作成されました。

▼フェーズ 2-3: Amazon RDS インスタンスを作成



ステップ **2-3-1**: データベースの作成



1. 「ダッシュボード」をクリックします。
2. 「データベースの作成」をクリックします。

[エンジンのオプション]

1. 「**MySQL**」を選択します。

[テンプレート]

1. 「開発/テスト」を選択します。

[設定]

DBインスタンス識別子とパスワードは、**redmine-**自分の名前とします。

2. 「**redmine-**自分の名前」と入力します。 例）redmine-user1

3. 「**admin**」と入力します。
4. admin のパスワード「**redmine-xxxx**」(xxxxはユーザー名など任意の文字列)を入力します。
   例）redmine-user1
5. 再度パスワードを入力します。

[DBインスタンスサイズ]

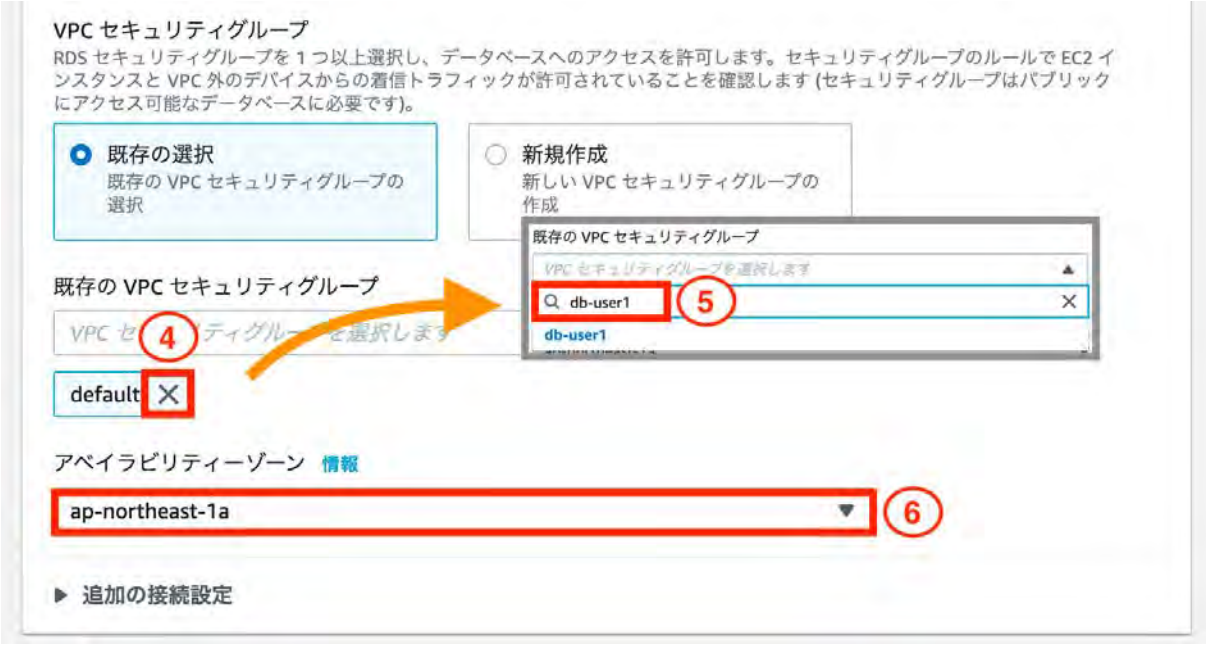
1. 「バースト可能クラス（tクラスを含む）」をクリックします。
2. 「**db.t3.micro**」を選択します。

[可用性と耐久性]

3. 「スタンバイインスタンスを作成しないでください」を選択します。

[接続]

1. フェーズ**1-1-5**で作成した**VPC**を選択します。例）handson-user1
2. 自動的に RDS サブネットグループが選択されます。
3. 「なし」を選択します。



4. 既存のVPCセキュリティグループでdefaultが選択されている場合は、「×」で外します。
5. 「ステップ 1: DB 用セキュリティグループを作成」で作成したセキュリティグループを選択します。 例）db-user1
6. 「**ap-northeast-1a**」を選択します。

[追加設定]

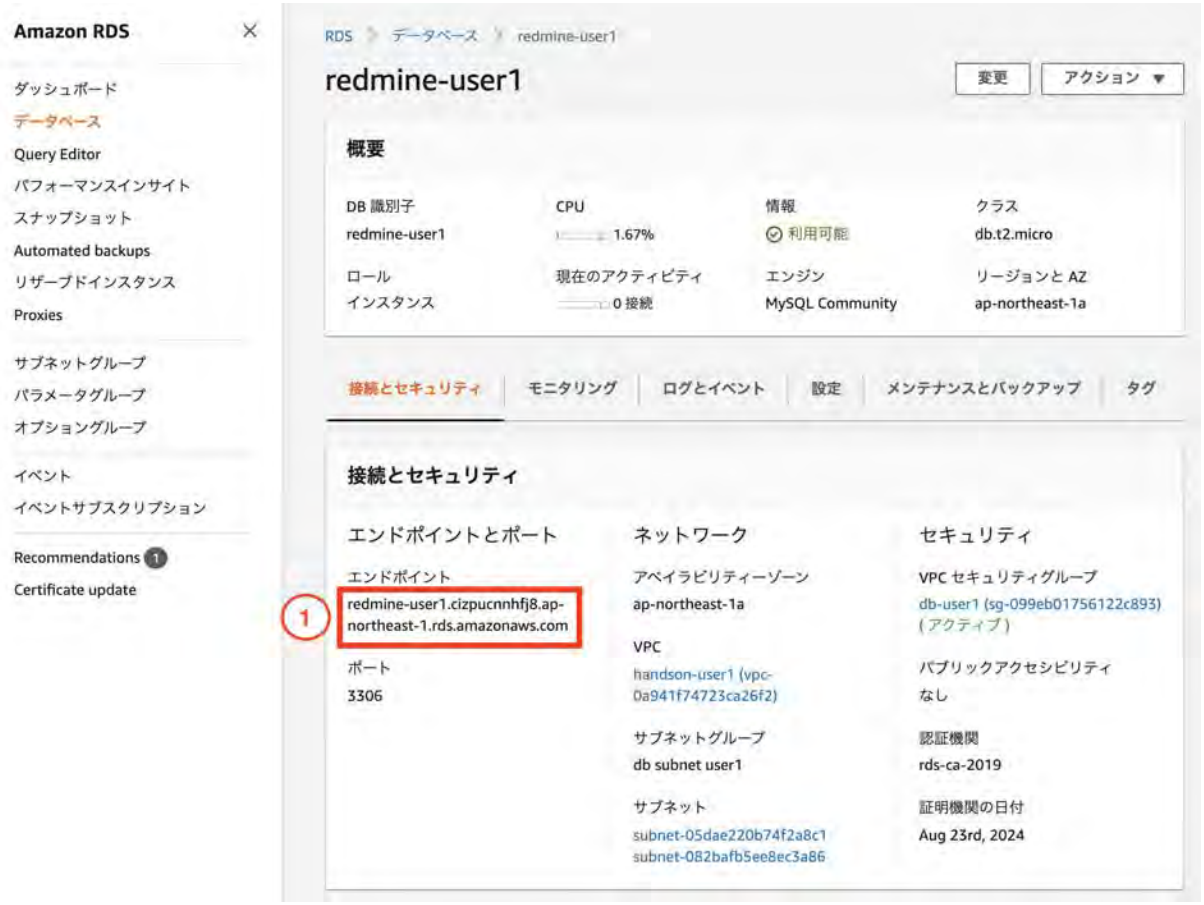
1. 「追加設定」はデフォルトでは折りたたまれているため、クリックして設定を表示します。
2. バックアップ保持期間で「**0**日間」を選択します。
3. 「データベースの作成」をクリックします。

▼フェーズ 2-4: RDSに接続

ステップ **2-4-1**: 作成した RDS インスタンスを確認

「サービス」→「**RDS**」画面を表示します。

1. 「データベース」をクリックします。
2. フェーズ**2-3-1**で作成した RDS インスタンスをクリックします。



RDS の各インスタンスにはエンドポイント（Endpoint）と呼ばれるホスト名が設定されます。エンドポイントをメモします。

表示されない場合は画面をリロードしてください。
※ 作成されるまで時間がかかります

1. エンドポイントをメモします。

ステップ 2-4-2: database.ymlをバックアップ

作成したインスタンスに接続します。

今回はsshで接続するため手元のターミナルを開きます。

1. EC2インスタンス作成時にダウンロードした「プライベートキーファイル（handson-2021xxxx.pem）」があるか確認する

```
ls ~/Downloads/handson-2021xxxx.pem
```

2. プライベートキーファイル（handson-2021xxxx.pem）をわかりやすいところに移動させる

```
mv ~/Downloads/handson-2021xxxx.pem [ディレクトリ名]/handson-2021xxxx.pem
```

3. 以下のコマンドを実行して権限を変更する

```
cd [ ディレクトリ ]
chmod 400 handson-2021xxxx.pem
```

4. インスタンス作成後にメモしたEIPを使用してインスタンスに接続する
(yes と入力)

```
ssh -i "handson-2021xxxx.pem" bitnami@[ webserver#1のElastic IPアドレス ]
```

接続ができたら、以下のコマンドを実行してMySQLのdatabase.ymlをバックアップする。

```
# root
sudo su
# redmineのディレクトリに移動
cd /opt/bitnami/apps/redmine/htdocs/
# database.ymlをバックアップ
cp config/database.yml config/database_bk.yml
```

ステップ 2-4-3: RDSに接続

引き続きターミナルで作業します。

以下のコマンドを実行してdatabase.ymlの以下の箇所を編集します。

```
# database.ymlを編集
vi config/database.yml
```

```
production:
  adapter: mysql2
  database: rds_redmine
  host: [メモしたRDS Endpoint]
  username: admin
  password: [redmine-自分の名前(例:redmine-user1)]
  encoding: utf8
```

以下のコマンドを実行します。
databaseを作成、マイグレーションをします。

```
#databaseを作成
bundle exec rake db:create RAILS_ENV=production
#マイグレーション
bundle exec rake db:migrate RAILS_ENV=production
```

設定が終了したらApacheを再起動して設定を反映させます。

```
#apacheの停止
/opt/bitnami/apache2/scripts/ctl.sh stop
#apacheの起動
```

```
/opt/bitnami/apache2/scripts/ctl.sh start
#apacheのステータス確認
/opt/bitnami/apache2/scripts/ctl.sh status
```

以下のコマンドを実行し、mysqlを停止します。
mysql停止後も**redmine**にアクセスできることを確認してください。

```
/opt/bitnami/mysql/scripts/ctl.sh stop
```

▼フェーズ 2-5: Redmine S3対応

ステップ 2-5-1: Redmineにダミーデータを登録

引き続きインスタンスに接続してコマンドを実行し、ダミーデータを登録します。
以下のコマンドを実行する事でredmineにダミーのデータが登録され、動作検証がスムーズに行えます。

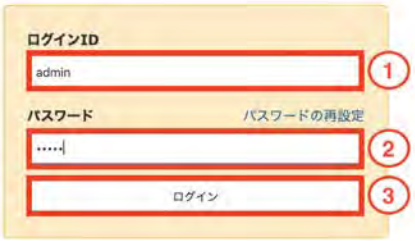
1. 以下のコマンドを実行します。

```
RAILS_ENV=production bundle exec rake db:fixtures:load
```

ステップ 2-5-2: Redmineにファイルをアップロード

ブラウザで**http://<Elastic IPアドレス>/** にアクセスしてredmineを表示し、ファイルをアップロードします。

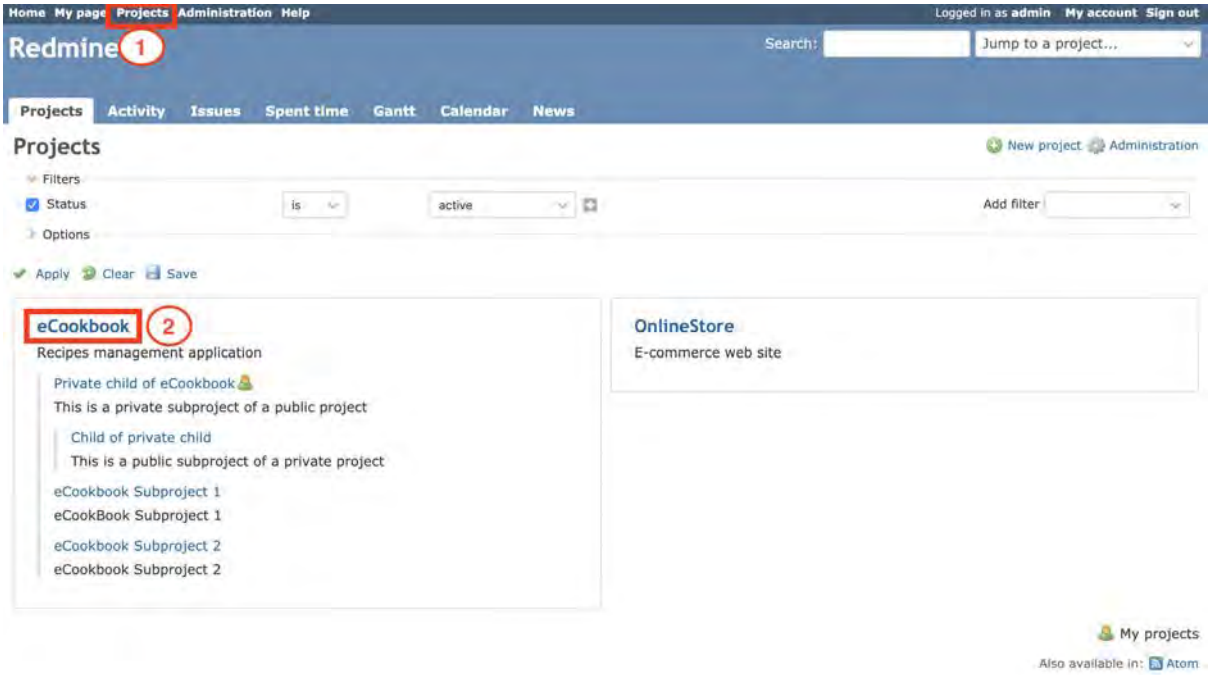
1. ブラウザで**http://<Elastic IPアドレス>/** にアクセスしてredmineを表示します。
2. Redmine画面右上の「ログイン」をクリックします。



先ほどfixtures:loadを実行しダミーのユーザが作成されているため、adminでログインします。

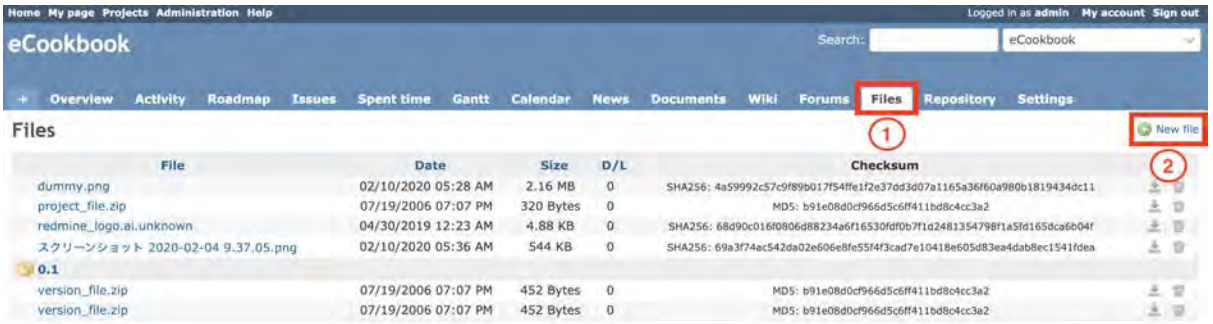
1. ログインIDに「**admin**」を入力します。
2. パスワードに「**admin**」を入力します。
3. ログインをクリックします。

Homeが表示されたらログイン成功です。

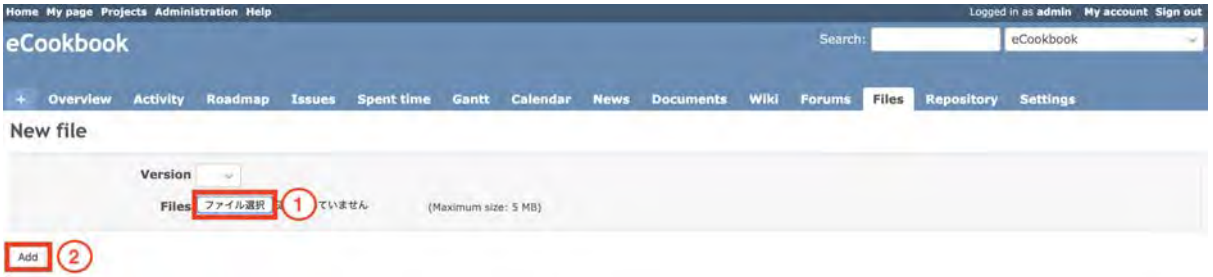


Redmineにログイン後、以下の手順でファイルをアップロードします。

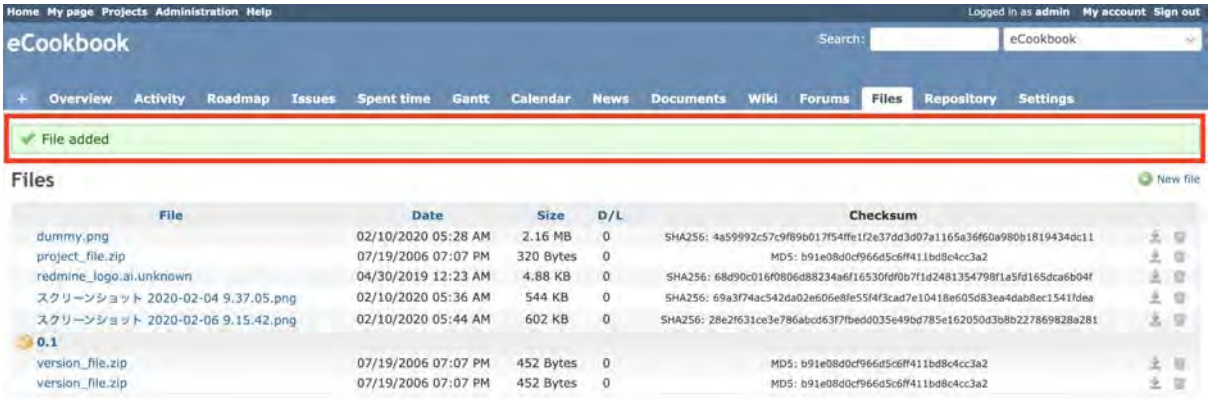
1. 「**projects**」をクリックします。
2. 「**eCookbook**」をクリックします。



1. 「**Files**」をクリックします。
2. 「**New file**」をクリックします。



1. 「ファイルを選択」をクリックして、アップロードするファイルを選択します。
2. 「**Add**」をクリックします。



redmineの画面に「**File added**」というアラートが表示されたら、ファイルのアップロードができています。

ステップ 2-5-3: ファイルのアップロード確認

ファイルがアップロードされているかをec2の中で確認します。

以下のコマンドを実行して、先ほどアップロードしたファイルがあることを確認します。

```
cd /opt/bitnami/apps/redmine/htdocs
# ファイルがあることを確認
ls files/20xx(年)/xx(月)/
→ xxxxxxxx.png などと表示されればOK
```

ステップ **2-5-4**: S3アクセス用のユーザーの作成

サービス検索窓でIAMを検索し選択します。

1. 「**iam**」を入力します。
2. 「**IAM**」をクリックします。

「ユーザーを追加」画面へ移動します。

1. 「ユーザー」をクリックします。

2. 「ユーザーを追加」をクリックします。

1. ユーザ名に「**s3access-2021xxxx**」と入力します。
2. 「プログラムによるアクセス」にチェックを入れます。
3. 「次のステップ: アクセス権限」をクリックします。

1. 「既存のポリシーを直接アタッチ」を選択します。
2. ポリシーのフィルタで「**AmazonS3FullAccess**」と入力して検索します。
3. 表示された「**AmazonS3FullAccess**」ポリシーにチェックを入れます。
4. 「次のステップ: タグ」をクリックします。

1. キーに「**Name**」を入力します。
2. 値に「**iam-**ユーザー名」を入力します。例）iam-user1
3. 「次のステップ: 確認」をクリックします。

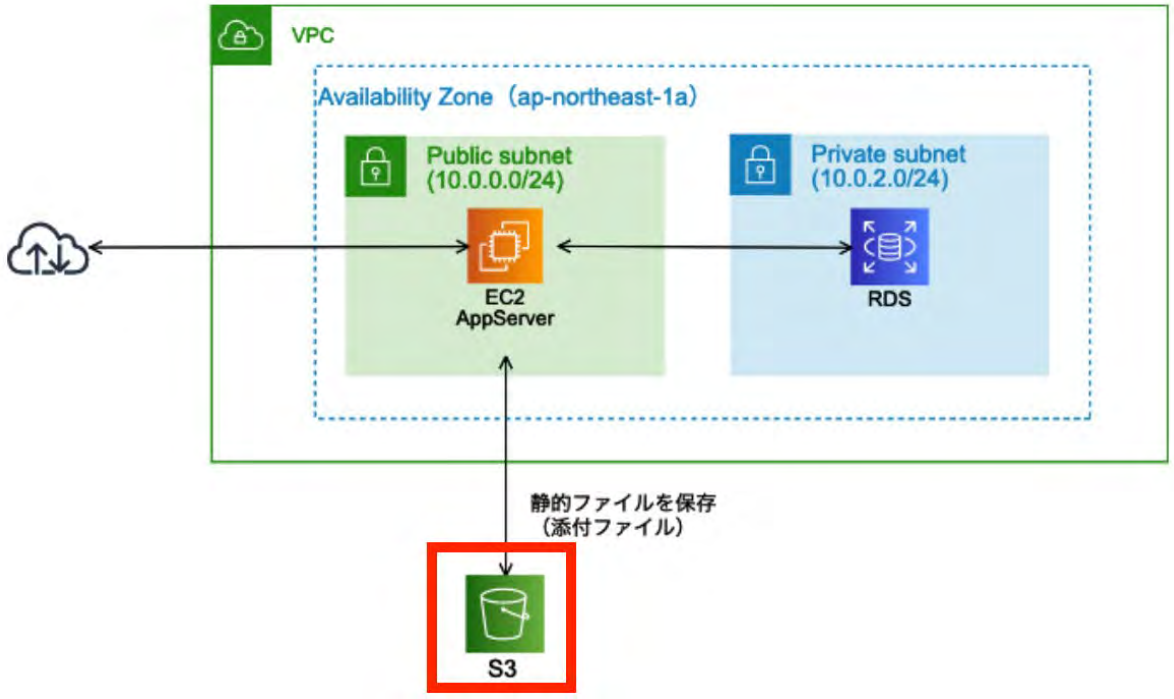
設定確認＆ユーザーの作成をします。

1. 設定内容を確認し「ユーザーの作成」をクリックします。

アクセスキーやシークレットアクセスキーが書かれているcsvをダウンロードする。
※ 後ほど使用するので大切に保管すること

1. 「**.csv**のダウンロード」をクリックします。
2. ダウンロードしたcsvは後ほど使用するため、大切に保管する。

ステップ **2-5-5**: S3 バケット作成



サービス検索窓でS3を検索し選択します。

1. 「**S3**」を入力します。
2. 「**S3**」をクリックします。

1. 「バケットを作成」をクリックします。

バケットを作成します。バケット名はグローバルで一意である必要があります。
xxxxは自分の名前等を入力し、他と被らないバケット名にしてください。

1. 「**redmine-xxxx(自分の名前)-2021xxxx**」と入力します。例) redmine-user1-20210210
2. 「アジアパシフィック（東京）」を選択します。
3. 「作成」をクリックします。

ステップ **2-5-6**: S3プラグインの導入

ssh接続した状態で以下のコマンドを実行します。

```
$ sudo su
# redmineのディレクトリに移動
cd /opt/bitnami/apps/redmine/htdocs/
# Pluginのダウンロード
git clone https://github.com/redmica/redmica_s3.git plugins/redmica_s3
```

```
# Pluginの設定ファイルの作成
cp plugins/redmica_s3/config/s3.yml.example config/s3.yml
vi config/s3.yml
```

s3.ymlファイルを開いて、以下のように設定します。
「**access_key_id**」「**secret_access_key**」は先ほどダウンロードしたCSVの情報を入力します。
また、bucketには先ほど作成したバケット名を入力します。例）redmine-user1-2021xxxx

```
production:
  access_key_id: CSVの情報を入力
  secret_access_key: CSVの情報を入力
  bucket: redmine-user1(自分の名前)-2021xxxx
  folder: files
  thumb_folder: tmp/thumbnails
  import_folder: tmp/imports
  region: ap-northeast-1
```

```
# 所有者の変更
chown -R bitnami:daemon plugins/redmica_s3
chown -R bitnami:daemon config/s3.yml
```

```
# Gemfileに以下の1行を追記
echo gem \'zip\',\'2.0.2\' >> Gemfile.local
# 必要ライブラリーのインストール
bundle install --no-deployment
bundle exec rake redmine:plugins RAILS_ENV=production
```

ステップ 2-5-7: Apacheの再起動&設定を反映

2. 以下のコマンドでapacheを再起動し、設定を反映させます。

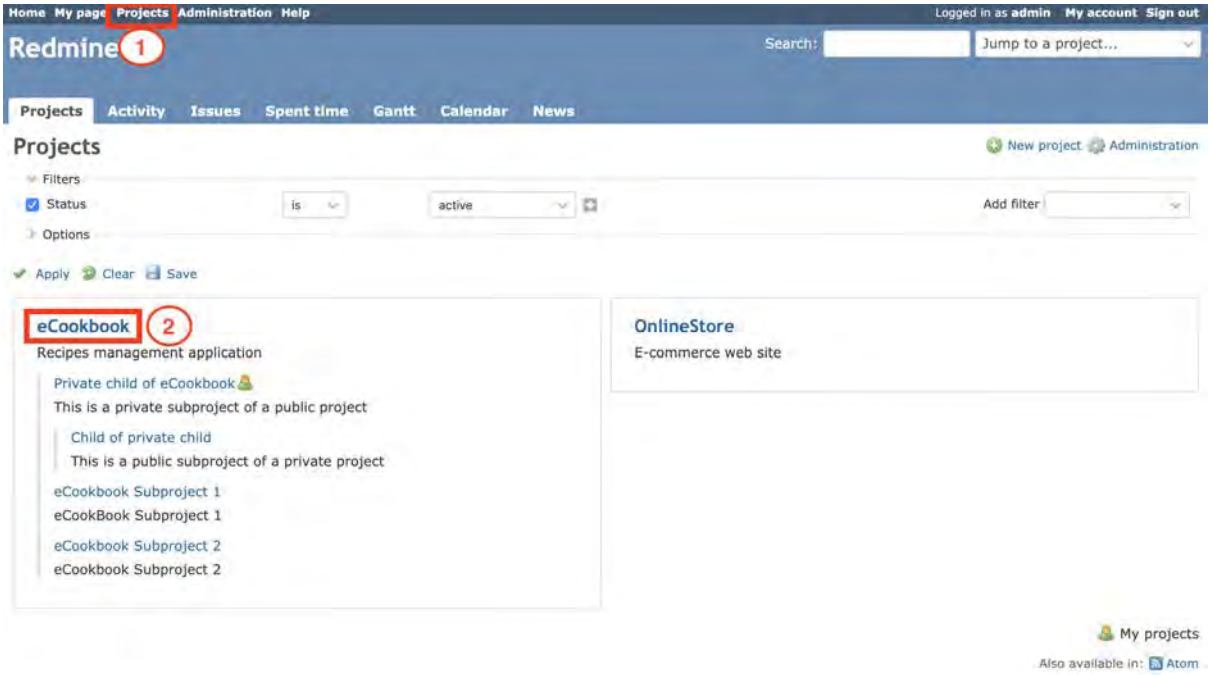
```
# apacheの停止
/opt/bitnami/apache2/scripts/ctl.sh stop

# apacheの起動
/opt/bitnami/apache2/scripts/ctl.sh start
```

3. 以下のコマンドでapacheのstatusを確認します。
「**apache already running**」と表示されることを確認します。

```
# apacheのステータスを確認
/opt/bitnami/apache2/scripts/ctl.sh status
```

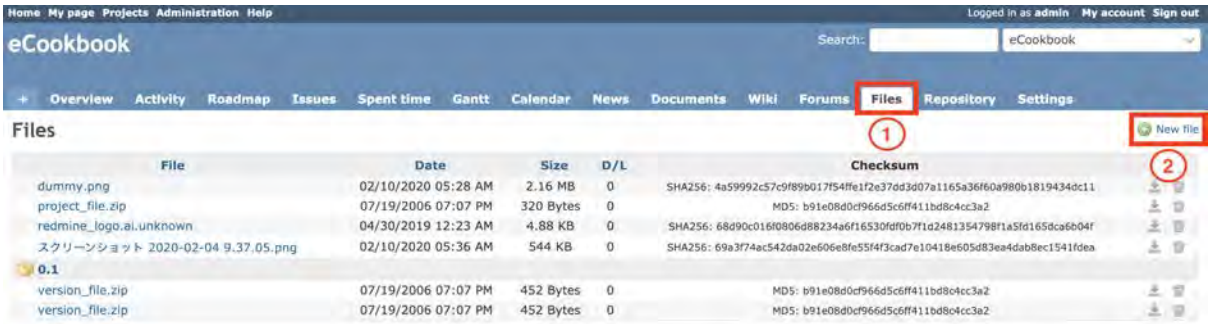
ステップ 2-5-8: 再度Redmineにファイルをアップロード

redmineに再度ファイルをアップロードします。
今回アップロードしたファイルはS3にも保存されます。

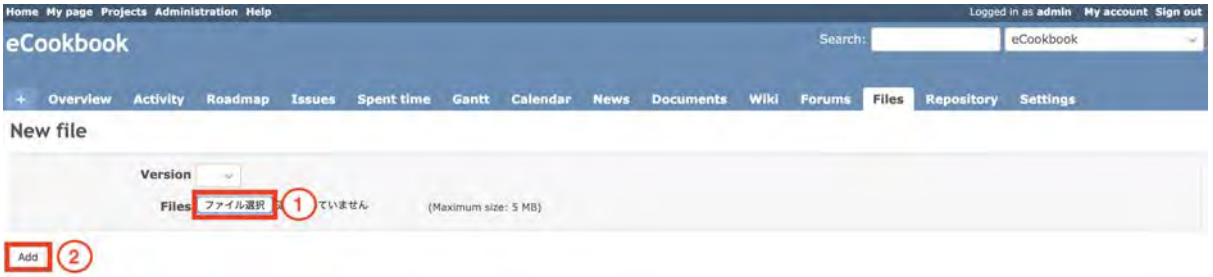


Redmineにアクセスして以下の手順でファイルをアップロードします。

3. 「**projects**」をクリックします。
4. 「**eCookbook**」をクリックします。



3. 「**Files**」をクリックします。
4. 「**New file**」をクリックします。



3. 「ファイルを選択」をクリックして、アップロードするファイルを選択します。
4. 「**Add**」をクリックします。

ステップ **2-5-9**: S3アップロード確認

redmineにアップロードしたファイルがS3に保存されているか確認します。

AWSコンソールを開きます。

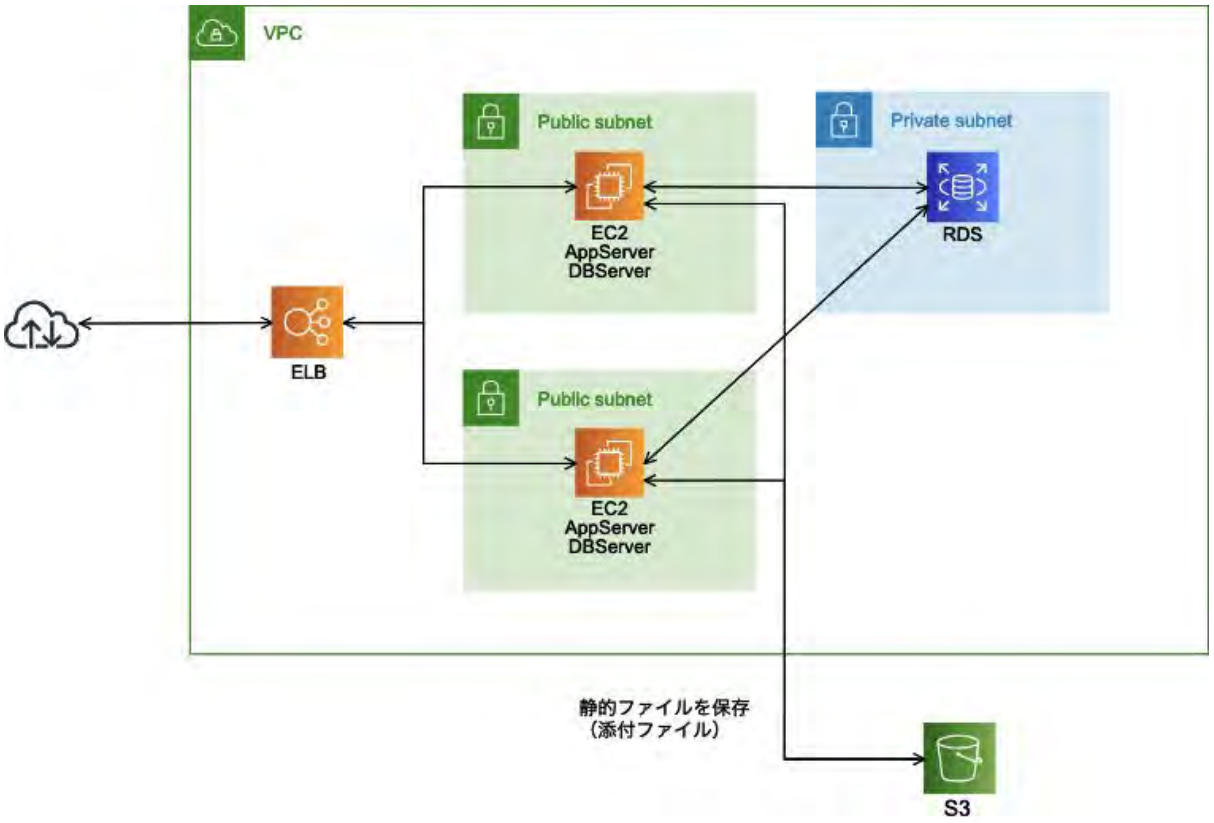
1. 「**s3**」を入力します。
2. 「**S3**」をクリックします。

先ほどフェーズ**2-5-5**で作成したバケット名をクリックします。自分の名前などで検索をかけると見つけやすいです。

1. 自分の名前などで検索します。(日付でも○)
2. フェーズ**2-5-5**で作成したバケット名をクリックします。

1. 「**files**」→「**2021**(年)」→「**02**(月)」をクリックして、先ほどredmineにアップロードしたファイルが保存されていることを確認します

[フェーズ 3]
~ ロードバランサーを使った負荷分散環境を構築 ~



▼フェーズ 3-1: Web サーバーの AMI（パッケージ）を作成

ステップ **3-1-1**: Amazon EC2 管理ページを開く

1. 「**ec2**」を入力します。
2. 「**EC2**」をクリックします。

ステップ **3-1-2**: Web サーバーの AMI を作成

1. 「インスタンス」をクリックします。
2. 「**webserver#1-**ユーザー名」をクリックします。
3. 「アクション」をクリックします。
4. 「イメージとテンプレート」-「イメージを作成」をクリックします。

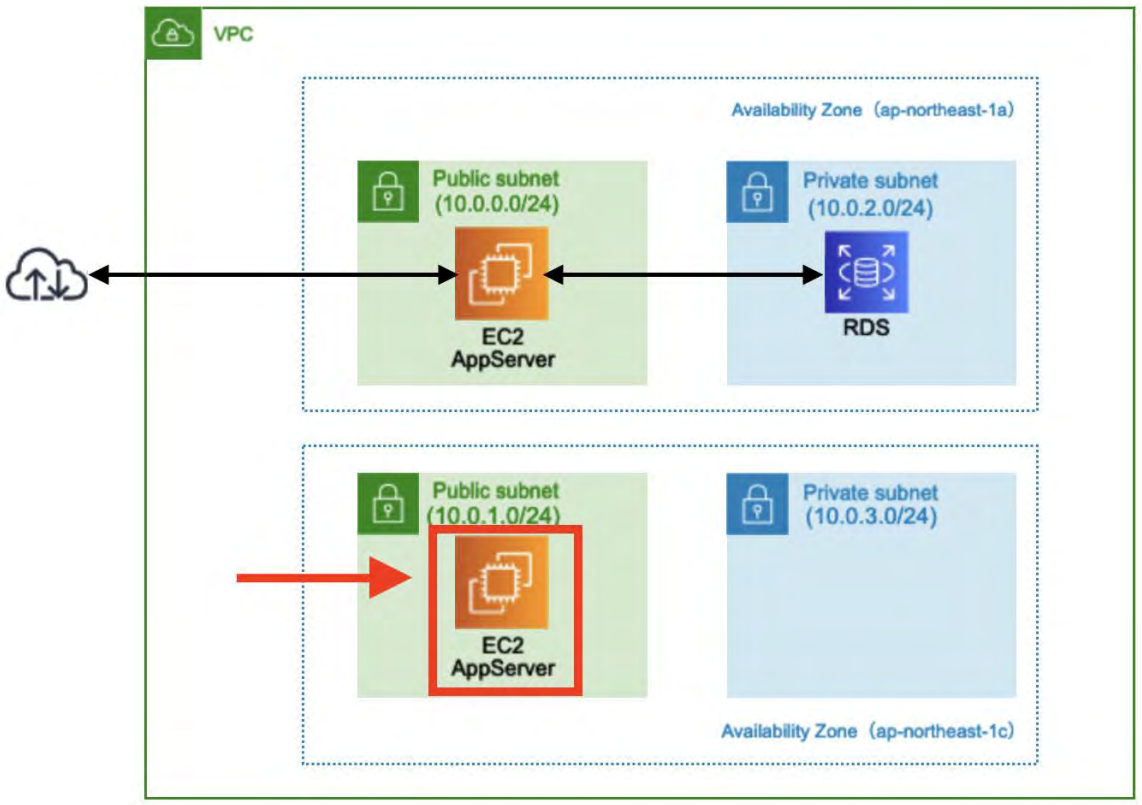
1. “redmine ユーザー名”などのイメージ名を入力します。
   例）[redmine user1]
2. 「イメージを作成」をクリックします。

AMI の画面で AMI 作成を待ちます。完了するまで数分かかります。

「状態」欄が「available」となれば作成完了です。

「available」が表示されない場合は画面をリロードしてください。

▼フェーズ 3-2: 2 個目の Amazon EC2 インスタンスを作成



ステップ 3-2-1: 2 個目の Amazon EC2 インスタンス作成

作成した AMI からインスタンスを作成します。

1. フェーズ**3-1-2**で作成した AMI を右クリックします。
   例）redmine user1
2. 「起動」をクリックします。

3. 「**t3.micro**」を選択します。
4. 「次のステップ: インスタンスの詳細の設定」をクリックします。

インスタンスは 1 個目と異なるアベイラビリティゾーンに作成します。

VPC とサブネットの選択に注意してください。

1. フェーズ**1-1-5**で作成した VPC を選択します。例）handson-user1
2. 「パブリックサブネット**|ap-northeast-1c]**」を選択します。
3. 「有効」を選択します。
4. 「次のステップ**:** ストレージの追加」をクリックします。

ストレージは変更せずに、次に進みます。

1. 「次のステップ**:** タグの追加」をクリックします。

インスタンスを区別できるようにタグに名前を設定します。

1. 「タグの追加」をクリックします。
2. キーに「**Name**」と入力します。

3. 「webserver#2- ユーザー名」とします。
   例）[webserver#2-user1]
4. 「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。

既に作ったセキュリティグループを使用します。

1. 「既存のセキュリティグループを選択する」をクリックします。
2. フェーズ **1-3-4** で作成したセキュリティグループ（**web-user1**等）をクリックします。
3. 「確認と作成」をクリックします。

設定内容を確認してから作成します。

1. 「起動」クリックします。

ステップ 3-2-2: キーペアを選択する

既存のキーペアを選択します。

1. 「既存のキーペアの選択」を選択します。
2. 先ほど作成した「**handson-2021xxxx**」を選択します。
3. チェックを入れます。
4. 「インスタンスの作成」をクリックします。

ステップ **3-2-3**: 作成した 2 個目の EC2 インスタンスを確認

インスタンスの作成が完了するのに数分間かかります。

他ユーザのインスタンスが表示されている場合は上部の検索ボックスにユーザー名を入れて絞り込んでください。

webserver#2-ユーザー名、ap-northeast-1c に作成されていることを確認してくだい。

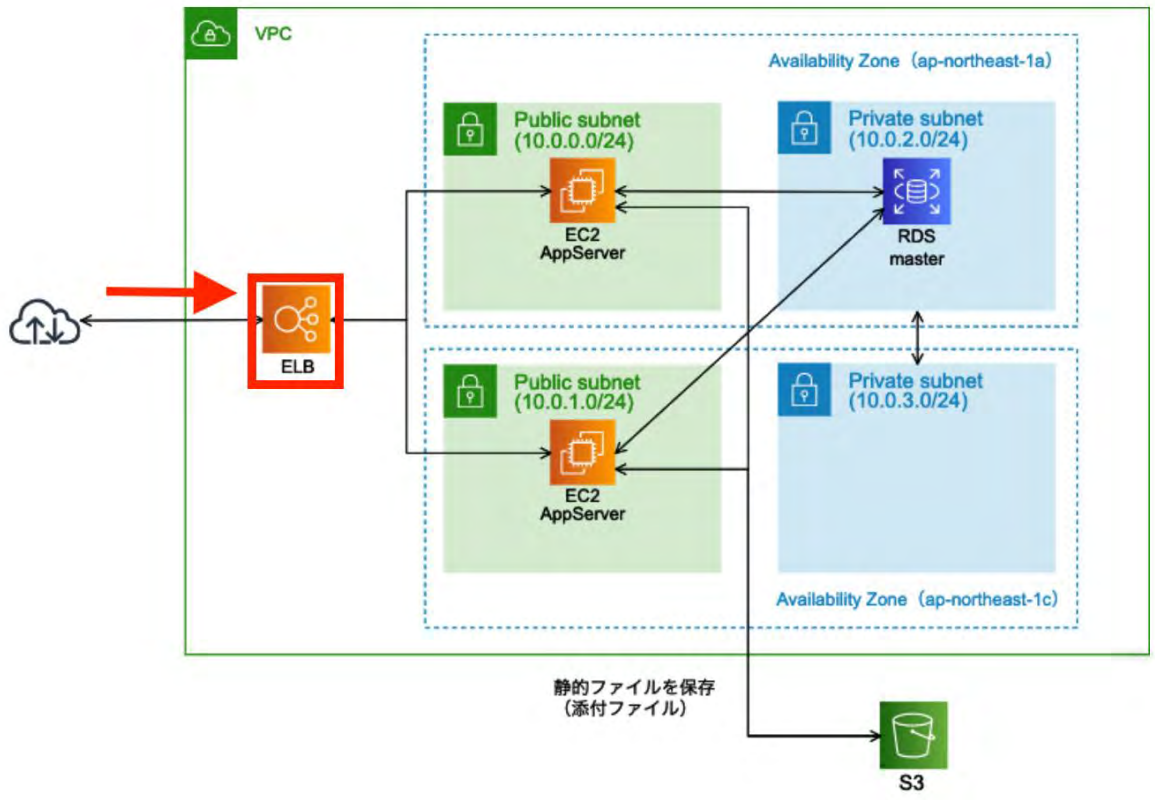
1. 2個目のインスタンスが作成されていることを確認する。
2. 2個目のインスタンスを選択し、「ネットワーキング」をクリックします。
3. アベイラビリティーゾーンが「**ap-northeast-1c**」であることを確認する。

ステップ 3-2-4: 2 個目の EC2 インスタンスのパブリックIPv4アドレスをメモ

1. 2個目のインスタンス「webserver#2- ユーザー名」を選択する
   例）[webserver#2-user1]
2. 「パブリック**IPv4**アドレス」をメモする。

▼フェーズ 3-3: Elastic Load Balancing（ロードバランサー）を作成

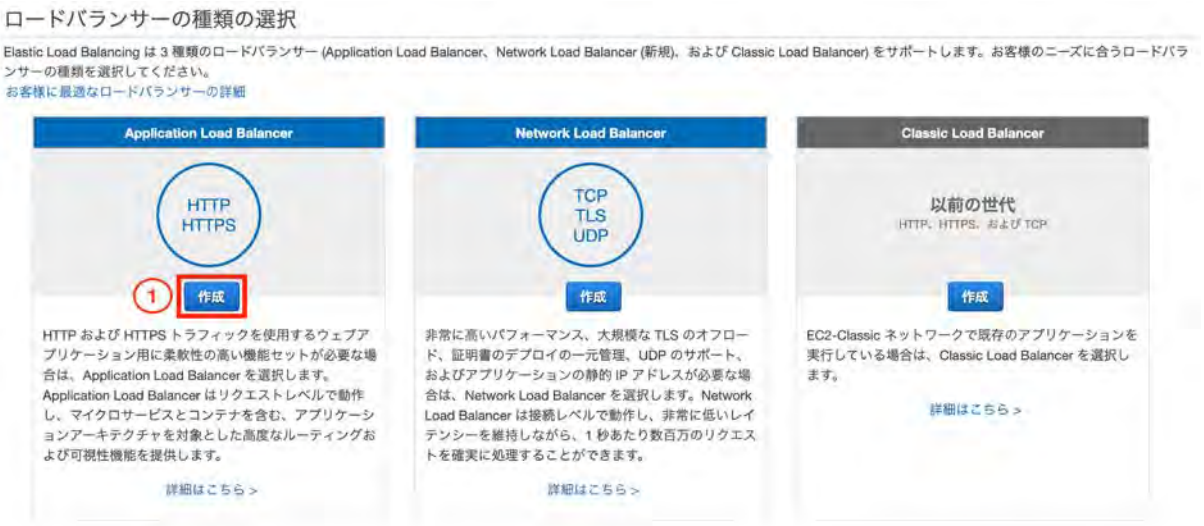
ステップ **3-3-1**: ELB を作成



2 台の Web サーバーへのアクセスを振り分ける ELB を作成します。

1. 「ロードバランサー」を選択します。(EC2のサイドメニュー)
2. 「ロードバランサーの作成」をクリックします。

ステップ **3-3-2**: ALBを選択



今回は「**Application Load Balancer** 」を選択します。

ステップ **3-3-3**: ELB を作成（1）

1. 「**elb-ユーザ名**」と入力します。
   例）elb-user1

ELB を 2 つのパブリックサブネットに配置します。

利用可能なサブネット一覧からパブリックサブネット 2 つを「＋」をクリックして選択してください。

1. VPCは「**handson-user1**」をクリックを選択します。
2. 「**ap-northeast-1a**」,「**ap-northeast-1c**」にチェックを入れ、それぞれ「パブリックサブネット」を選択します。
3. 「次の手順: セキュリティ設定の構成」をクリックします。

特に設定しないため、そのまま次の手順へ進みます。

4. 「次の手順: セキュリティグループの設定」をクリックします。

ステップ **3-3-4**: ELB を作成（2）

1. 「新しいセキュリティグループを作成する」を選択します。
2. セキュリティグループ名と説明に「**elb-**ユーザ名」名前を入力します。
   例）elb-user1
3. 「**HTTP**」を選択します。
4. 「任意の場所」を選択します。
5. 「次の手順: ルーティングの設定」をクリックします。

ステップ **3-3-5**: ELB を作成（3）

新しいターゲットグループを作成します。

1. 「**elb-**ユーザ名」と名前を入力します。
   例）elb-user1

ステップ **3-3-6**: ELB を作成（4）

ヘルスチェックの条件を変更します。

1. 「**/login**」に変更します。
2. 「次の手順: ターゲットの登録」をクリックします。

ステップ **3-3-7**: ELB を作成（5）

1. 「インスタンス」に表示されている2つのインスタンスを選択する。
2. 「登録済みに追加」をクリックする。

登録済みターゲットに2つのインスタンスが追加されていることを確認する。

1. 「登録済みターゲット」に2つのインスタンスが追加されていることを確認する。
2. 「次の手順: 確認」をクリックする。

ステップ 3-3-8: ELB を作成（6）

1. 「作成」をクリックします。

ステップ 3-3-10: 作成されたELBを確認

ELB が作成されました。

1. 「閉じる」をクリックします。

作成された ELB の DNS 名（ホスト名）をメモします。

(Aレコード)は省きます。

1. ユーザー名で絞りこみます。
2. 先ほど作成した **ELB** を選択します。
3. **DNS**名をメモします。

※ ロードバランサーが作成されるまで少し時間がかかります。「状態」がactiveになったのを確認してからアクセスしてください。

▼フェーズ 3-4: Elastic Load Balancing 経由でアクセス

ステップ 3-4-1: ELB 経由でアクセス

http:///<ELB の DNS 名>/ を開いてredmine が表示されることを確認します。

ステップ 3-4-2: 両方のサーバにアクセスがされているか確認

webserver#1, webserver#2 それぞれに ssh でログインし、以下のコマンドを実行してアクセスログを表示させることが可能です。

ELB の定期的なヘルスチェックが実行されたり、redmine でページをリロードするたびに双方の EC2 へアクセスされている状況を確認できます。

[**webserver#1**]

以下のコマンドを実行します。

```
ssh -i "handson-2021xxxx.pem" bitnami@[ webserver#1のElastic IPアドレス ]
```

```
$ sudo su
```

```
# redmineのディレクトリに移動
$ cd /opt/bitnami/apps/redmine/htdocs/
# アクセスログを表示
$ tail -f log/production.log
```

[**webserver#2**]

別ターミナルを開き、**webserver#1**と同様にssh接続してアクセスログを表示します。

```
ssh -i "handson-2021xxxx.pem" bitnami@[ webserver#2のパブリックIPv4アドレス
]
```

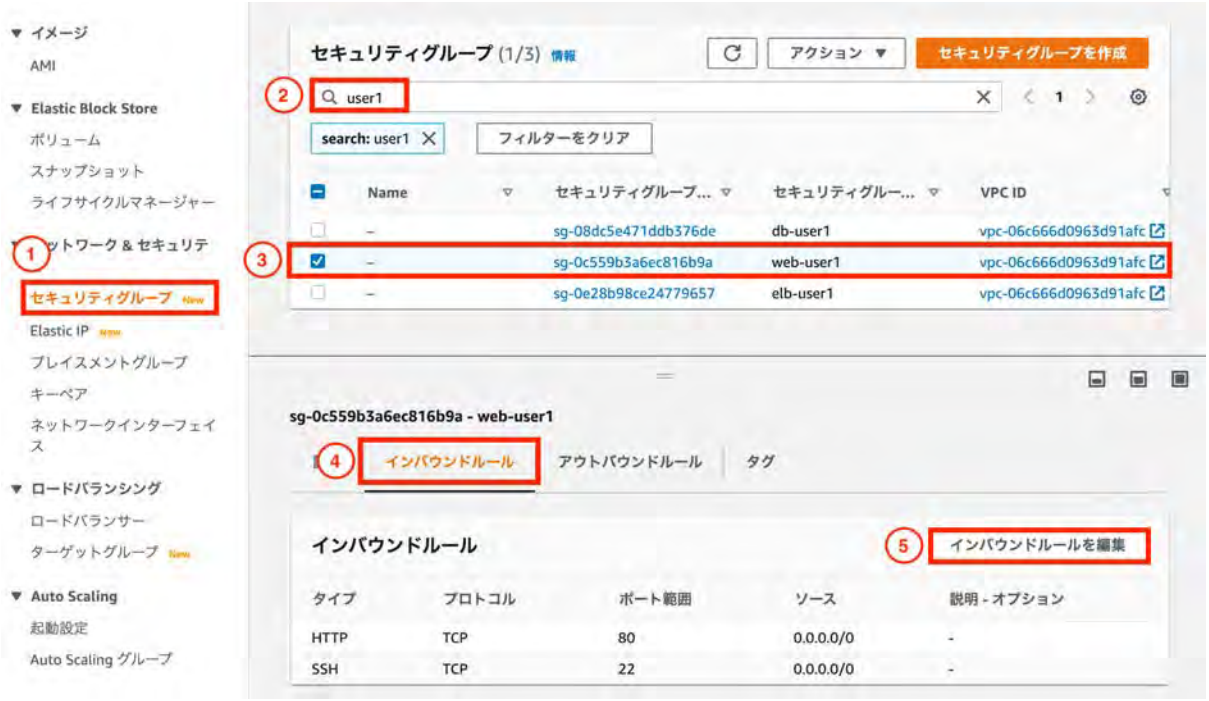
```
$ sudo su
# redmineのディレクトリに移動
$ cd /opt/bitnami/apps/redmine/htdocs/
# アクセスログを表示
$ tail -f log/production.log
```

redmineをリロード等してログがそれぞれに流れることを確認してください。

ログ表示はCtrl + C で終了できます。

▼フェーズ 3-5: セキュリティグループ設定変更

ステップ 3-5-1: セキュリティグループ設定変更



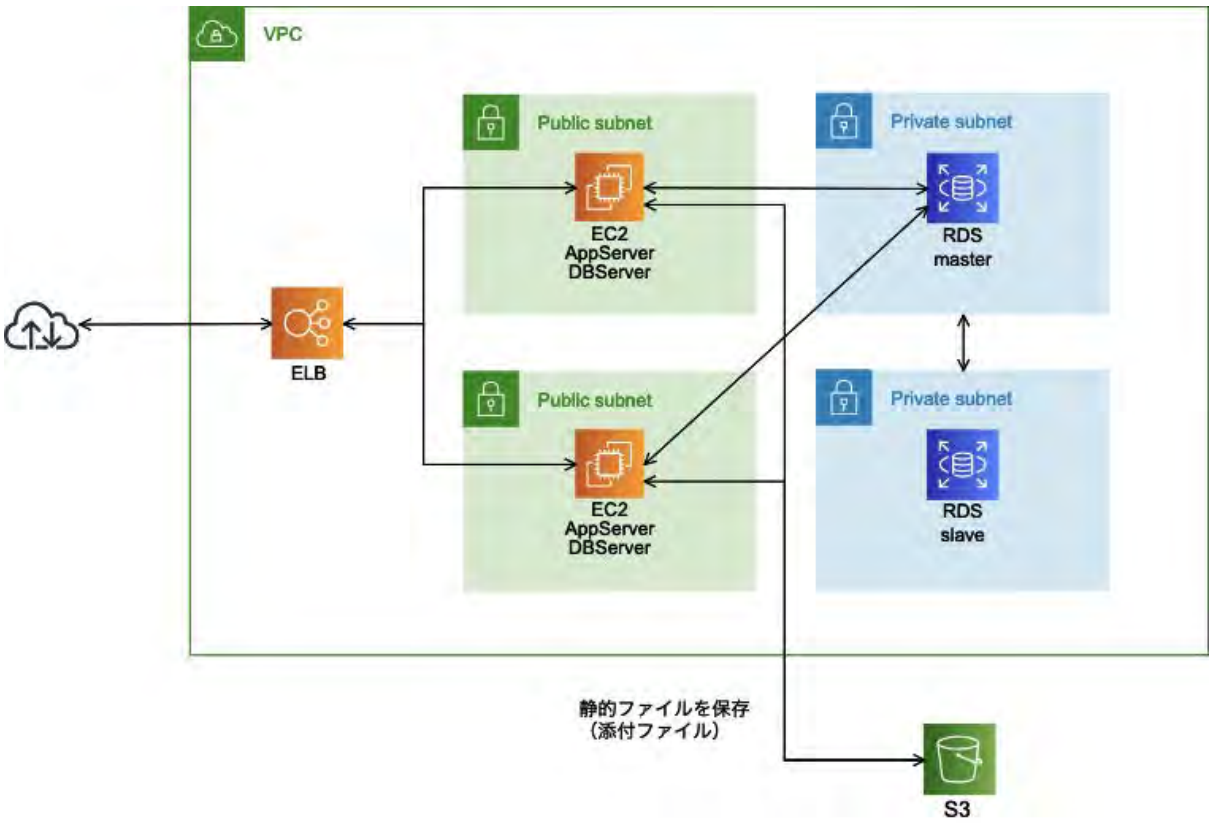
セキュリティグループの設定を変更し、Web サーバーへの HTTP アクセスは ELB からに限定するようにします。

1. 「セキュリティ グループ」を クリックします。(EC2のサイドメニュー)
2. ユーザー名で絞込みます。
3. グループ名「**web-**ユーザー名」を選択します。
4. 「インバウンドルール」をクリックします。
5. 「インバウンドルールを編集」をクリックします。

1. タイプ httpのリソースの「0.0.0.0/0」を x をクリックして削除します。
2. 入力欄に「**elb**」と入力して候補を表示させます。
3. 表示された候補から「**elb-**ユーザー名」を選択します。
4. 「ルールを保存」をクリックします。

保存後、ロードバランサーのDNSからはアクセス可能で、EC2のパブリックDNSからアクセスできないことを確認してください。

[フェーズ 4]
~ Amazon RDS を Multi-AZ 構成に変更 ~



▼フェーズ 4: Amazon RDS を Multi-AZ 構成に変更

ステップ 4-1: RDS 管理ページを開く

1. 「**rds**」を入力します。

2. 「**RDS**」をクリックします。

ステップ **4-2**: RDS インスタンスの設定変更

1. 「データベース」を選択します。
2. フェーズ**2-3-1**で作成した**RDS**インスタンスを選択します。
3. 「変更」をクリックします。

ステップ **4-3**: Multi-AZ を有効にする

マルチAZ配置設定

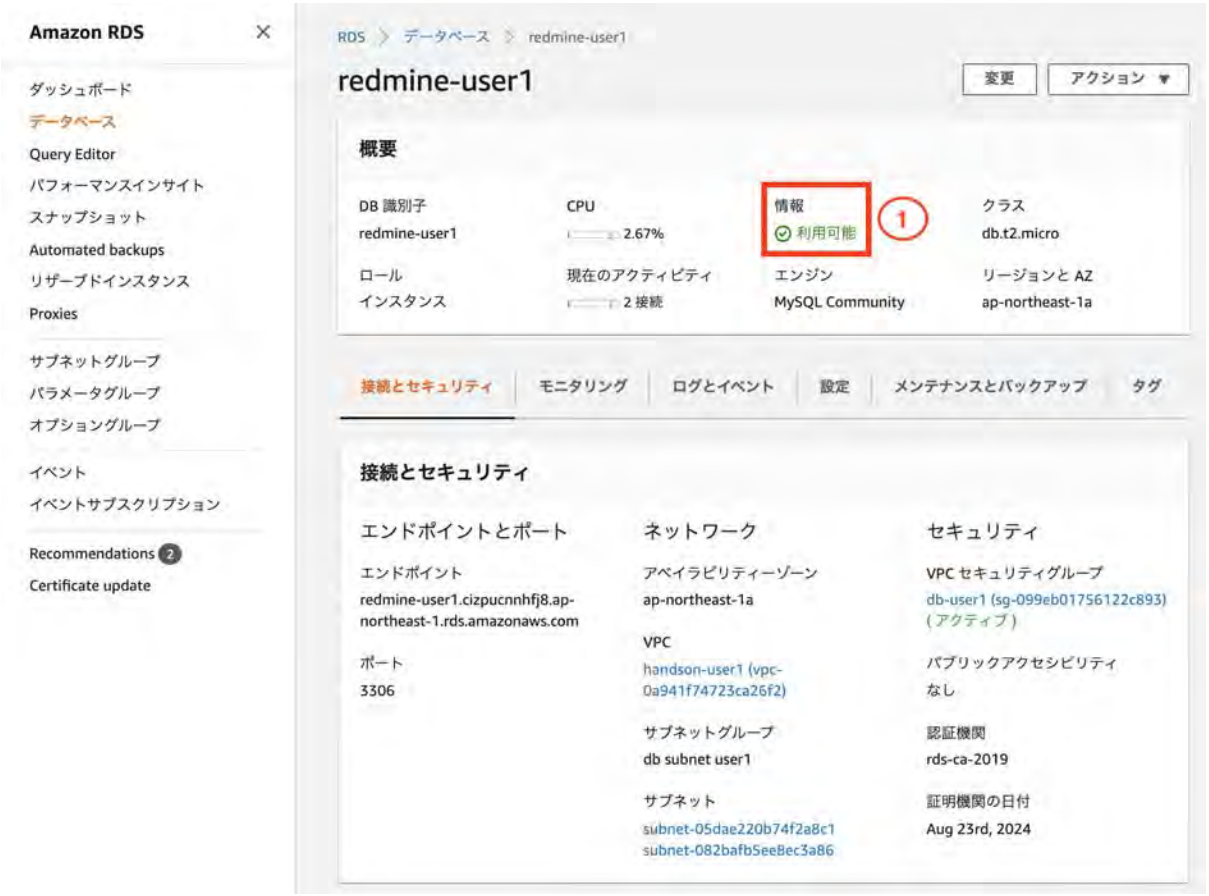
1. 「可溶性と耐久性」の項目で「スタンバイインスタンスを作成する」を選択します。
2. 選択後、ページ下部の「続行」をクリックします。

「すぐに適用」をオンにしなければ、サーバーの停止や負荷が伴う変更は次のメンテナンスウィンドウのタイミングで適用されますが、今回は「すぐに適用」を行います。

1. 「すぐに適用」にチェックを入れます。
2. 「**DB**インスタンスの変更」をクリックします。

ステップ 4-4: Multi-AZ 化の完了を確認

1. フェーズ**2-3-1**で作成した**DB**インスタンス（例**: redmine-user1**）をクリックします。



変更完了を待ちます（約 10 分間かかります）。

ステータスが[利用可能]にならない場合は、画面を更新して再描画します。

ステップ 4-5: 設定変更内容を確認

Multu-AZ 配置への設定がすぐに適用されることを確認します。

1. 「設定」をクリックします。
2. 「マルチ**AZ**」がありであることを確認します。

ステップ **4-6**: RDS インスタンスをフェイルオーバーさせる

RDS をスタンバイ側に切り替え、挙動を確認します。

1. 「データベース」をクリックします。
2. フェーズ**2-3-1**で作成したインスタンスを選択します。
3. 「アクション」をクリックします。
4. 「再起動」をクリックします。

フェイルオーバーを選択して再起動させます。(再起動が完了するまでは redmine にアクセスできなくなります。再起動が完了すると元通りアクセスできるようになります。)

1. 「フェイルオーバーし再起動します」にチェックを入れます。
2. 「再起動」をクリックします。

起動後、「リージョンとAZ」が **ap-northeast-1c**に変更されていることを確認してください。

~ 構築した環境の後片付け ~

今回構築した環境は、そのままにしておくと費用が発生するものがあります。

フェーズ 4 までの作業終了・または途中で作業を終了される場合は、以下の手順で構築した環境の後片付けをお願いします。

以下の手順で構築した環境の後片付けをしてください。

[RDS]

＊ データベース
DB識別子が「**redmine-**自分の名前（**user1**）」を削除

1. 選択->アクション->削除
2. 「最終スナップショット を作成しますか？」のチェックを外す。
3. 「インスタンスの削除後、システムスナップショットとポイントインタイムの復元を含む自動バックアップが利用不可となることを了承しました。」にチェックをいれる
4. 「delete me」を入力後、削除する

削除するのに時間がかかるため、RDS以外を先に削除する

[ec2]

＊ インスタンス

webserver#1-自分の名前（**user1**）と**webserver#2-**自分の名前（**user1**）それぞれ削除

1. 選択 -> アクション -> インスタンスの状態 -> 終了
2. インスタンスの状態が「terminated」となれば OK

＊ Elastic IP アドレス
自分が作成したインスタンスと関連付けている**Elastic IP** アドレスを削除

1. 選択 -> Actions -> Elastic IPアドレスの関連付けの解除
2. その後、もう一度選択して Elastic IPアドレスの関連付けの開放をする

＊ AMI

「**redmine** 自分の名前（**user1**）」を登録解除

1. 選択 -> アクション -> 登録解除

＊ロードバランサー

「**elb-**自分の名前（**user1**）」を削除

1. 選択 -> アクション -> 削除

[s3]

＊ バケット

「**redmine-**自分の名前（**user1**）**-2021xxxx**」を削除

1. バケット名をクリックし、「files/」を選択 -> 削除
2. 「完全に削除」と入力し、オブジェクトを削除
3. バケット一覧画面に戻り、選択 -> 削除
4. バケット名を入力後、削除

[IAM]

＊ ユーザー

「**s3access-2021xxxx**」を削除

1. 選択 -> ユーザーの削除
2. チェックボックスをオンにしたあと、削除

[ec2]

＊キーペア

本日利用した「**handson-2021xxxx**」キーペアを削除

1. 選択 -> アクション -> 削除
2. 削除を入力後、削除ボタンをクリック

＊セキュリティグループ

「**db-**自分の名前（**user1**）」「**web-**自分の名前（**user1**）」「**elb-**自分の名前（**user1**）」の順でそれぞれ削除する

※ 削除されない場合は、時間を置いてください。

1. 選択 -> アクション -> セキュリティグループの削除

＊ターゲットグループ

「**elb-**自分の名前（**user1**）」を削除

1. 選択 -> アクション -> 削除

[VPC]

＊VPC

「**handson-**自分の名前（**user1**）」を削除

1. 選択 -> アクション -> 削除

[RDS]

＊サブネットグループ

「**db subnet** 自分の名前（**user1**）」を削除

1. データベースが削除されるまで待ちます
2. 選択->削除