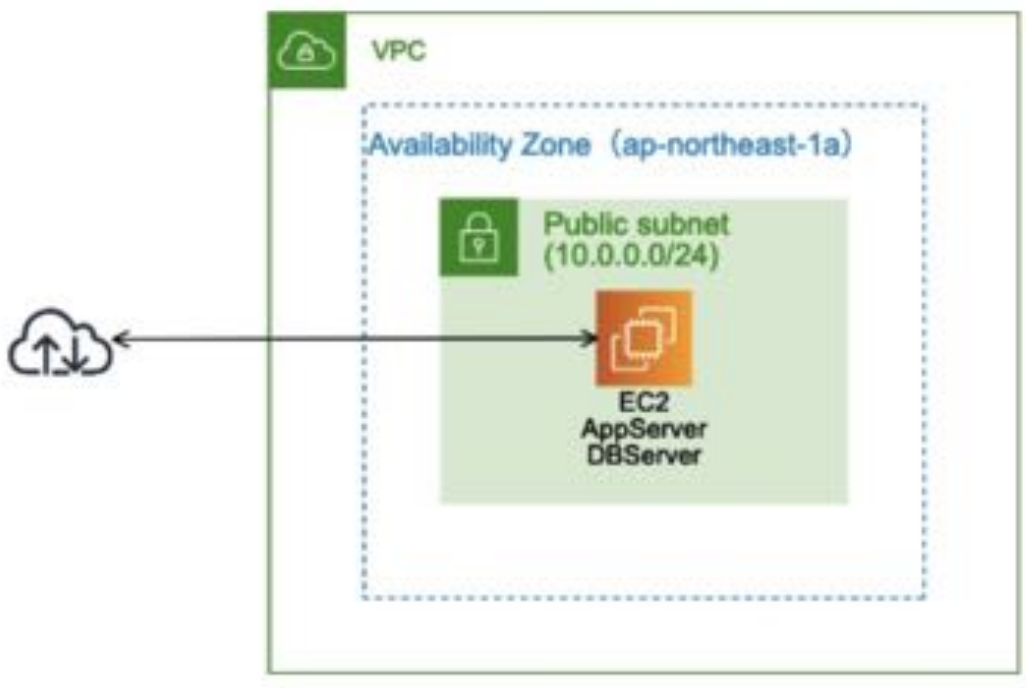
AWSハンズオン

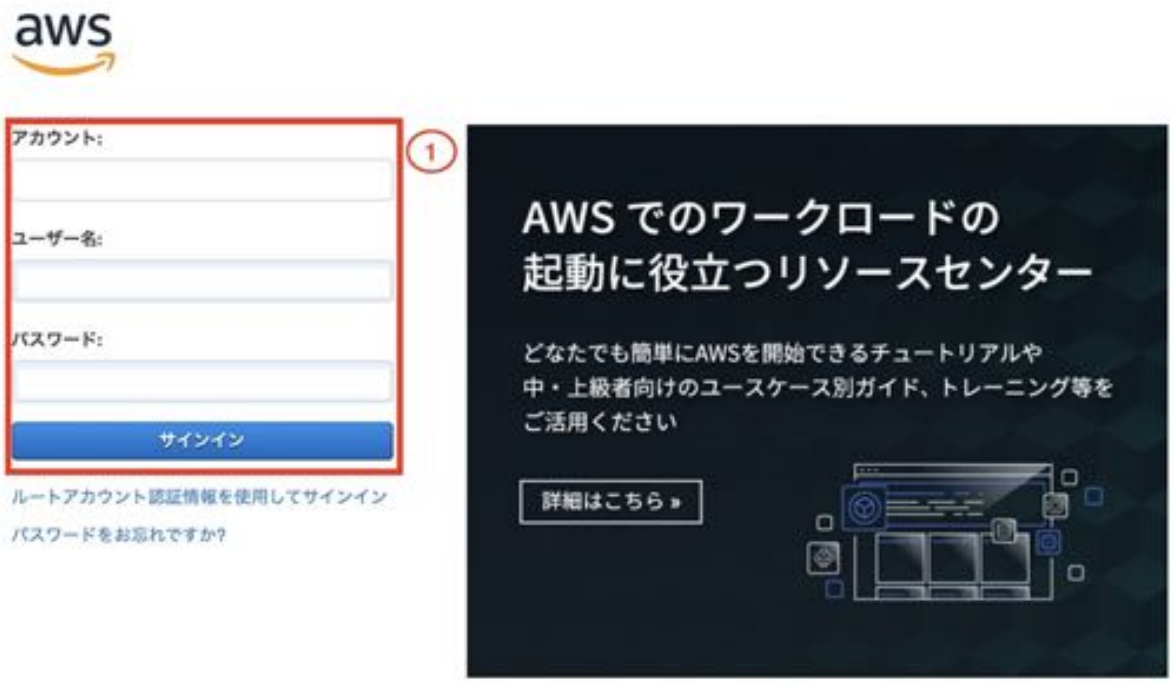
参考サイト
https://aws.amazon.com/jp/getting-started/projects/scalable-wordpress-website/

[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

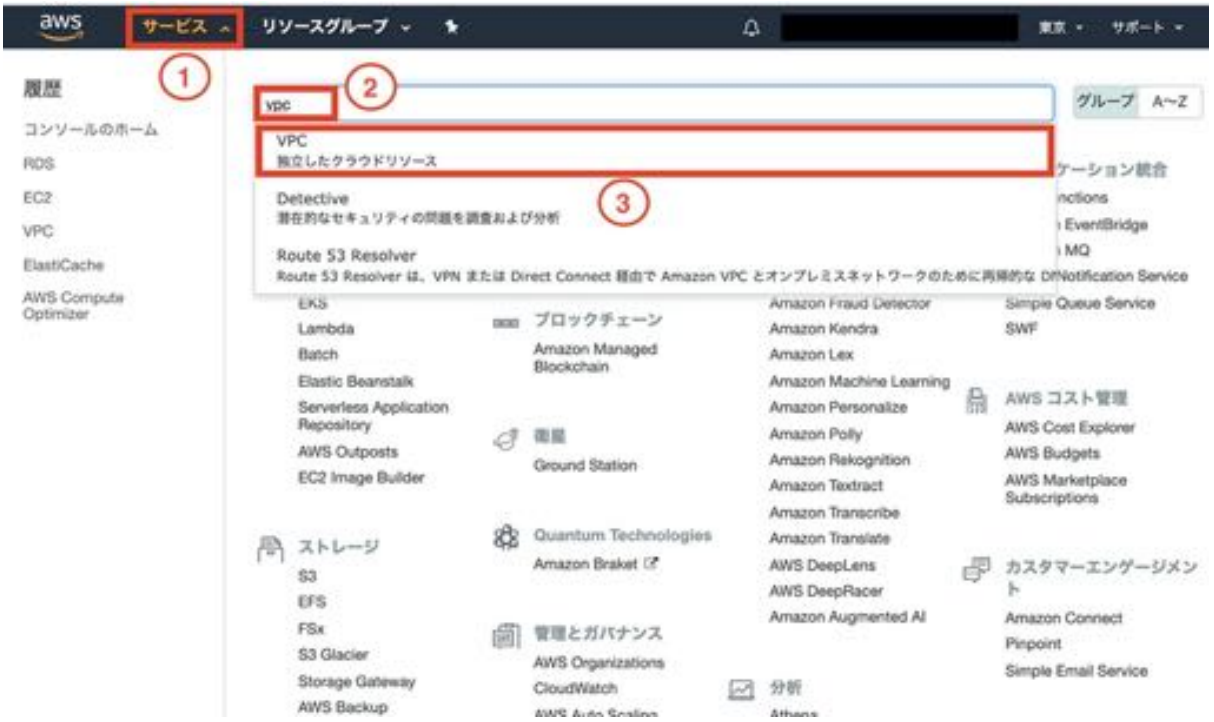
- https://console.aws.amazon.com にブラウザでアクセスしてください。
- アカウントの E メールアドレスとパスワードでログインしてください。

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. 「**サービス**」をクリックします。
2. **VPC** と入力します。
3. 「**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. 一度画面をリロード後、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。

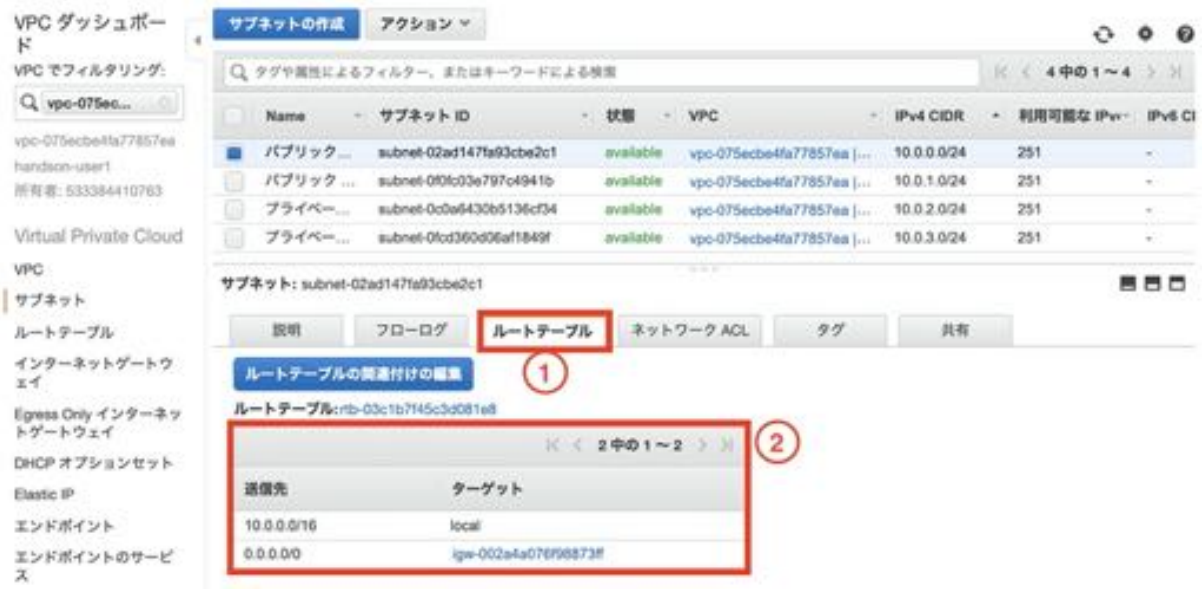
ステップ 1-1-7: 作成された VPC の確認

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. 下記の表の通り入力します。**VPC 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つ目はすでに作成されています

	ネームタグ	VPC	アベイラビリティゾーン	CIDR ブロック
2つ目	パブリック サブネットc	ご自身のVPCを選択してください。	ap-northeast-1c	10.0.1.0/24
3つ目	プライベート サブネットa		ap-northeast-1a	10.0.2.0/24
4つ目	プライベート サブネットc		ap-northeast-1c	10.0.3.0/24

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

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

追加した2つ目のサブネット**「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。（変更するサブネットは 2 つ目のみです）

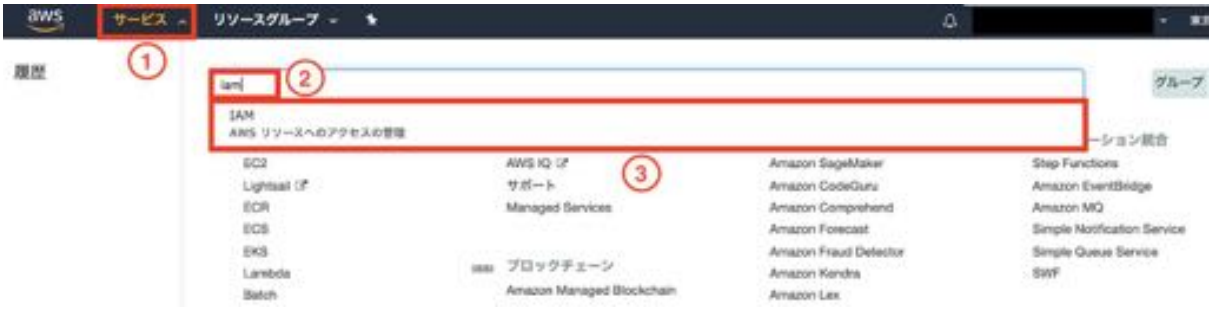
1. **「10.0.1.0/24」**のサブネットをクリックします。
2. **「ルートテーブル」**をクリックします。
3. **「ルートテーブルの関連付けの編集」**を クリックします。

1. これまでと異なるものを選択してくだい。
   ※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

ステップ 1-3-1: IAMページを開く



1. 「**サービス**」をクリックします。
2. 「**IAM**」を入力します。
3. 「**IAM**」をクリックします。

ステップ 1-3-2: Roleを作成する

IAMロールを使うためにRoleの作成をします。

1. 「**ロール**」をクリックします。
2. 「**ロールの作成**」をクリックします。

1. 「**AWSサービス**」を選択します。
2. 「**EC2**」を選択します。
3. 「**次のステップ: アクセス権限**」をクリックします。

1. 「AmazonEC2RoleforSSM」を入力して検索をかけます。
2. 「AmazonEC2RoleforSSM」にチェックを入れます。

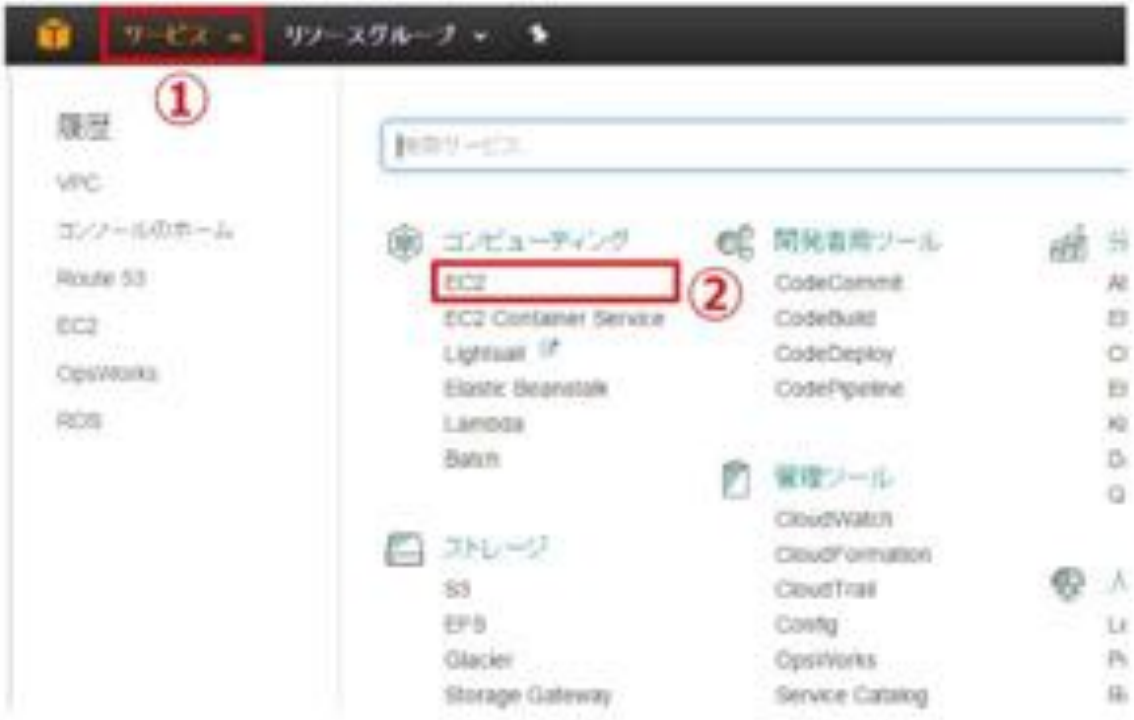
3. 「次のステップ: タグ」をクリックします。

注）**AmazonSSMManagedInstanceCore**の方が権限が狭く推奨されています。

1. キーに「Name」を入力します。
2. 値（オプション）に「session-manager-20200228」を入力します。
3. 「次のステップ: 確認」をクリックします。

1. ロール名に「session-manager-20200228」を入力します。
2. 「**ロールの作成**」をクリックしてロールを作成します。

ステップ 1-3-3: EC2 管理ページを開く



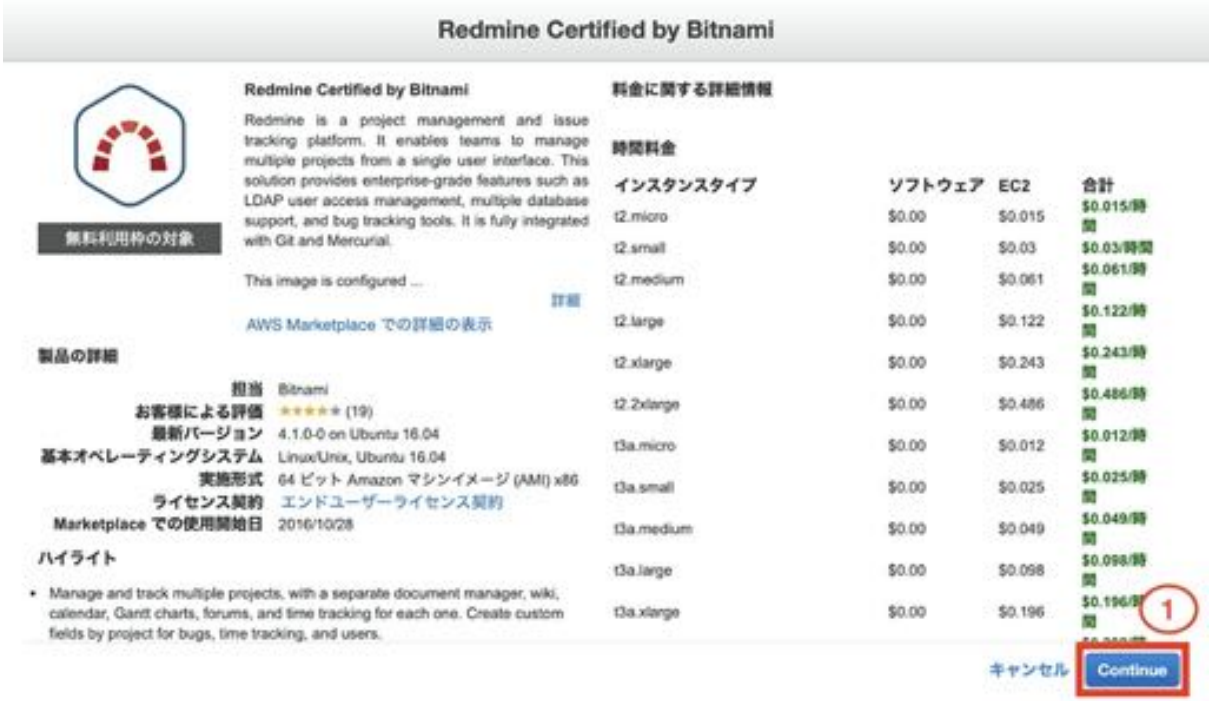
4. 「**サービス**」をクリックします。
5. 「**EC2**」をクリックします。

ステップ 1-3-4: EC2 インスタンスの作成（その1）

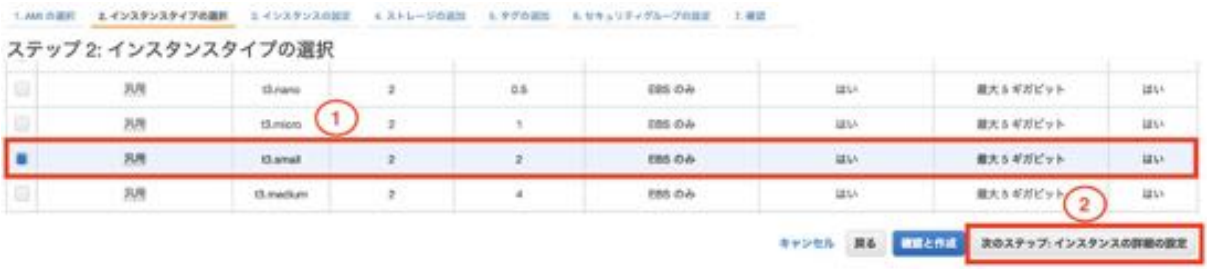
Web サーバーの作成を行います。

1. **「インスタンス」**をクリックします。
2. 「**インスタンスの作成**」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.small」**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. **フェーズ1-3-2で作成した**「session-manager-20200228」を選択します。
5. **「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。-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。複数のルールタイプが表示されますが、ルールタイプ「**HTTP**」ソース「**任意の場所**」のもの**1つ**だけに設定します。

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. ソースタイプを「**HTTP**」に設定します。
4. ソースを「任意の場所」に設定します。

5. その他のルールタイプは「✖」をクリックして削除します。

画像のようにタイプ「HTTP」ソース「任意の場所」のルールタイプが1つだけ設定されていることを確認します。

1. タイプ「**HTTP**」ソース「**任意の場所**」のルールタイプが**1つ**だけ設定されていることを確認します。
2. 「**確認と作成**」をクリックします。

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 1-3-5: キーペアを選択する

キーペアはなしで続行します。

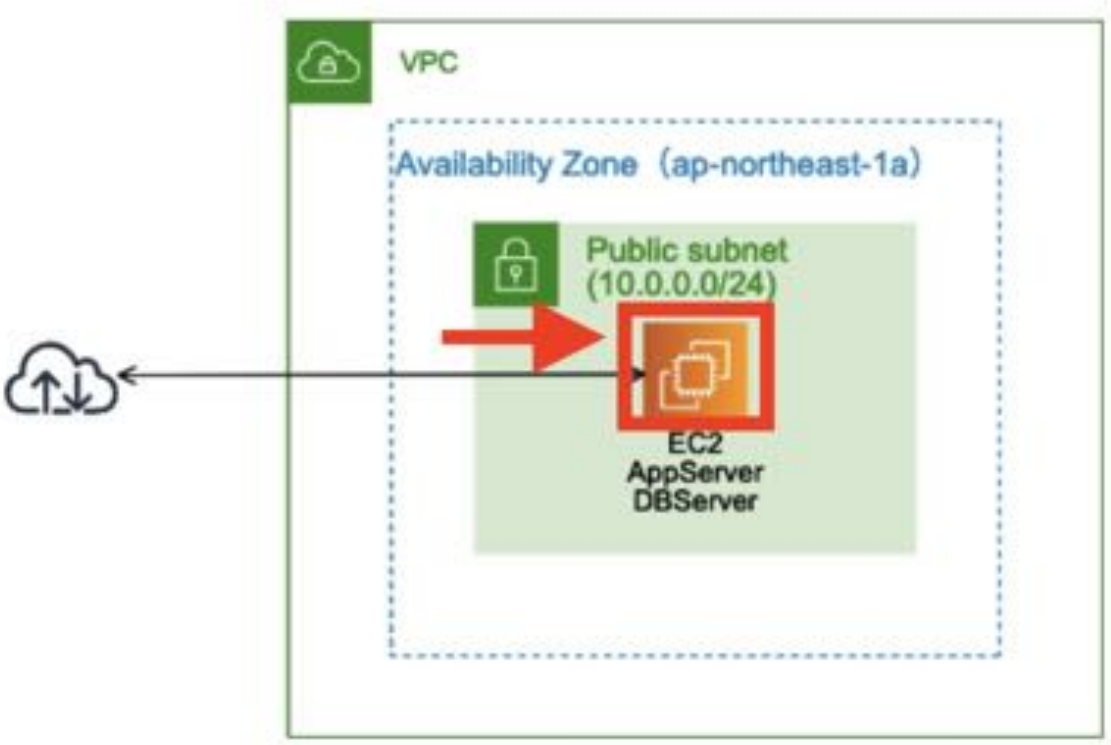
1. **「キーペアなしで続行」**を選択します。
2. **「このAMIに組み込まれたパスワードがわからないと、このインスタンスに接続できないことを認識しています。」**にチェックを入れます。
3. **「インスタンスの作成」**を選択します。

ステップ 1-3-6: EC2 インスタンスの作成

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。

図のオレンジの部分を作成しました。
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 1-3-7: 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。

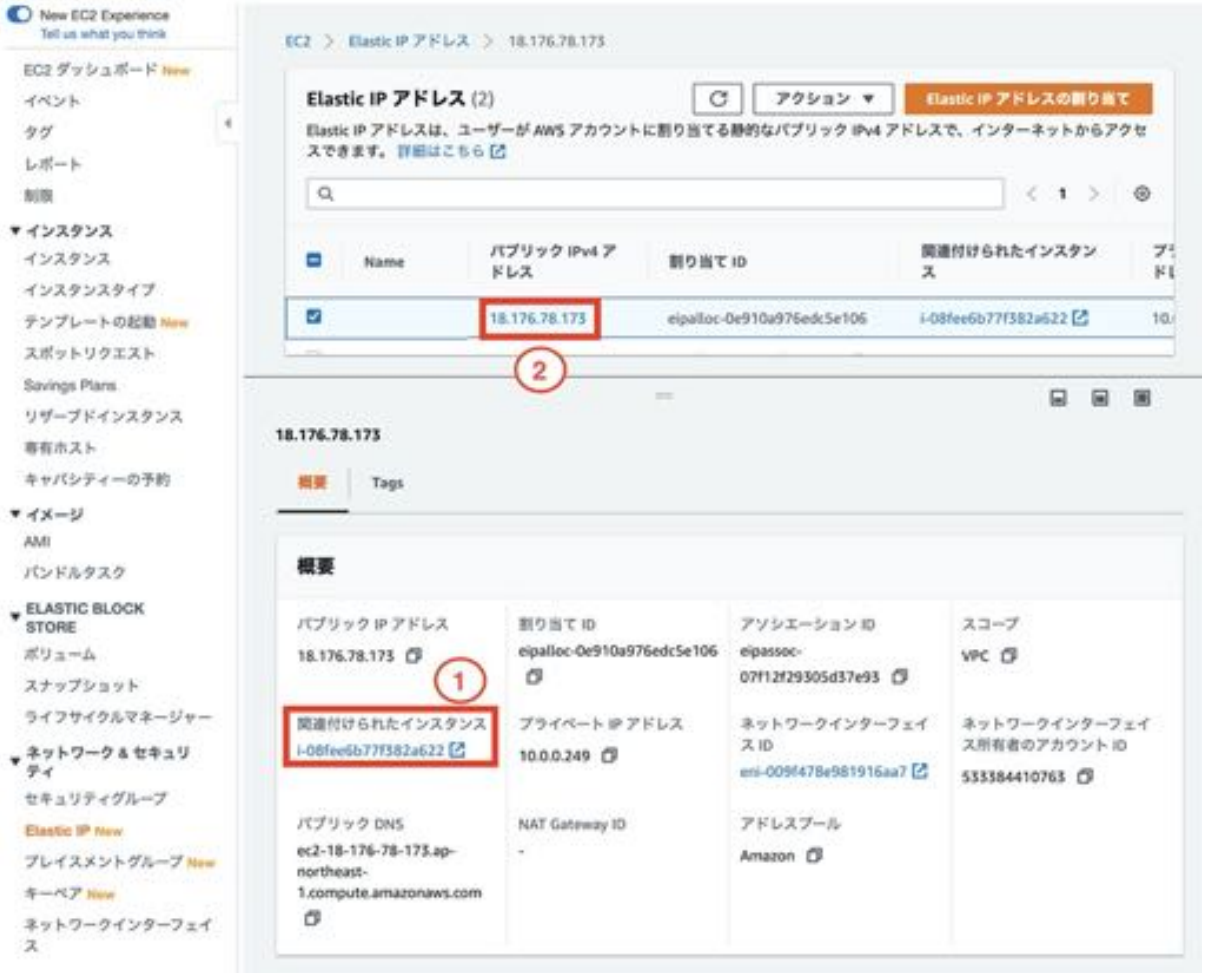
ステップ 1-4-2: Elastic IP (EIP) をインスタンスに紐付け

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1-3-4**で作成した EC2 インスタンスを選択してください。

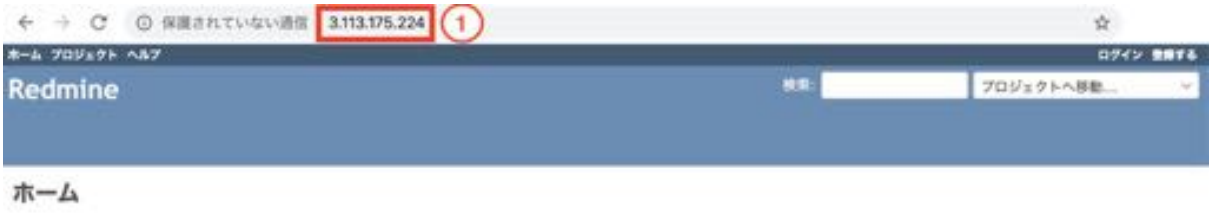
1. クリックすると候補が表示されます 自分の名前（例. user1） 等を入力し**フェーズ1-3-4**で作成した EC2 インスタンスを選択してください。
   例）[webserver#1-user1]等
2. **「関連付ける」**をクリックします。



紐付けされた EC2 インスタンスと EIP 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. 正しくインスタンスに紐付けられたかを確認します。
2. EIP をメモします。

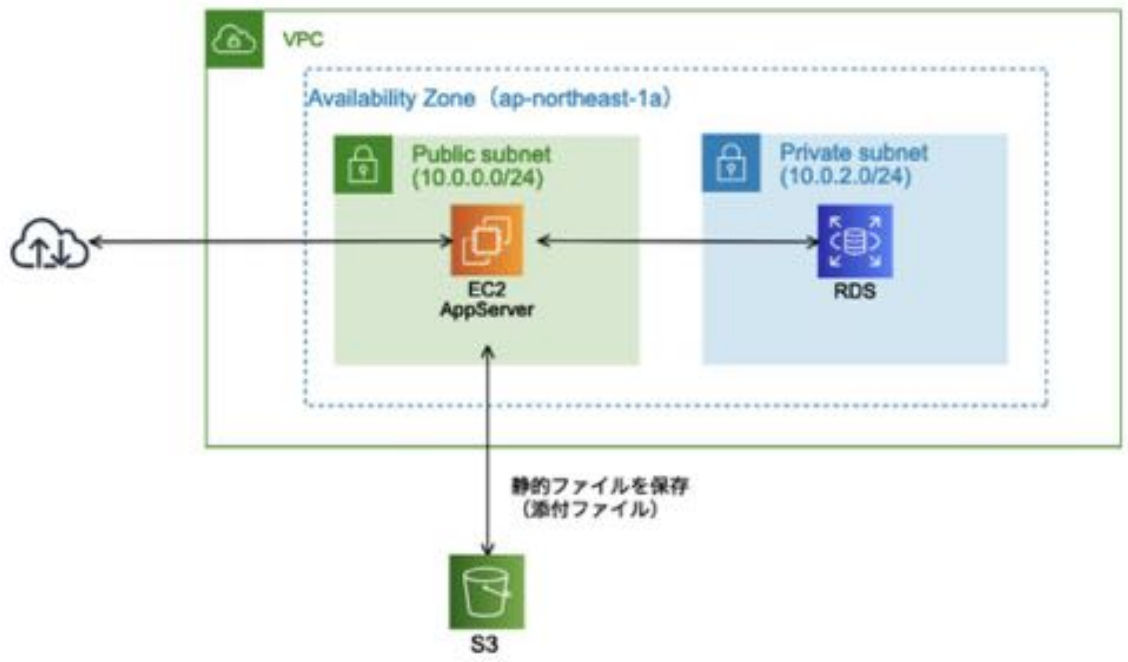
ステップ 1-4-3: Redmineにアクセス



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. 「 **サービス**」をクリックします。
2. 「**ec2**」を入力します。

3. 「**EC2**」をクリックします。

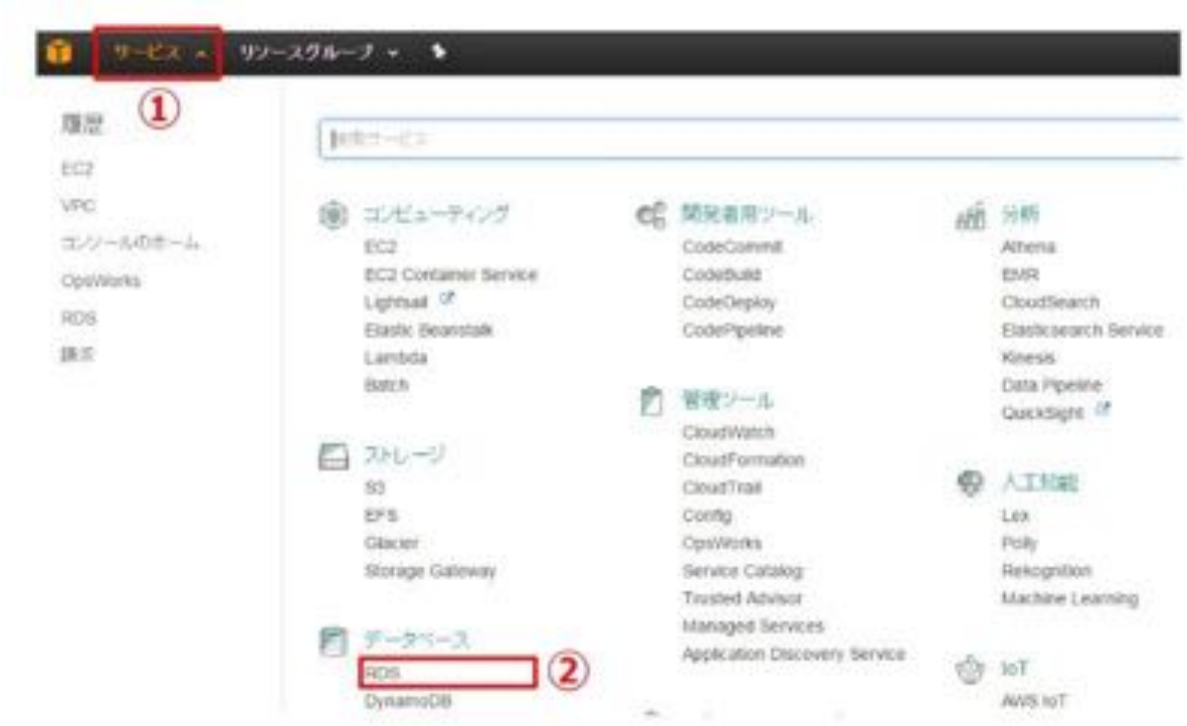
4. 「**セキュリティグループ**」をクリックします。
5. 「**セキュリティグループの作成**」をクリックします。

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。

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. 「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. 「**サービス**」をクリックします。
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。

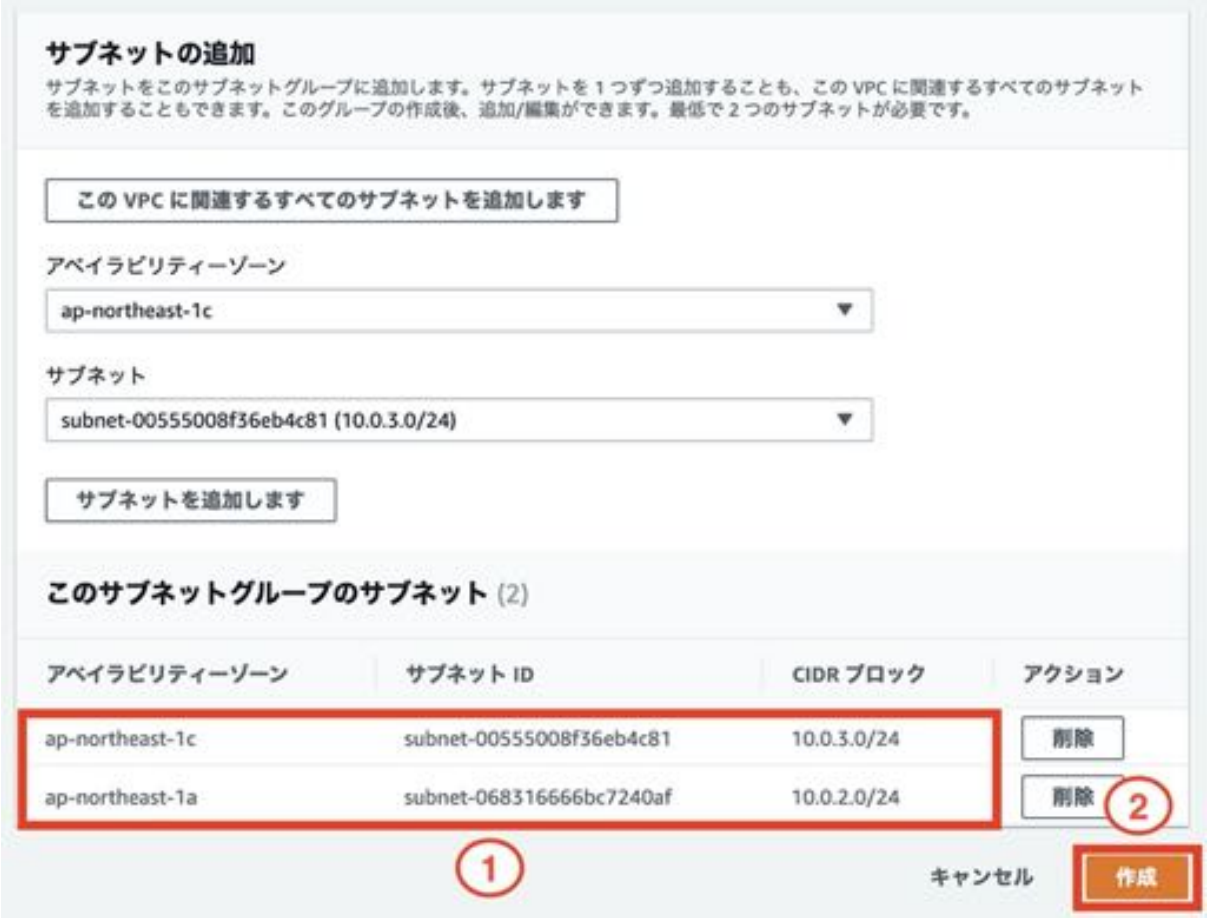
1. **「サブネットグループ」**をクリックします。
2. **「DB サブネットグループの作成」**をクリックします。

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

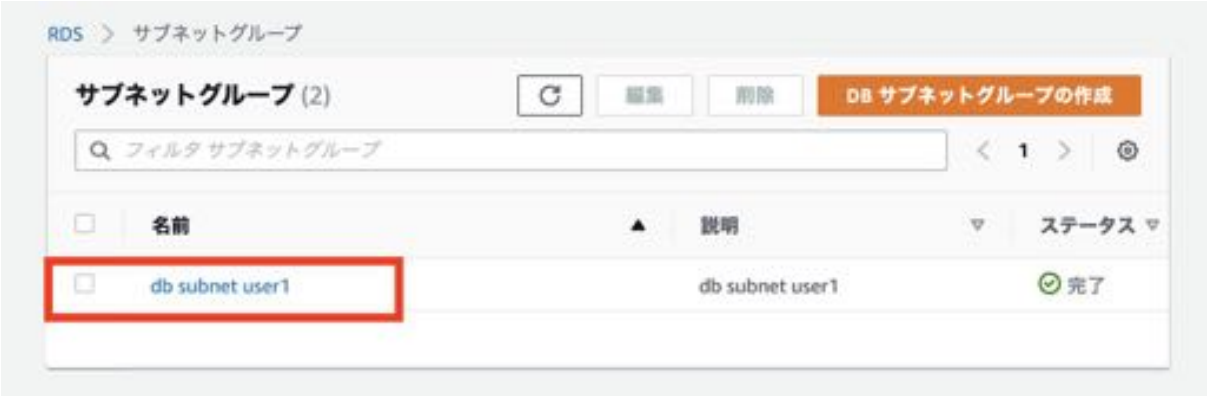
1. **「db subnet ユーザー名」**を入力します。 例）db subnet user1
2. **「RDS for MySQL」**などと入力します。
3. **フェーズ1-1-5** で作成した **VPC**を選択します。 例）[handson-user1]
4. **「ap-northeast-1a」**を選択します。
5. **「プライベートサブネット(10.0.2.0/24)」**を選択します。
6. **「サブネットを追加します」** をクリックします。

続けて、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-1c」**を選択します。
2. **「プライベートサブネット(10.0.3.0/24)」**を選択します。
3. **「サブネットを追加します」**をクリックします。

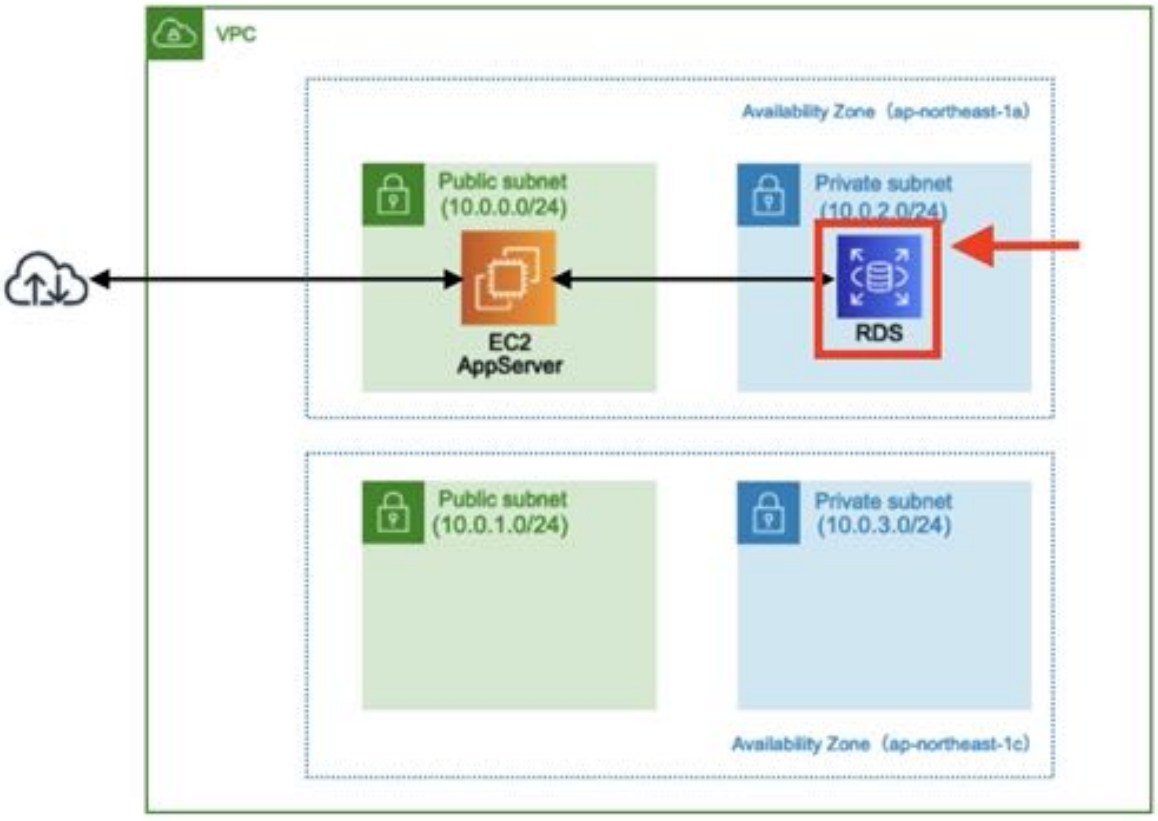


1. 異なるアベイラビリティゾーンにある 2 つのプライベートサブネットが追加されたことを確認します。
2. **「作成」**をクリックします。

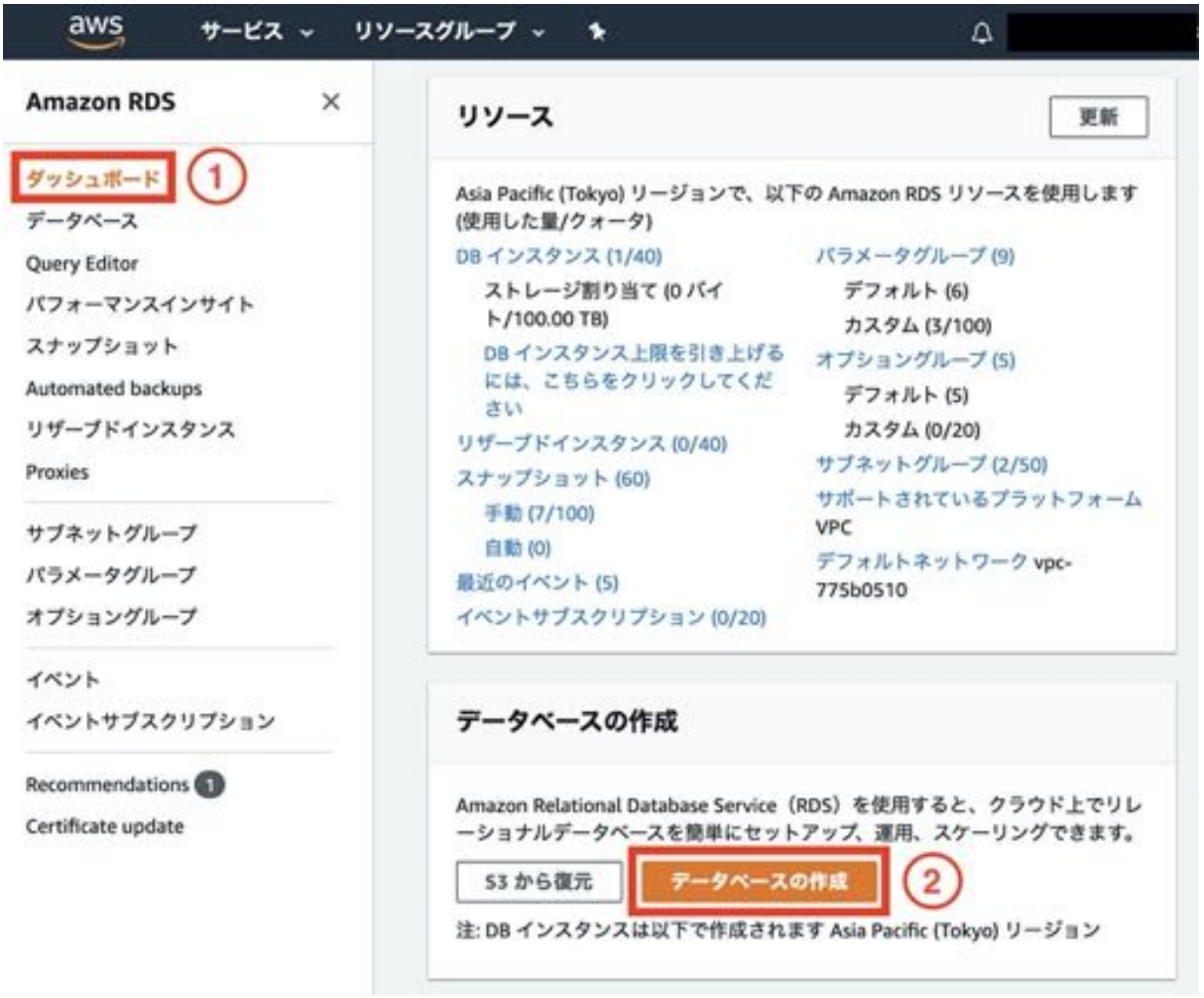


DBサブネットが作成されました。

▼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.t2.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. 「**追加の接続設定**」をクリックします。
3. 自動的に RDS サブネットグループが選択されます。
4. 「**なし**」を選択します。
5. 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。
6. 「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
7. 「**ap-northeast-1a**」を選択します。

[追加設定]

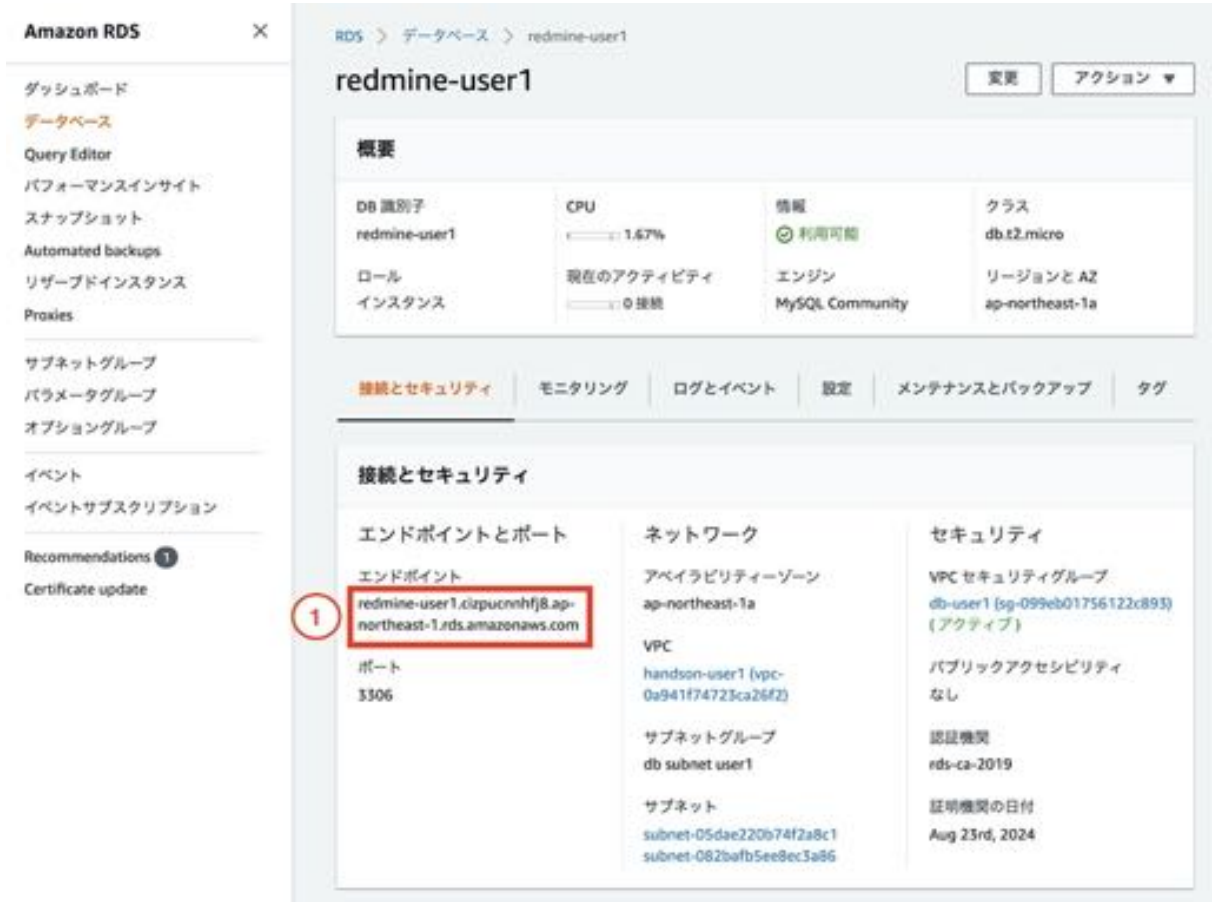
1. 「**追加設定**」をクリックします。
2. 「**0日間**」を選択します。
3. 「**データベースの作成**」をクリックします。

▼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

再度セッションマネージャーに接続をします。

1. 「**サービス**」→「**EC2**」→「**インスタンス**」をクリックして表示します。
2. インスタンス（例 webserver#1-user1）を選択して「**接続**」をクリックします。

1. 「**セッションマネージャー**」を選択します。
2. 「**接続**」をクリックします。

```
# root
sudo su
```

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。

```
# 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

引き続きセッションマネージャーで作業します。

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、デフォルトデータを登録します。
このコマンドにより、データベースがデフォルトの状態になるためプロジェクトなど何もない状態になります。

```
#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
#デフォルトデータを登録
bundle exec rake redmine:load_default_data RAILS_ENV=production
→ Select language: 「ja」 と入力
```

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
```

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

コンソールを操作します。
セッションマネージャー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。

1. 「**サービス**」をクリックします。
2. 「**EC2**」をクリックします。

先ほど**フェーズ1-3-4**で作成したインスタンスを選択します。

1. 「**インスタンス**」をクリックします。
2. **フェーズ1-3-4**で作成した**インスタンス**を選択します。
   例）[webserver#1-user1]
3. インスタンスを選択した状態で、「**接続**」をクリックします。

インスタンスに接続します。

1. 「**セッションマネージャー**」にチェックを入れます。
2. 「**接続**」をクリックします。

「**$**」が表示されたら以下を実行します。
表示されない場合は、右上の「終了」ボタンをクリックして一旦終了し、もう一度**接続し直して**ください。

```
# root
sudo su
```

```
# redmineのディレクトリに移動
cd /opt/bitnami/apps/redmine/htdocs/
```

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。

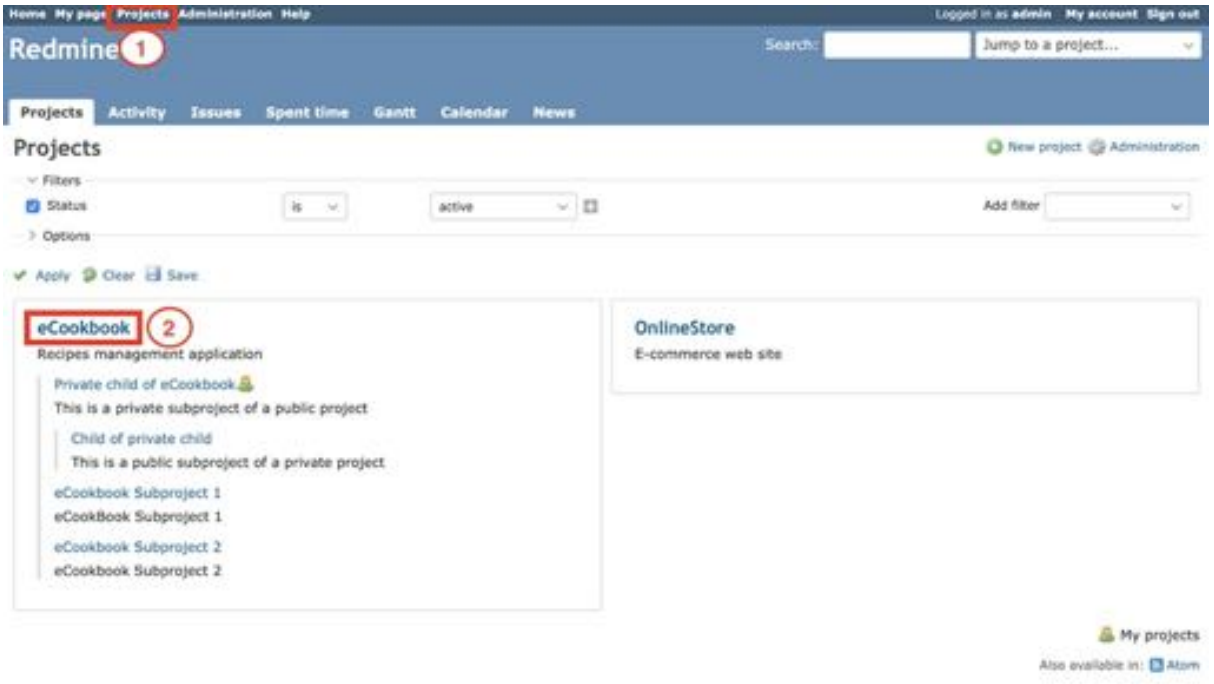
1. ブラウザで**http://<Elastic IPアドレス>/** にアクセスしてredmineを表示します。
2. Redmine画面右上の「**ログイン**」をクリックします。

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。

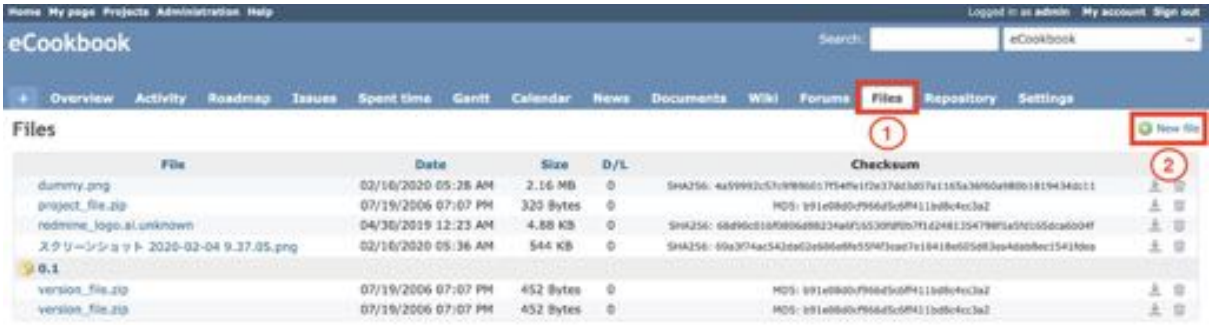
3. **ログイン**をクリックします。

Homeが表示されたらログイン成功です。



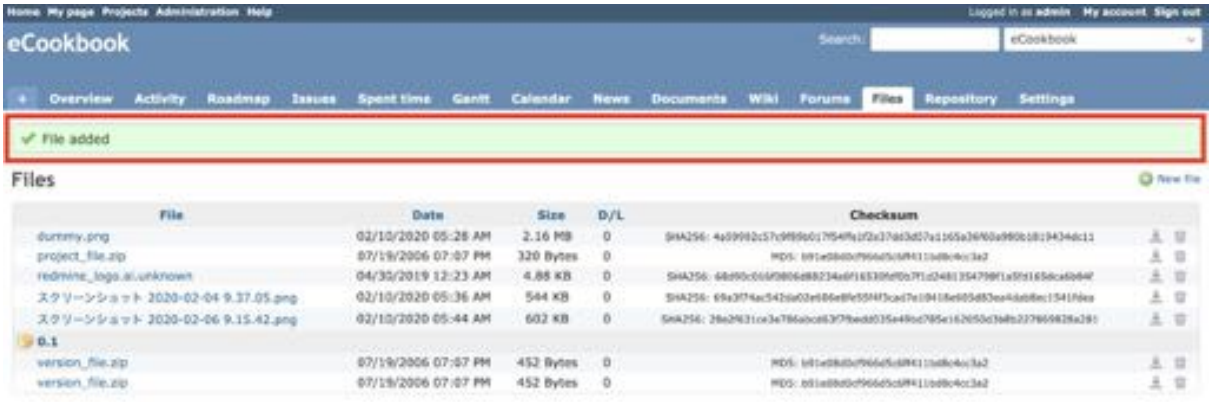
Redmineにログイン後、以下の手順でファイルをアップロードします。

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。

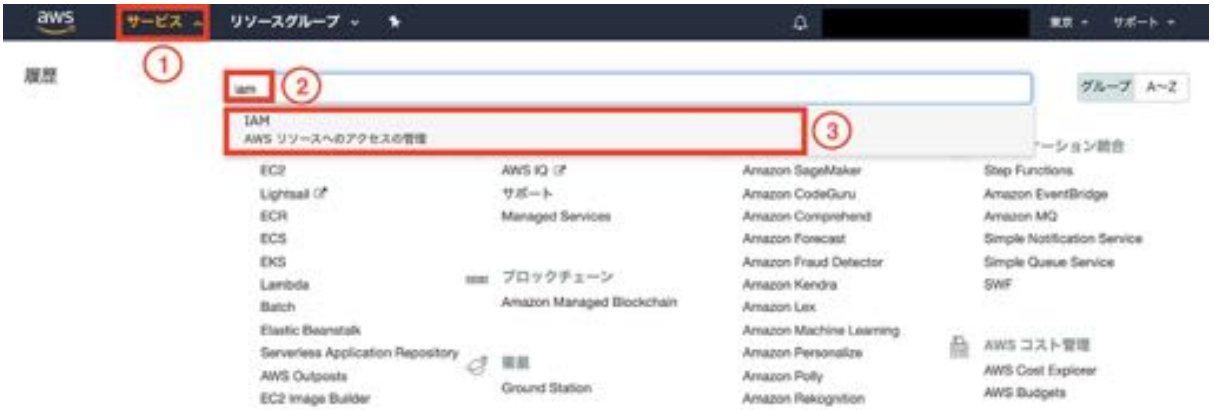
ステップ 2-5-3: ファイルのアップロード確認

ファイルがアップロードされているかをセッションマネージャーの中で確認します。

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。

```
# ファイルがあることを確認
ls files/2020/02/
→ 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



サービスからIAMを選択します。

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

3. 「**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-20200228**」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。

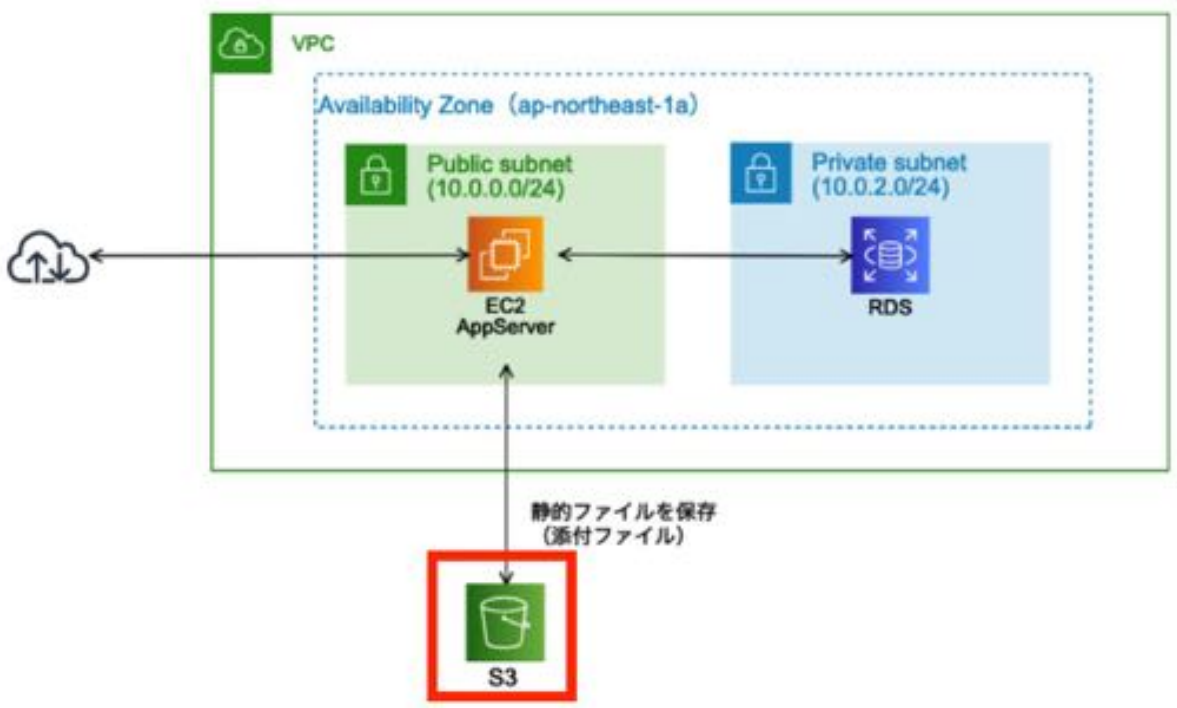
設定確認＆ユーザーの作成をします。

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



サービスからS3を表示します。

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

1. 「**バケットを作成する**」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-20200228**」と入力します。xxxxは自分の名前等を入力してください。例）redmine-user1-20200228
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

セッションマネージャーの中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-user1-20200228

```
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)-20200228
  folder: files
  thumb_folder: tmp/thumbnails
  region: ap-northeast-1
```

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

```
# 必要ライブラリーのインストール
bundle install --no-deployment
export AWS_REGION=ap-northeast-1
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。

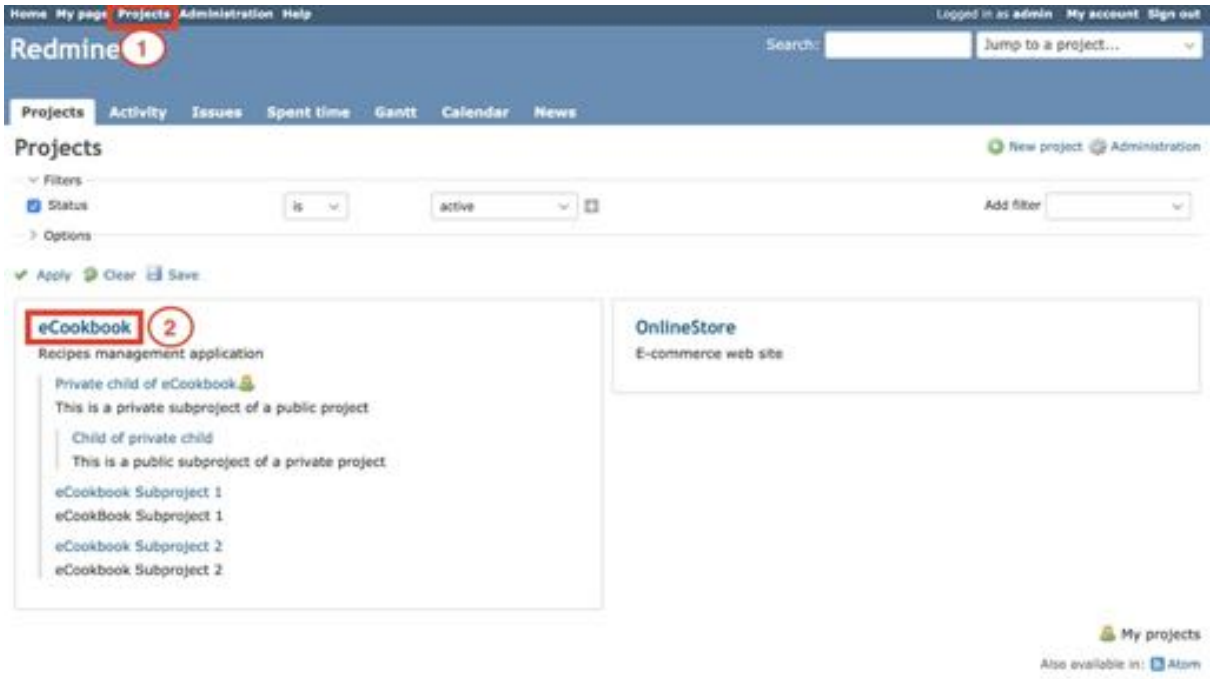
```
# apacheの停止
/opt/bitnami/apache2/scripts/ctl.sh stop

# 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
```

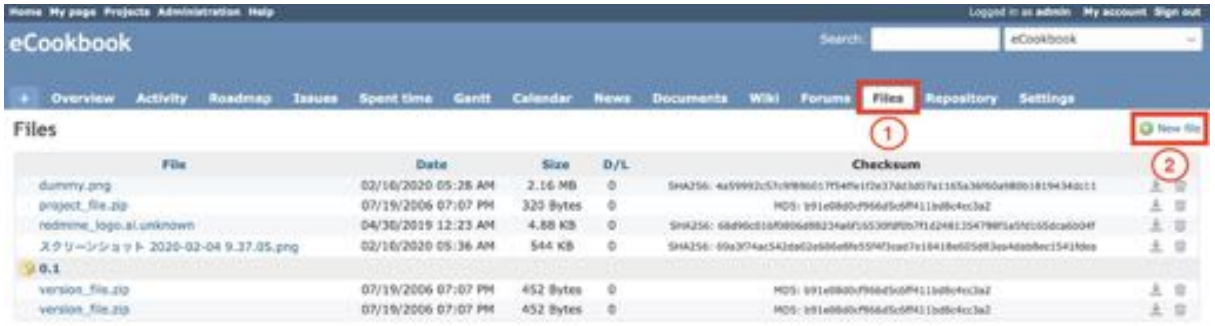
ステップ 2-5-8: 再度Redmineにファイルをアップロード

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。



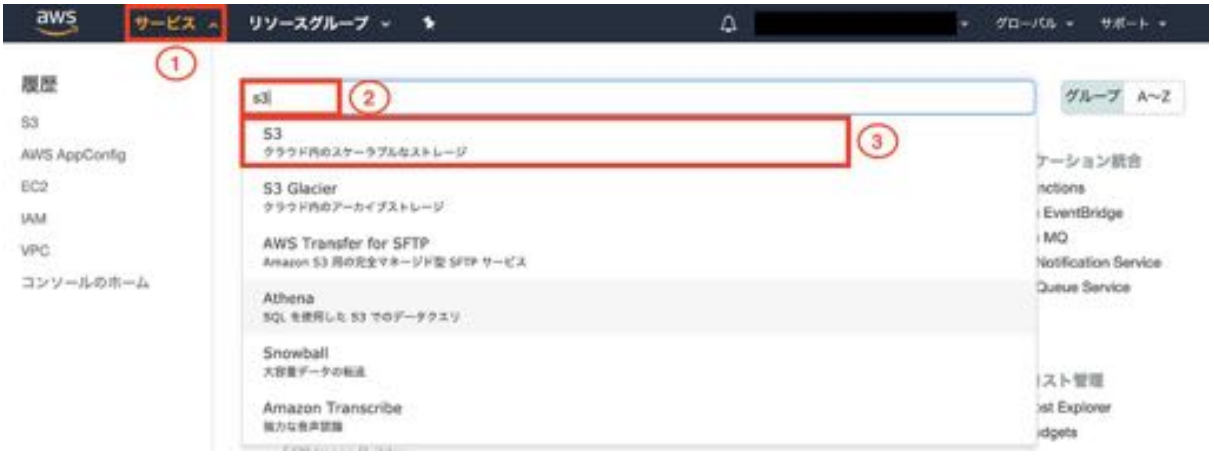
3. 「**Files**」をクリックします。
4. 「**New file**」をクリックします。

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。

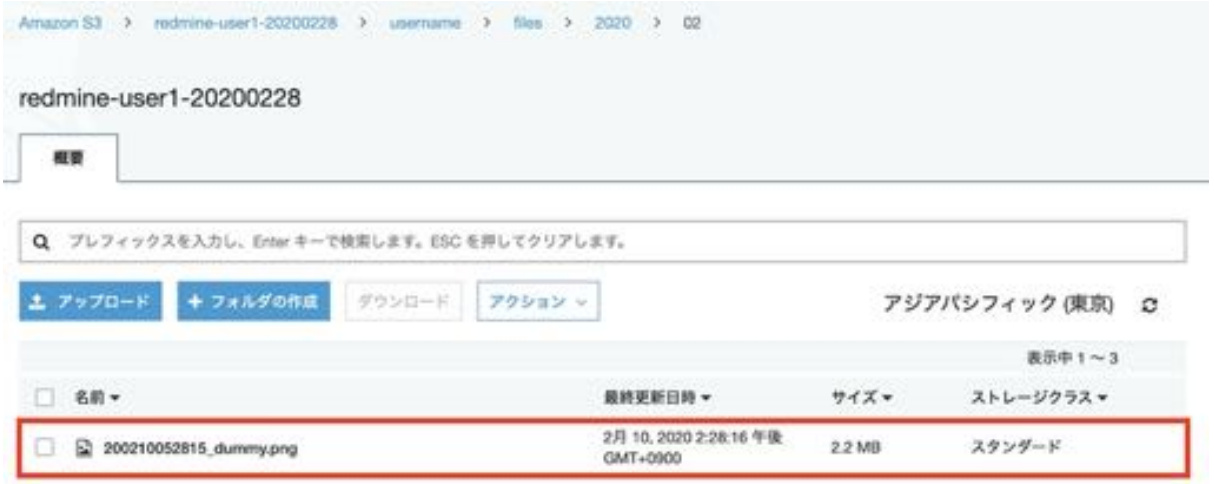


1. 「**サービス**」をクリックします。
2. 「**s3**」を入力します。

3. 「**S3**」をクリックします。

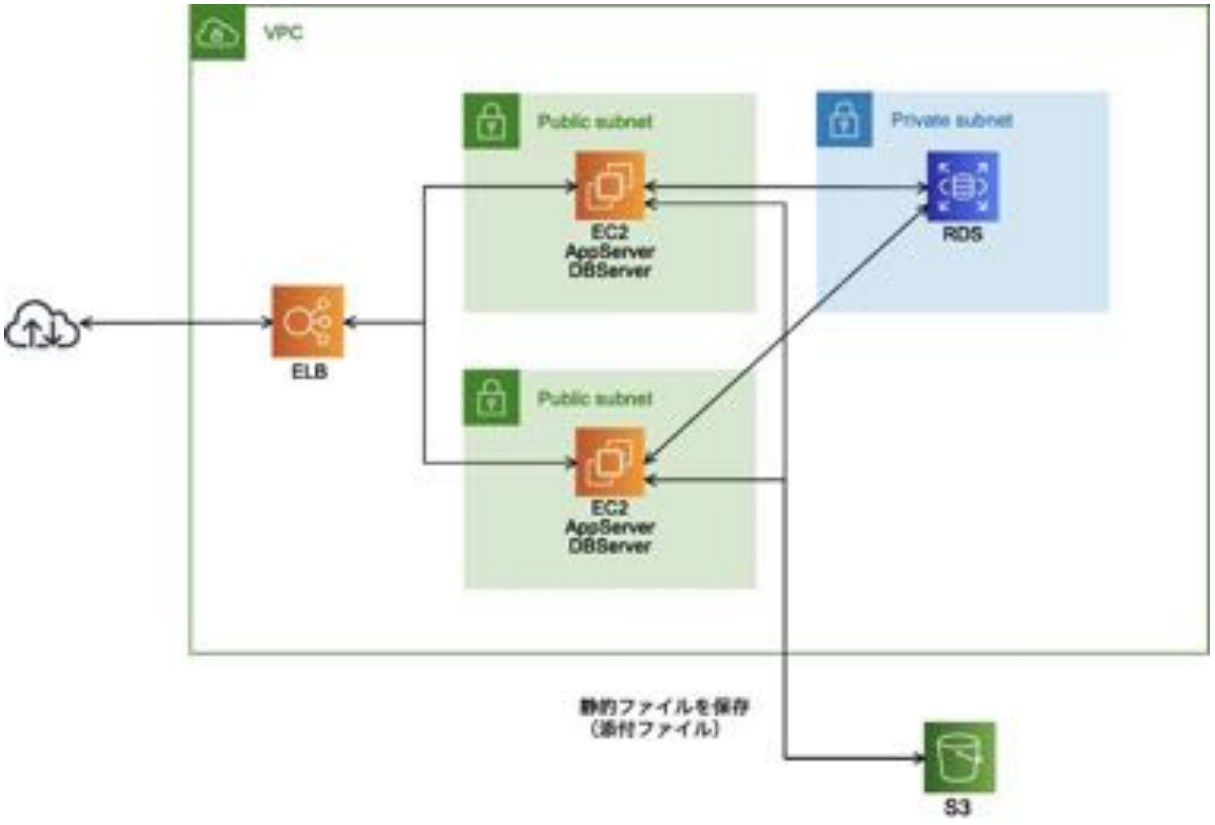
先ほど**フェーズ2-5-5**で作成したバケット名をクリックします。自分の名前などで検索をかけると見つけやすいです。

1. 自分の名前などで検索します。(日付でも〇)
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。



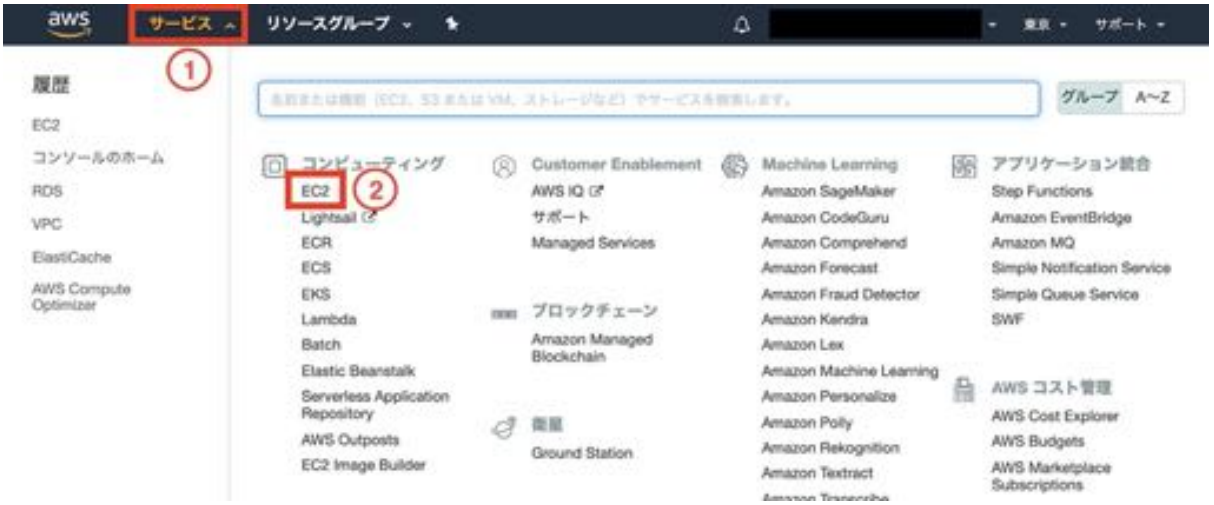
1. 「**files**」→「**2020**」→「**02**」をクリックして、先ほど**フェーズ2-5-2**でredmineにアップロードしたファイルが保存されていることを確認します

[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

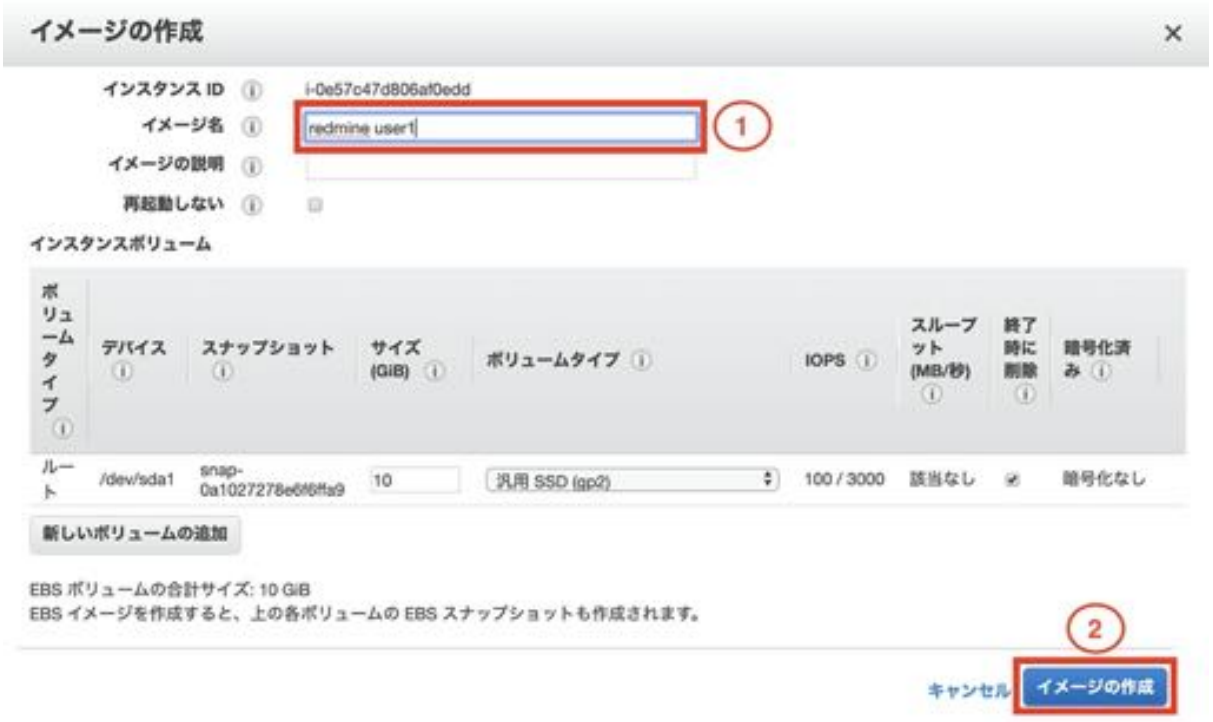


1. **「サービス」**をクリックします。
2. **「EC2」**をクリックします。

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-base-ユーザー名」**を右クリックします。

3. **「アクション」–「イメージ」–「イメージの作成」**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. **「イメージの作成」**をクリックします。

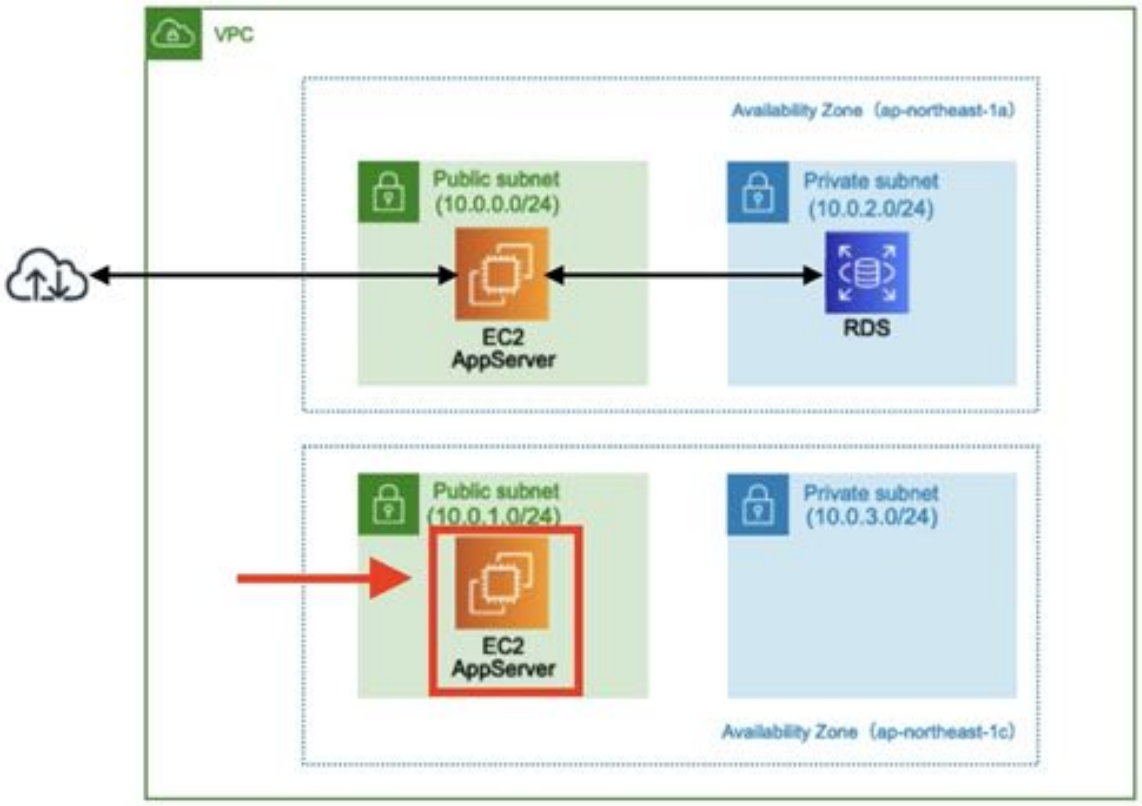
「保留中のイメージの表示」をクリックすることで、作成した AMI の状況だけが絞りこまれて表示されます。

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。

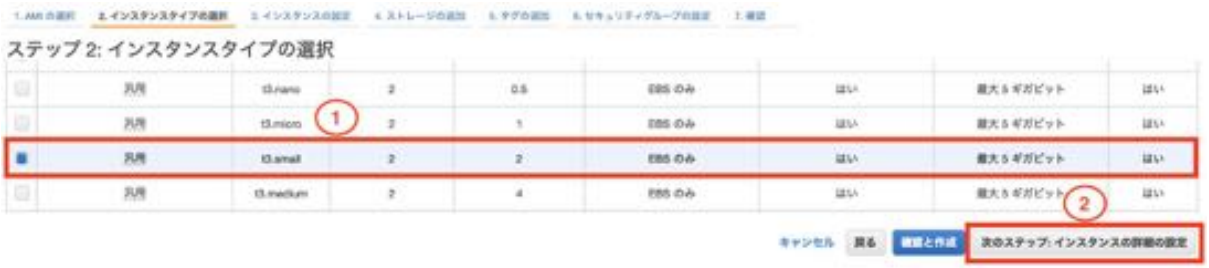
▼フェーズ 3-2: 2 個目の Amazon EC2 インスタンスを作成



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.small」**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. IAMロールは「**session-manager-20200228**」を選択します。
5. **「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。

警告が表示されますが、次へをクリックします。

1. 「**次へ**」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

キーペアはなしで続行します。

1. 「**キーペアなしで続行**」を選択します。
2. **「このAMIに組み込まれたパスワードがわからないと、このインスタンスに接続できないことを認識しています。」**にチェックを入れます。
3. 「**インスタンスの作成**」を選択します。

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

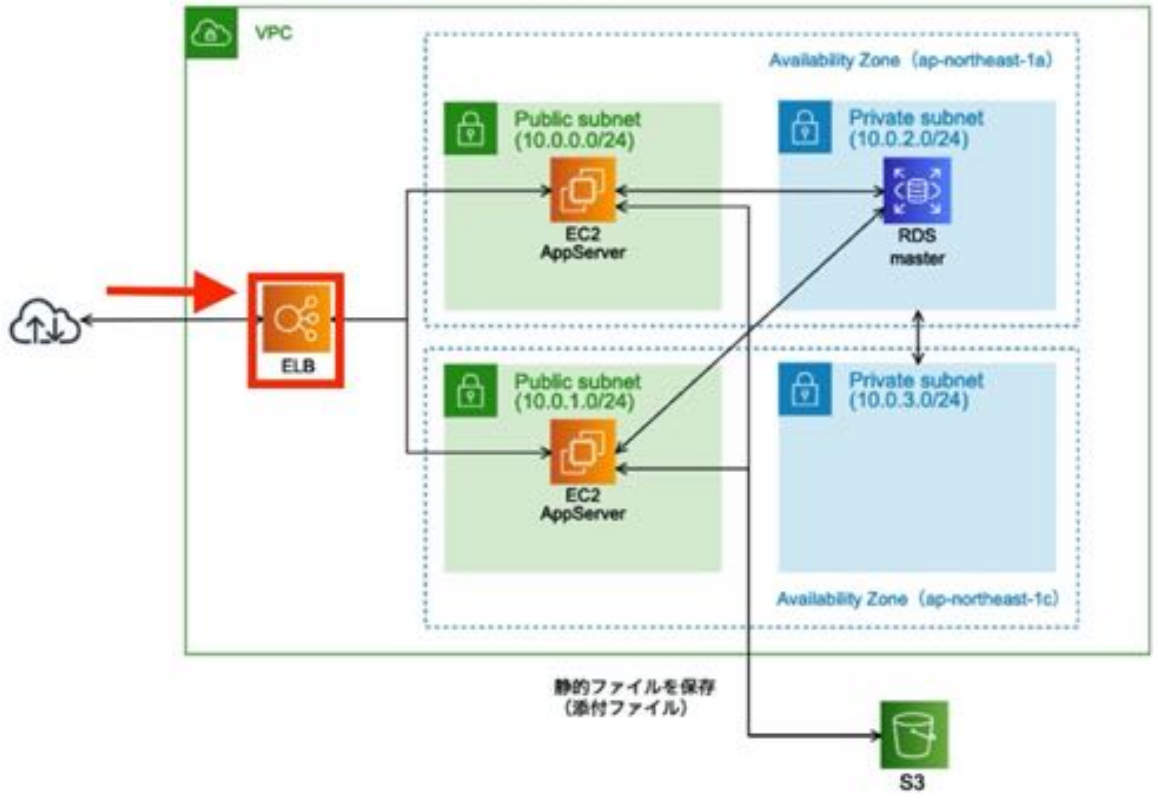
インスタンスの作成が完了するのに数分間かかります。

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。

▼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。
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: 右側のロードバランサーを選択

今回は**「Classic Load Balancer（標準ロードバランサー）」**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」**と入力します。
   例）elb-user1

2. **フェーズ1-1-5で作成した VPC を選択**します。
   例）handson-ユーザ名

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. **「10.0.0.0/24 パブリックサブネット」**の + をクリックします。
2. **「10.0.1.0/24 パブリックサブネット」**の + をクリックします。

1. **「パブリックサブネット」**のみであることを確認します。
2. **「アベイラビリティゾーン」**が 2 種類あることを確認します。
3. **「次の手順」**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. **「elb-ユーザ名」**と名前を入力します。
   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: セキュリティ設定の構成」**をクリックします。

ステップ 3-3-5: ELB を作成(3)

今回は SSL を使用しないため、何も設定せず次に進みます。

1. **「次の手順: ヘルスチェックの設定」**をクリックします。

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. 以下の設定に変更します。
   応答タイムアウトに「5」と入力します。
   間隔に「10」と入力します。
   非正常のしきい値で「2」を選択します。
   正常のしきい値で「2」を選択します。
3. **「次の手順: EC2インスタンスの追加」**をクリックします。

ステップ 3-3-7: ELB を作成(5)

HTTP アクセスの振り分け先として、WebServer 2 台を指定します。

1. **「webserver#1-ユーザ名」**と**「webserver#2-ユーザ名」**の 2 つを選択します。
2. **「次の手順: タグの追加」**をクリックします。

ステップ 3-3-8: ELB を作成（6）

1. キーに「**Name**」を入力します。
2. 値に「**elb-ユーザー名**」を入力します。例）elb-user1
3. 「**確認と作成**」をクリックします。

ステップ 3-3-9: ELB を作成（7）

設定内容を確認します。

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. **ホスト名**をメモします。

ELB 配下の 2 つの EC2 インスタンスが「In Service」と認識されると、正しく稼動できています。

1. **「インスタンス」**を選択します。
2. 状態が**「In Service」**に変わるのを確認します。

▼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 セッションマネージャー 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。

ec2画面へ移動します。

[**webserver#1**]

1. 「**インスタンス**」をクリックします。
2. 「**webserver#1−自分の名前(webserver#1−user1)**」を選択します。
3. 「**接続**」をクリックします。
4. 「**セッションマネージャー**」を選択します。
5. 「**接続**」をクリックします。

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/
# アクセスログを表示
$ tail -f log/production.log
```

[**webserver#2**]

1. 「**インスタンス**」をクリックします。
2. 「**webserver#2-自分の名前(webserver#2-user1)**」を選択します。
3. 「**接続**」をクリックします。
4. 「**セッションマネージャー**」を選択します。
5. 「**接続**」をクリックします。

以下のコマンドを実行します。

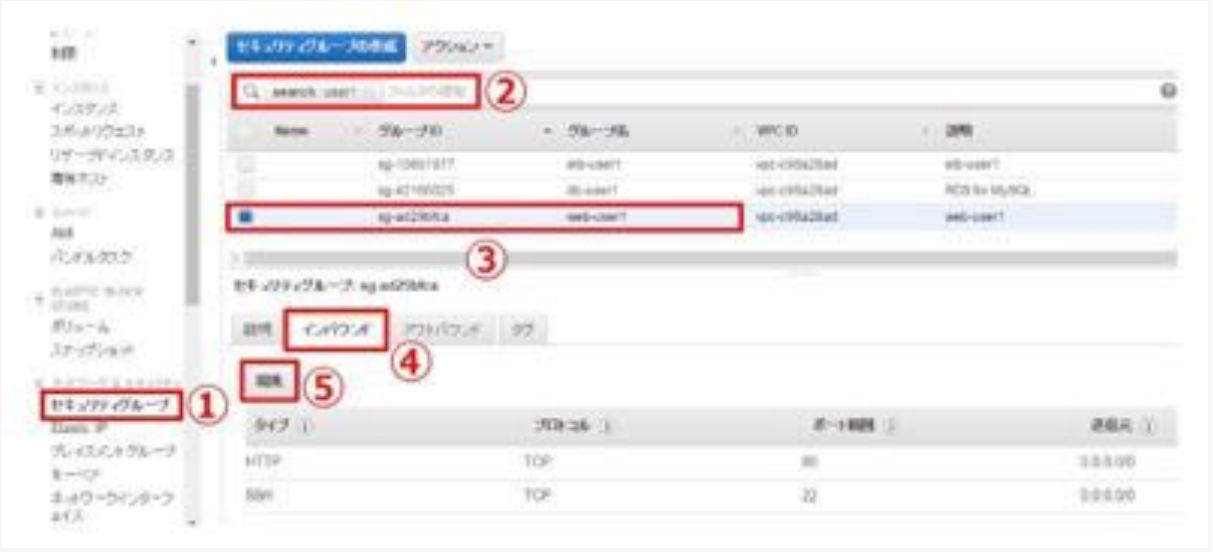
```
$ sudo su
# redmineのディレクトリに移動
$ cd /opt/bitnami/apps/redmine/htdocs/
# アクセスログを表示
$ tail -f log/production.log
```

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



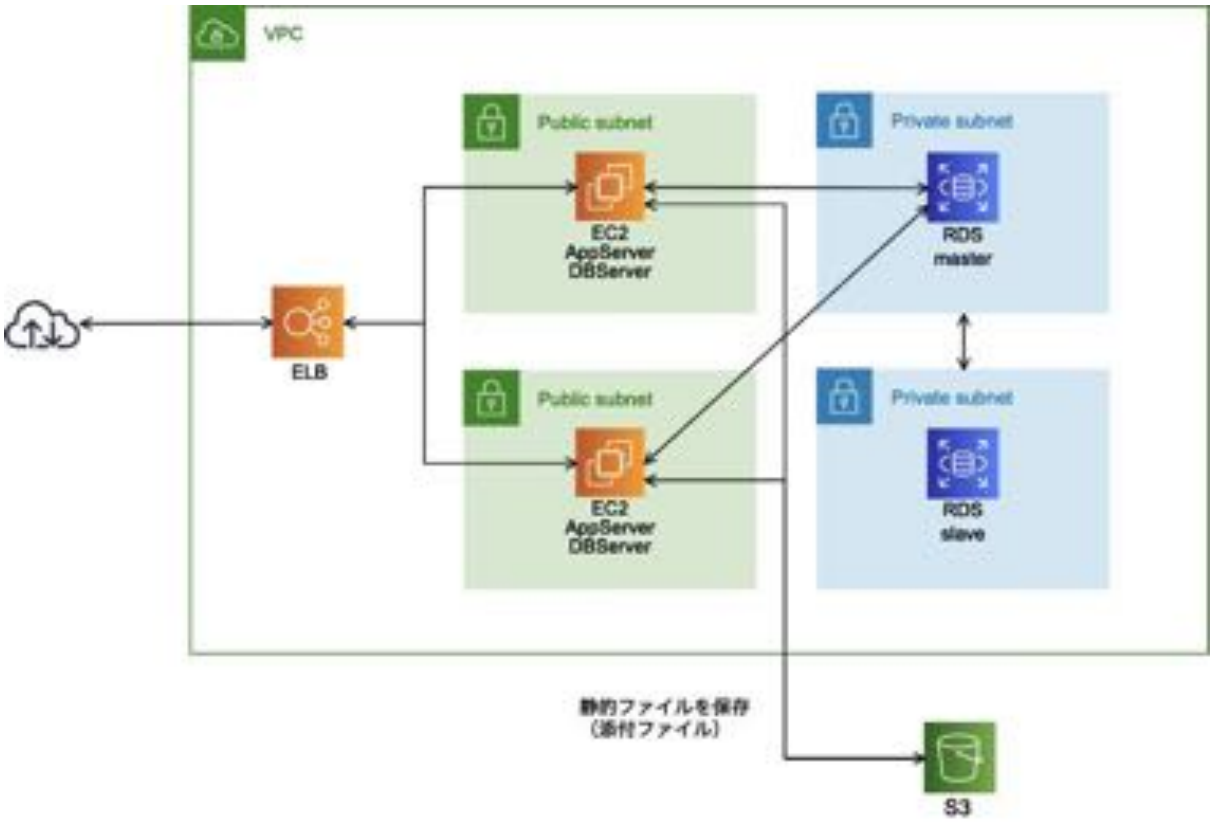
セキュリティグループの設定を変更し、Web サーバーへの HTTP アクセスは ELB からに限定するようにします。

1. **「セキュリティ グループ」**を クリックします。
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」**をクリックします。
5. **「編集」**をクリックします。

1. **「elb」**と入力して候補を表示させます。

2. 表示された候補から**「elb-ユーザー名」**を選択します。
3. **「保存」**をクリックします。

[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. **「サービス」**をクリックします。
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

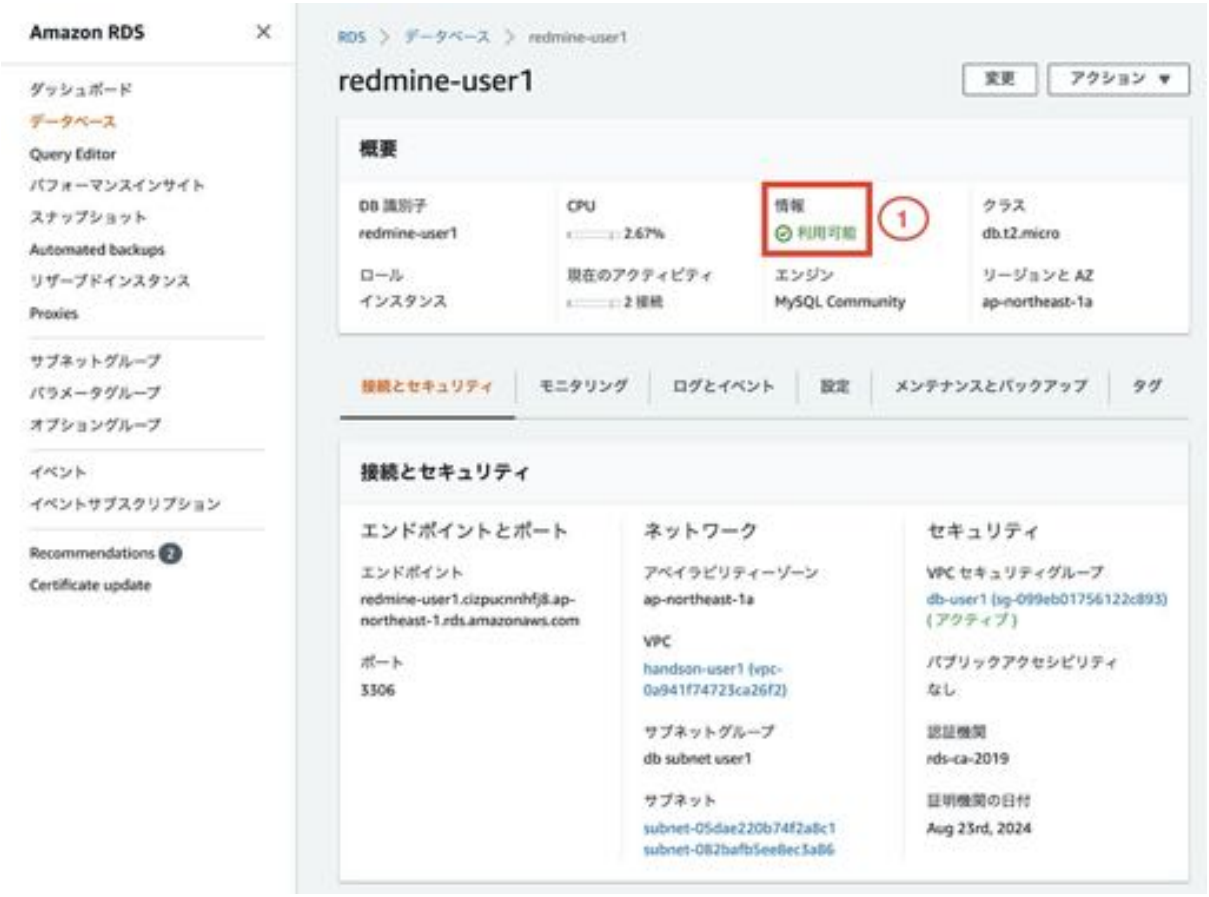
1. **「はい」**を選択します。
2. **「次へ」**をクリックします。

「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。

~ 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）-20200228**」を削除

1. 選択 -> 削除
2. バケット名を入力後、削除

[IAM]

＊ ユーザー

「**s3access-20200228**」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

＊ ロール

「**session-manager-20200228**」を削除

1. 選択 -> ロールの削除

[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

[ec2]

＊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

1. 選択 -> アクション -> セキュリティグループの削除

[VPC]

＊VPC

「**handson-自分の名前（user1）**」を削除

1. 選択 -> アクション -> 削除