

AWSハンズオン

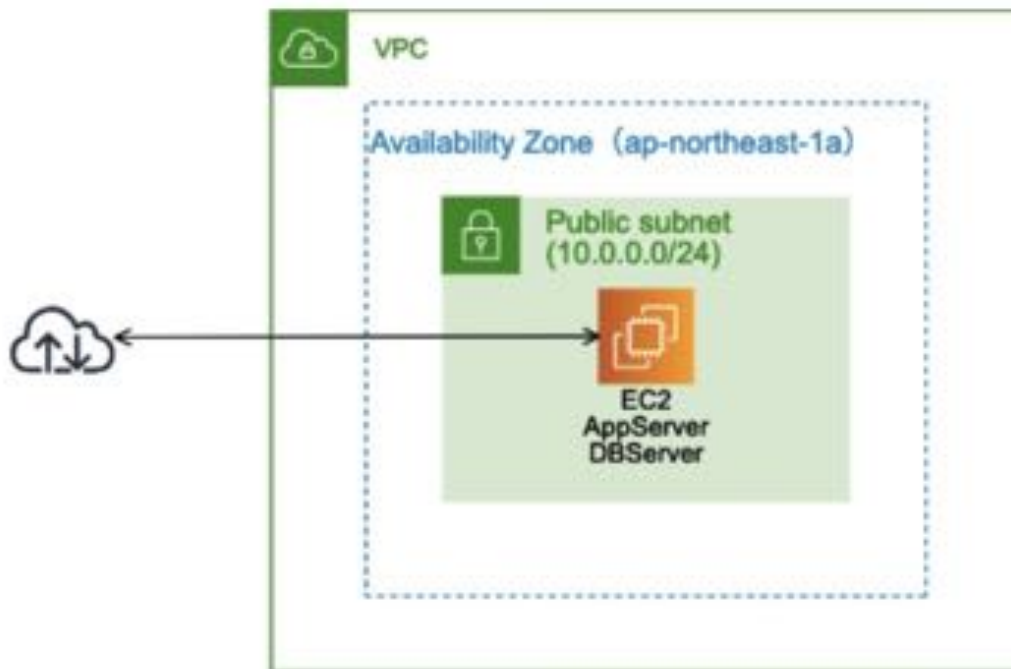
[フェーズ1]	
~サーバー 1 台構成で Redmine 環境を構築~	4
▼フェーズ 1-1: コンソールへのログインと VPC (ネットワーク) の作成	4
▼フェーズ 1-2: サブネットを追加作成	15
▼フェーズ 1-3: Amazon EC2 インスタンスの作成	19
▼フェーズ 1-4: Elastic IP (固定 IP) の割り当て	32
[フェーズ 2]	
~拡張性を向上しつつDB 運用負荷を軽減する構成を構築~	37
▼フェーズ 2-1: Amazon RDS のセキュリティグループを作成	37
▼フェーズ 2-2: DB サブネットグループを作成	40
▼フェーズ 2-3: Amazon RDS インスタンスを作成	45
▼フェーズ 2-4: RDSに接続	52
▼フェーズ 2-5: Redmine S3対応	57
[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~	75
▼フェーズ 3-1: Web サーバーの AMI (パッケージ) を作成	75
▼フェーズ 3-2: 2 個目の Amazon EC2 インスタンスを作成	79
▼フェーズ 3-3: Elastic Load Balancing (ロードバランサー) を作成	85
▼フェーズ 3-4: Elastic Load Balancing 経由でアクセス	96
▼フェーズ 3-5: セキュリティグループ設定変更	99
[フェーズ 4]	
~ Amazon RDS を Multi-AZ 構成に変更 ~	104
▼フェーズ 4: Amazon RDS を Multi-AZ 構成に変更	104
~ 構築した環境の後片付け ~	112

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

The screenshot shows the AWS Management Console interface. At the top right, the current region is set to '東京' (Tokyo), which is highlighted with a red box and a circled '1'. A dropdown menu is open, showing a list of regions. The region 'アジアパシフィック (東京) ap-northeast-1' is highlighted with a red box and a circled '2'. The main content area shows 'AWS のサービス' (AWS Services) and 'ソリューションの構築' (Build Solutions) sections.

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。

ステップ 2: 1 個のパブリックサブネットを持つ VPC

IPv4 CIDR ブロック: ① (65531 利用可能な IP アドレス)

IPv6 CIDR ブロック: IPv6 CIDR ブロックなし
 Amazon が提供した IPv6 CIDR ブロック

VPC 名: ②

パブリックサブネットの IPv4 CIDR: ③ (251 利用可能な IP アドレス)

アベイラビリティゾーン: ④

サブネット名:

VPC を作成した後は、より多くのサブネットを追加できます。

サービスエンドポイント

DNS ホスト名を有効化: はい いいえ

ハードウェアのテナンシー:

⑤

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。

- 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

The screenshot shows the AWS VPC console interface. On the left, the 'Virtual Private Cloud' menu is visible, with 'サブネット' (Subnets) highlighted and circled with a red '1'. The main area displays a table of subnets. The first row is selected, with a blue square icon circled with a red '2'. The 'IPv4 CIDR' column for this subnet is '10.0.0.0/24', circled with a red '3'. Below the table, the details for the selected subnet 'subnet-02ad147fa93cbe2c1' are shown. The 'アベイラビリティゾーン' (Availability Zone) is 'ap-northeast-1a', circled with a red '4'. Other details include VPC ID 'vpc-075ecbe4fa77857ea', status 'available', and owner '533384410763'.

Name	サブネット ID	状態	VPC	IPv4 CIDR	利用可能な IPv4	IPv6 CIDR
パブリック...	subnet-02ad147fa93cbe2c1	available	vpc-075ecbe4fa77857ea [...]	10.0.0.0/24	251	-

説明	フローログ	ルートテーブル	ネットワーク ACL	タグ	共有
サブネット ID	subnet-02ad147fa93cbe2c1	状態	available	IPv4 CIDR	10.0.0.0/24
VPC	vpc-075ecbe4fa77857ea handson-user1	IPv6 CIDR	-	ルートテーブル	rtb-03c1b7145c3d081e8
利用可能な IPv4 アドレス	251	デフォルトのサブネット	いいえ	IPv6 アドレスの自動割り当て	いいえ
アベイラビリティゾーン	ap-northeast-1a	所有者	533384410763		
ネットワーク ACL	acl-007f9ece39c600083				
パブリック IPv4 アドレスの自動割り当て	いいえ				
Outpost ID	-				

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

The screenshot shows the AWS VPC console interface. On the left, there is a navigation menu with options like 'サブネット', 'ルートテーブル', and 'インターネットゲートウェイ'. The main area displays a list of subnets, with the selected subnet 'subnet-02ad147fa93cbe2c1' highlighted. Below this, the 'ルートテーブル' (Route Tables) tab is active, showing the configuration for 'ルートテーブル: rtb-03c1b7f45c3d081e8'. A table of routes is visible, with two entries: a default route (0.0.0.0) pointing to 'local' and a route for 10.0.0.0/16 pointing to an Internet Gateway (igw-002a4a076f96873f). Red boxes and numbers 1 and 2 highlight the 'ルートテーブル' tab and the route table content, respectively.

送信先	ターゲット
10.0.0.0/16	local
0.0.0.0	igw-002a4a076f96873f

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。



▼ 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

VPC ダッシュボード

VPC でフィルタリング:

サブネットの作成 アクション

タグや属性によるフィルター、またはキーワードによる検索

Name	サブネット ID	状態	VPC	IPv4 CIDR	利用可能な IPv4	IPv6 CIDR
パブリック...	subnet-0bdeae1e233de5ee2	available	vpc-047f4caec88c5613 ...	10.0.0.0/24	251	-
パブリック...	subnet-0d33bb62d8787124b	available	vpc-047f4caec88c5613 ...	10.0.1.0/24	251	-
プライベート...	subnet-59dc92e84b57d72bd	available	vpc-047f4caec88c5613 ...	10.0.2.0/24	251	-
プライベート...	subnet-03a01402d0edbc1cf	available	vpc-047f4caec88c5613 ...	10.0.3.0/24	251	-

Virtual Private Cloud

VPC

サブネット

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

VPC ダッシュボード
VPC でフィルタリング:
vpc-047f4caec8f8c5613
handson-user1
所有者:
Virtual Private Cloud
VPC
サブネット
ルートテーブル
インターネットゲートウェイ

サブネットの作成 アクション

タグや属性によるフィルター、またはキーワードによる検索

Name	サブネット ID	状態	VPC	IPv4 CIDR
パブリック...	subnet-0bdeae1e233de5ee2	available	vpc-047f4caec8f8c5613 ...	10.0.0.0/24
パブリック...	subnet-0d33bb82d8787124b	available	vpc-047f4caec8f8c5613 ...	10.0.1.0/24
プライベート...	subnet-09dd92e84b07d72bd	available	vpc-047f4caec8f8c5613 ...	10.0.2.0/24
プライベート...	subnet-03a01402d0edbc1cf	available	vpc-047f4caec8f8c5613 ...	10.0.3.0/24

サブネット: subnet-0d33bb82d8787124b

説明 フローログ ルートテーブル ネットワーク ACL タグ 共有

ルートテーブルの関連付けの編集

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。

サブネット > ルートテーブルの関連付けの編集

ルートテーブルの関連付けの編集

サブネット ID: subnet-0d33bb82d8787124b

ルートテーブル ID: rtb-0d1c9a2b55b2aa3f9

2 中の 1 ~ 2

送信元	ターゲット
10.0.0.0/16	local
0.0.0.0/0	igw-061c06c43d4f95f87

*必須

キャンセル 保存

1. これまでと異なるものを選択してください。
※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。

ロールの作成 1 2 3 4

信頼されたエンティティの種類を選択

AWS サービス
EC2、Lambda、およびその他

1

別の AWS アカウント
お客様またはサードパーティーに属しています

ウェブ ID
Cognito または任意の OpenID プロバイダ

SAML 2.0 フェデレーション
企業ディレクトリ

AWS のサービスによるアクションの代行を許可します。 [詳細はこちら](#)

ユースケースの選択

一般的なユースケース

EC2
Allows EC2 instances to call AWS services on your behalf.

2

Lambda
Allows Lambda functions to call AWS services on your behalf.

3

キャンセル 次のステップ: アクセス権限

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。

ロールの作成 1 2 3 4

▼ Attach アクセス権限ポリシー

新しいロールにアタッチするポリシーを1つ以上選択します。

ポリシーの作成 🔄

ポリシーのフィルタ 1 1件の結果を表示中

	ポリシー名	次として使用
2	<input checked="" type="checkbox"/> AmazonEC2RoleforSSM	Permissions policy (2)

キャンセル 戻る 次のステップ: タグ

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。

ロールの作成 1 2 3 4

タグの追加 (オプション)

IAM タグは、ロール に追加できるキーと値のペアです。タグには、Eメールアドレスなどのユーザー情報を含めるか、役職などの説明文とすることができます。タグを使用して、この ロール のアクセスを整理、追跡、制御できます。 [詳細はこちら](#)

キー	値 (オプション)	削除
Name	session-manager-20200228	✕
<input type="text" value="新しいキーを追加"/>	<input type="text"/>	

さらに 49 個のタグを追加できます。

キャンセル 戻る 次のステップ: 確認

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。

ロールの作成

1 2 3 4

確認

以下に必要な情報を指定してこのロールを見直してから、作成してください。

ロール名 ①
英数字と「+、@、_」を使用します。最大 64 文字。

ロールの説明
最大 1000 文字。英数字と「+、@、_」を使用します。

信頼されたエンティティ AWS のサービス: ec2.amazonaws.com

ポリシー  AmazonEC2RoleforSSM [🔗](#)

アクセス権限の境界 アクセス権限の境界が設定されていません

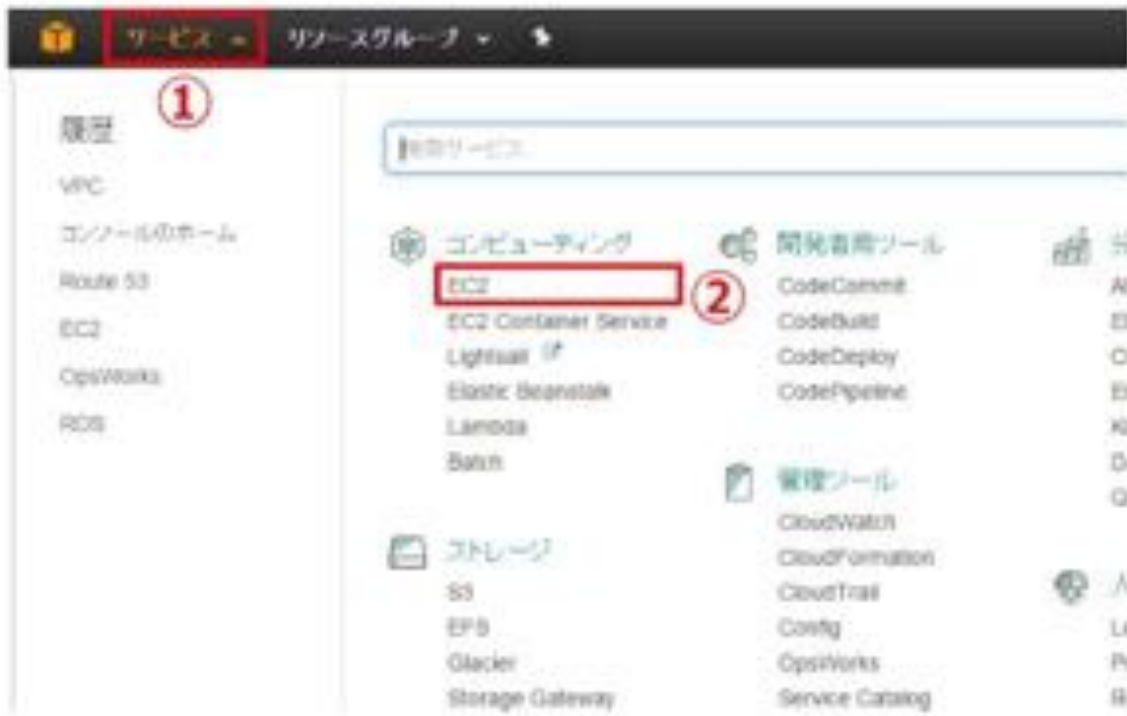
キャンセル

戻る

②
ロールの作成

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。

Redmine Certified by Bitnami



無料利用枠の対象

Redmine Certified by Bitnami
Redmine is a project management and issue tracking platform. It enables teams to manage multiple projects from a single user interface. This solution provides enterprise-grade features such as LDAP user access management, multiple database support, and bug tracking tools. It is fully integrated with Git and Mercurial.

This image is configured ... [詳細](#)

[AWS Marketplace での詳細の表示](#)

製品の詳細

担当 Bitnami
お客様による評価 ★★★★★ (19)
最新バージョン 4.1.0-0 on Ubuntu 16.04
基本オペレーティングシステム Linux/Unix, Ubuntu 16.04
実施形式 64 ビット Amazon マシンイメージ (AMI) x86
ライセンス契約 エンドユーザーライセンス契約
Marketplace での使用開始日 2016/10/28

ハイライト

- Manage and track multiple projects, with a separate document manager, wiki, calendar, Ganit charts, forums, and time tracking for each one. Create custom fields by project for bugs, time tracking, and users.

料金に関する詳細情報

時間料金

インスタンスタイプ	ソフトウェア	EC2	合計
t2.micro	\$0.00	\$0.015	\$0.015/時間
t2.small	\$0.00	\$0.03	\$0.03/時間
t2.medium	\$0.00	\$0.061	\$0.061/時間
t2.large	\$0.00	\$0.122	\$0.122/時間
t2.xlarge	\$0.00	\$0.243	\$0.243/時間
t2.2xlarge	\$0.00	\$0.486	\$0.486/時間
t3a.micro	\$0.00	\$0.012	\$0.012/時間
t3a.small	\$0.00	\$0.025	\$0.025/時間
t3a.medium	\$0.00	\$0.049	\$0.049/時間
t3a.large	\$0.00	\$0.098	\$0.098/時間
t3a.xlarge	\$0.00	\$0.196	\$0.196/時間

キャンセル Continue 1

1. 「Continue」をクリックします。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの設定 5. タグの設定 6. セキュリティグループの設定 7. 確認

ステップ 2: インスタンスタイプの選択

選択	名前	OS	コア	メモリ	ディスク	インターネット接続	最大 I/O	最大 IOPS
<input type="checkbox"/>	t3.nano	Linux	2	0.5	16 GB のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	t3.micro	Linux	2	1	16 GB のみ	はい	最大 5 ギガビット	はい
<input checked="" type="checkbox"/>	t3.small	Linux	2	2	16 GB のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	t3.medium	Linux	2	4	16 GB のみ	はい	最大 5 ギガビット	はい

キャンセル 戻る 確認と作成 次のステップ: インスタンスの詳細の設定 2

- 「t3.small」を選択します。
- 「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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 3: インスタンスの詳細の設定

要件に合わせてインスタンスを設定します。同じAMIからの複数インスタンス作成や、より低料金を実現するためのスポットインスタンスのリクエスト、インスタンスへのアクセス管理ロール割り当てなどを行うことができます。

インスタンス数 ① 1 Auto Scaling グループを作成する ②

購入のオプション ③ スポットインスタンスのリクエスト

ネットワーク ④ vpc-9cc56234af4649696 | handson-ase1 ① 1. VPC の作成

サブネット ④ subnet-08c7d0849e1e533a | プライベートサブネット ② 2. サブネットの作成

自動割り当てパブリックIP ④ 有効 ③ 3.

放置グループ ④ インスタンスをプレースメントグループに追加します。

キャパシティの手続 ④ 置く ④ 新しいキャパシティの手続の作成

IAM ロール ④ session-manager-20200228 ④ 4. IAM ロールの作成

キャンセル 戻る 確認と作成 ⑤ 次のステップ: ストレージの追加

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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 4: ストレージの追加

インスタンスは次のストレージデバイス設定を使用して作成されます。インスタンスに追加の EBS ボリュームやインスタンスストアボリュームをアタッチするか、ルートボリュームの設定を編集することができます。また、インスタンスを作成してから追加の EBS ボリュームをアタッチすることもできますが、インスタンスストアボリュームはアタッチできません。Amazon EC2 のストレージオプションに関する [詳細](#) はこちらをご覧ください。

ボリュームタイプ ①	デバイス ①	スナップショット ①	サイズ (GiB) ①	ボリュームタイプ ①	IOPS ①	スループット (MB/秒) ①	終了時に削除 ①	暗号化 ①
ルート	/dev/xvda	snap-0a3a21e764482ce15	8	汎用 SSD (gp2)	100 / 3000	該当なし	<input checked="" type="checkbox"/>	暗号化 <input type="checkbox"/>

新しいボリュームの追加

キャンセル 戻る 確認と作成 ① 次のステップ: タグの追加

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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 5: タグの追加

タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、キーに「Name」、値に「Webserver」を使用してタグを定義することができます。タグのコピーは、ボリューム、インスタンス、またはその両方に適用できます。タグは、すべてのインスタンスとボリュームに適用されます。Amazon EC2 リソースのタグ付けに関する [詳細はこちら](#)。

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. 「webserve#1- ユーザー名」とします。
例) [webserve#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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 6: セキュリティグループの設定

セキュリティグループは、インスタンスのトラフィックを制御するファイアウォールのルールセットです。このページで、特定のトラフィックに対してインスタンスへの到達を許可するルールを追加できます。たとえば、ウェブサーバーをセットアップして、インターネットトラフィックにインスタンスへの到達を許可する場合、HTTP および HTTPS ポートに無制限のアクセス権限を与えます。新しいセキュリティグループを作成するか、次の既存のセキュリティグループから選択することができます。Amazon EC2 セキュリティグループに関する [詳細はこちら](#)。

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 ①
 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: ②
 説明:

タイプ ①	プロトコル ①	ポート範囲 ①	ソース ①	説明 ①
HTTP ③	TCP	80	任意の場所 ④ 0.0.0.0/0 ::0	例: SSH for Admin Desktop
SSH	TCP	22	カスタム CIDR, IP またはセキュリティグループ	例: SSH for Admin Desktop ⑤

ルールの追加

キャンセル 戻る 確認と作成

「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。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 6: セキュリティグループの設定

セキュリティグループは、インスタンスのトラフィックを制御するファイアウォールのルールセットです。このページで、特定のトラフィックに対してインスタンスへの到達を許可するルールを追加できます。たとえば、ウェブサーバーをセットアップして、インターネットトラフィックにインスタンスへの到達を許可する場合、HTTP および HTTPS ポートに無制限のアクセス権限を与えます。新しいセキュリティグループを作成するか、次の既存のセキュリティグループから選択することができます。Amazon EC2 セキュリティグループに関する [詳細はこちら](#)。

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
 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:
 説明:

タイプ (1)	プロトコル (1)	ポート範囲 (1)	ソース (1)	説明 (1)
HTTP	TCP	80	任意の場所	例: SSH for Admin Desktop

ルールの追加

警告
 AMI では、アクセスを可能にするためにポート 22 を開く必要があるため、このインスタンスに接続できません。現在のセキュリティグループでは、ポート 22 が開いていません。

キャンセル 戻る **確認と作成**

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。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更を行うことができます。[作成] をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

警告 インスタンスのセキュリティを強化してください。セキュリティグループ `web-user1` は世界に向けて開かれています。このインスタンスには、どの IP アドレスからもアクセスできる可能性があります。セキュリティグループのルールを更新して、適切な IP アドレスからのみアクセスできるようにすることを検討します。また、セキュリティグループの追加ポートを開いて、実行中のアプリケーションやサービスへのアクセスを容易にすることもできます。たとえば、ウェブサーバー用に HTTP (80) を開きます。 [セキュリティグループの構築](#)

警告 お客様のインスタンス設定は無料利用枠の対象ではありません。無料利用枠の対象であるインスタンスを起動するには、選択している AMI、インスタンスタイプ、設定オプション、ストレージデバイスをチェックします。無料利用枠 の利用枠と使用制限に関する詳細情報をご覧ください。 [詳細はこちらを確認してください](#)

AMI の詳細 AMI の編集

Redmine Certified by Bitnami
 This image may not be the latest version available and might include security vulnerabilities. Please check the latest, up-to-date, available version at <https://bitnami.com/stacks>.
 ルートデバイスタイプ: x86_64 64bit アーキテクチャ

ソフトウェアの著作権料: 30.001 時間あたり 1.50 USD インスタンス 追加の料金または料金が適用される場合があります。この AMI から作成するとソフトウェアの課金が発生し、インスタンスを終了するまで続きます。

この製品を起動することで、このソフトウェアにサブスクリプションし、このソフトウェアの使用により料金表条件と販売者の条件に同意することになるものとします。
[エンドユーザーライセンス契約](#)

インスタンスタイプ インスタンスタイプの編集

キャンセル 戻る **起動**

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

既存のキーペアを選択するか、新しいキーペアを作成します。 ×

キーペアは、AWS が保存するパブリックキーとユーザーが保存するプライベートキーファイルで構成されます。組み合わせて使用することで、インスタンスに安全に接続できます。Windows AMI の場合、プライベートキーファイルは、インスタンスへのログインに使用されるパスワードを取得するために必要です。Linux AMI の場合、プライベートキーファイルを使用してインスタンスに SSH で安全に接続できます。

注: 選択したキーペアは、このインスタンスに対して権限がある一連のキーに追加されます。「パブリック AMI から既存のキーペアを削除する」の詳細情報をご覧ください。

② **キーペアなしで続行** ①

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。 ③

キャンセル **インスタンスの作成**

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

作成ステータス

1 インスタンスは現在作成中です
この EC2 インスタンスの作成が開始されました。 [作成ログを表示](#)

2 準備が完了した後に受け取る
請求アラートは、Amazon 請求書の予定額を超過した金額を通知するために、無料枠を超えた場合に、メール通知を受け取ります。

EC2 インスタンスへの接続方法

EC2 インスタンスは仮想マシンで、実行中の状態にあり、通常はインターネットに接続されることがあります。新しい EC2 インスタンスの接続情報は、事前に指定した EC2 インスタンスプロファイルに格納されています。
[EC2 インスタンスの接続] をクリックして、EC2 インスタンスの状態を監視します。EC2 インスタンスが一度実行中の状態になると、EC2 インスタンスの接続が開始されます。EC2 インスタンスへの接続方法の詳細はこちら。

ここには、作業を始めるのに役立つリソースがあります

- [最新 EC2 インスタンスの構築方法](#)
- [Amazon EC2 ユーザーガイド](#)
- [AWS 無料枠の制限](#)
- [Amazon EC2 チュートリアルプログラム](#)

EC2 インスタンスの作成中、次のことを行うことができます

- [EC2 インスタンスの作成中に EC2 インスタンスがスタンバイ状態になる場合のトラブルシューティング](#) (接続情報の変更が行われる場合があります)
- [EC2 インスタンスの作成中にアラートを受信する](#) (接続情報の変更が行われる場合があります)
- [EC2 インスタンスの管理](#)

1

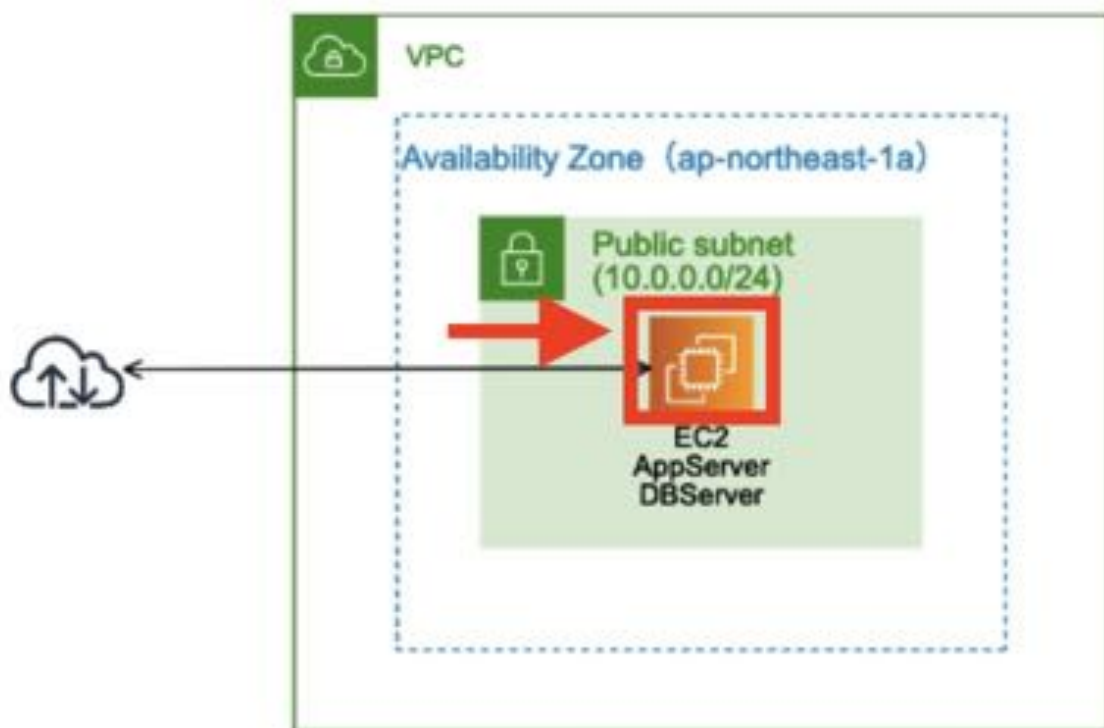
[インスタンスの表示](#)

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



ユーザー名等で絞り込むと便利です。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

Elastic IP アドレスが割り当てられました。
Elastic IP アドレス 18.176.78.173

この Elastic IP アドレスを関連付ける

1

EC2 > Elastic IP アドレス > 18.176.78.173

Elastic IP アドレス (1) リフレッシュ アクション Elastic IP アドレスの割り当て

Elastic IP アドレスは、ユーザーが AWS アカウントに割り当てる静的なパブリック IPv4 アドレスで、インターネットからアクセスできます。 [詳細はこちら](#)

検索

パブリック IPv4 アドレス: 18.176.78.173 × フィルターをクリアする

<input checked="" type="checkbox"/>	Name	パブリック IPv4 アドレス	割り当て ID	関連付けられたインスタンス	プロテクト
<input checked="" type="checkbox"/>		18.176.78.173	eipalloc-0e910a976edc5e106	-	-

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。

EC2 > Elastic IP アドレス > Elastic IP アドレスの関連付け

Elastic IP アドレスの関連付け

この Elastic IP アドレスに関連付けるインスタンスまたはネットワークインターフェイスを選択します (18.178.232.102)

Elastic IP アドレス: 18.178.232.102

リソースタイプ
Elastic IP アドレスに関連付けるリソースのタイプを選択します。

インスタンス
 ネットワークインターフェイス

警告 すでに Elastic IP アドレスが関連付けられているインスタンスに Elastic IP アドレスを関連付けると、前に関連付けられていた Elastic IP アドレスの関連付けが解除されますが、アカウントへの割り当ては維持されます。[詳細はこちら](#)。

インスタンス

Q user1 X [refresh]

i-0e57c47d806af0edd (webserver#1-user1) - running

Elastic IP アドレスに関連付けるプライベート IP アドレスです。

Q プライベート IP アドレスを選択します

再関連付け
Elastic IP アドレスがすでにリソースに関連付けられている場合に、そのアドレスを別のリソースに再度関連付けることができるかどうかを指定します。

Elastic IP アドレスの再関連付けを許可する

キャンセル **関連付ける**

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。

EC2 > Elastic IP アドレス > 18.176.78.173

Elastic IP アドレス (2) アクション Elastic IP アドレスの割り当て

Elastic IP アドレスは、ユーザーが AWS アカウントに割り当てる静的なパブリック IPv4 アドレスで、インターネットからアクセスできます。 [詳細はこちら](#)

Name	パブリック IPv4 アドレス	割り当て ID	関連付けられたインスタンス	アソシエーション ID
	18.176.78.173	eipalloc-0e910a976edc5e106	i-08fee6b77f382a622	07f112f29305d37e93

18.176.78.173

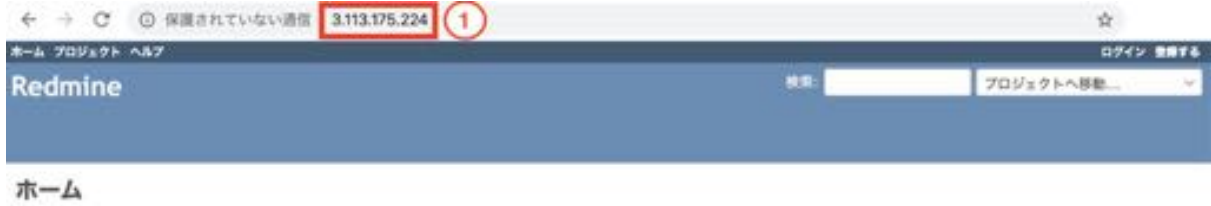
概要

パブリック IP アドレス 18.176.78.173	割り当て ID eipalloc-0e910a976edc5e106	アソシエーション ID eipassoc-07f112f29305d37e93	スコープ VPC
関連付けられたインスタンス i-08fee6b77f382a622	プライベート IP アドレス 10.0.0.249	ネットワークインターフェイス ID eni-009f478e981916aa7	ネットワークインターフェイス所有者のアカウント ID 533384410763
パブリック DNS ec2-18-176-78-173.ap-northeast-1.compute.amazonaws.com	NAT Gateway ID -	アドレスプール Amazon	

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

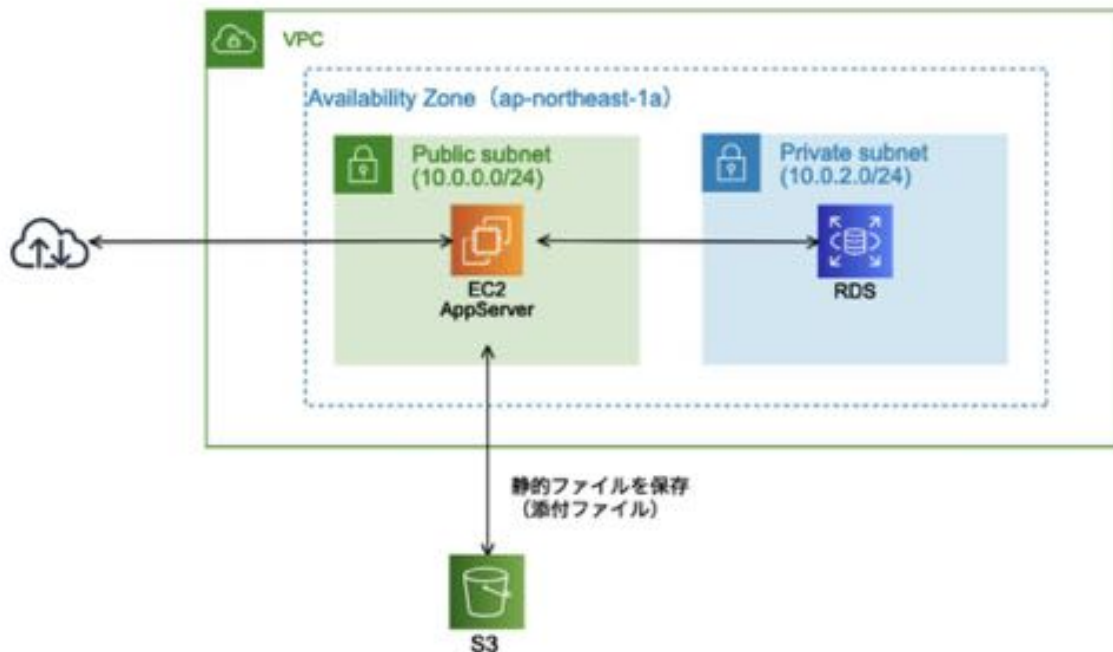


先ほどメモした EIP にアクセスし、redmineが表示されることを確認します。

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。

セキュリティグループの作成

セキュリティグループ名 ① db-user1
 説明 ① RDS for MySQL
 VPC ① vpc-c96a25ad | handson-user1

セキュリティグループのルール

インバウンド アウトバウンド

タイプ ①	プロトコル ①	ポート範囲 ①	送信元 ①
MySQL/Aurora	TCP	3306	カスタム web

ルールの追加 ①

sg-ad73b6ca - web-user1 ④

キャンセル 作成 ⑤

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。

RDS > サブネットグループ > DB サブネットグループの作成

DB サブネットグループの作成

新しいサブネットグループを作成するには、名前と説明を入力し、既存の VPC を選択します。次に、その VPC に関連するサブネットを追加できます。

サブネットグループの詳細

名前

サブネットグループの作成後に名前を変更することはできません。

1

1～255 文字を含める必要があります。英数字、スペース、ハイフン、アンダースコア、ピリオドを使用できます。

説明

2

VPC

DB サブネットグループに使用するサブネットに対応する VPC 識別子を選択します。サブネットグループの作成後に別の VPC 識別子を選択することはできません。

3

サブネットの追加

サブネットをこのサブネットグループに追加します。サブネットを1つずつ追加することも、この VPC に関連するすべてのサブネットを追加することもできます。このグループの作成後、追加/編集ができます。最低で2つのサブネットが必要です。

アベイラビリティゾーン

4

サブネット

5

6

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。

サブネットの追加

サブネットをこのサブネットグループに追加します。サブネットを1つずつ追加することも、この VPC に関連するすべてのサブネットを追加することもできます。このグループの作成後、追加/編集ができます。最低で2つのサブネットが必要です。

この VPC に関連するすべてのサブネットを追加します

アベイラビリティゾーン

ap-northeast-1c 1

サブネット

subnet-00555008f36eb4c81 (10.0.3.0/24) 2

サブネットを追加します 3

このサブネットグループのサブネット (1)

アベイラビリティゾーン	サブネット ID	CIDR ブロック	アクション
ap-northeast-1a	subnet-068316666bc7240af	10.0.2.0/24	削除

キャンセル 作成

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。

サブネットの追加

サブネットをこのサブネットグループに追加します。サブネットを1つずつ追加することも、このVPCに関連するすべてのサブネットを追加することもできます。このグループの作成後、追加/編集ができます。最低で2つのサブネットが必要です。

このVPCに関連するすべてのサブネットを追加します

アベイラビリティゾーン
ap-northeast-1c

サブネット
subnet-00555008f36eb4c81 (10.0.3.0/24)

サブネットを追加します

このサブネットグループのサブネット (2)

アベイラビリティゾーン	サブネット ID	CIDR ブロック	アクション
ap-northeast-1c	subnet-00555008f36eb4c81	10.0.3.0/24	削除
ap-northeast-1a	subnet-068316666bc7240af	10.0.2.0/24	削除

1

キャンセル **作成** 2

- 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。
- 「作成」をクリックします。

RDS > サブネットグループ

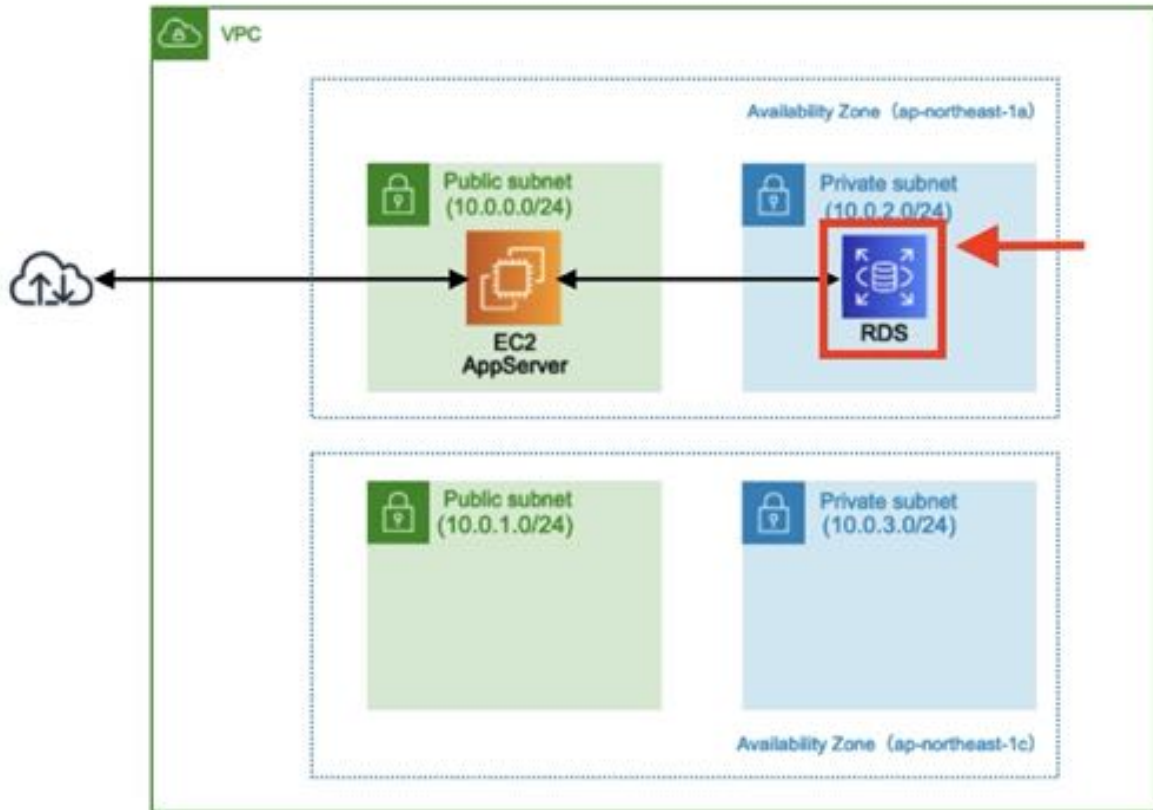
サブネットグループ (2) リフレッシュ 編集 削除 DB サブネットグループの作成

フィルタ サブネットグループ < 1 >

名前	説明	ステータス
db subnet user1	db subnet user1	完了

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

The screenshot shows the AWS Management Console interface for Amazon RDS. On the left, the navigation menu includes 'ダッシュボード' (Dashboard), 'データベース' (Databases), 'Query Editor', 'パフォーマンスインサイト' (Performance Insights), 'スナップショット' (Snapshots), 'Automated backups', 'リザーブドインスタンス' (Reserved Instances), 'Proxies', 'サブネットグループ' (Subnet Groups), 'パラメータグループ' (Parameter Groups), 'オプショングループ' (Option Groups), 'イベント' (Events), 'イベントサブスクリプション' (Event Subscriptions), 'Recommendations', and 'Certificate update'. The 'ダッシュボード' link is highlighted with a red box and a circled '1'. The main content area is titled 'リソース' (Resources) and shows a list of resources for the Asia Pacific (Tokyo) region. The 'データベースの作成' (Create Database) link is highlighted with a red box and a circled '2'.

リソース 更新

Asia Pacific (Tokyo) リージョンで、以下の Amazon RDS リソースを使用します (使用した量/クォータ)

DB インスタンス (1/40)	パラメータグループ (9)
ストレージ割り当て (0 バイト/100.00 TB)	デフォルト (6)
	カスタム (3/100)
DB インスタンス上限を引き上げるには、こちらをクリックしてください	オプショングループ (5)
	デフォルト (5)
	カスタム (0/20)
リザーブドインスタンス (0/40)	サブネットグループ (2/50)
スナップショット (60)	サポートされているプラットフォーム VPC
手動 (7/100)	デフォルトネットワーク vpc-775b0510
自動 (0)	
最近のイベント (5)	
イベントサブスクリプション (0/20)	

データベースの作成

Amazon Relational Database Service (RDS) を使用すると、クラウド上でレシヨナルデータベースを簡単にセットアップ、運用、スケーリングできます。

S3 から復元 データベースの作成 2

注: DB インスタンスは以下で作成されず Asia Pacific (Tokyo) リージョン

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。

データベースの作成

データベース作成方法を選択 情報

標準作成

可用性、セキュリティ、バックアップ、メンテナンスといったすべての設定オプションを設定します。

簡単作成

推奨されるベストプラクティス設定を使用します。一部の設定オプションは、データベースの作成後に変更できます。

エンジンのオプション

エンジンのタイプ 情報

Amazon Aurora



MySQL



MariaDB



PostgreSQL



Oracle

ORACLE

Microsoft SQL Server



エディション

MySQL Community

バージョン 情報

MySQL 5.7.22

[エンジンのオプション]

1. 「MySQL」を選択します。

テンプレート

お客様のユースケースに合わせてサンプルテンプレートを選択します。

1

<p><input type="radio"/> 本番稼働用 高い可用性と、高速で安定したパフォーマンスのためには、デフォルト値を使用します。</p>	<p><input checked="" type="radio"/> 開発/テスト このインスタンスは本番稼働環境ではない開発で使用します。</p>	<p><input type="radio"/> 無料利用枠 RDS 無料利用枠を利用すると、新しいアプリケーションの開発、既存のアプリケーションのテスト、Amazon RDS の実践経験の蓄積が可能です。 情報</p>
--	--	---

設定

DB インスタンス識別子 情報

DB インスタンスの名前を入力します。この名前は、AWS アカウントが現在の AWS リージョンで所有しているすべての DB インスタンスにおいて一意である必要があります。

redmine-user1

2

DB インスタンス識別子は、大文字と小文字を区別しませんが、すべて小文字で保存されます (例: 'mydbinstance')。制約として、使用できるのは 1~60 文字以内で英数字またはハイフンのみです (SQL Server は 1~15 文字)。1 文字目は英文字でなければなりません。また、ハイフンを連続で 2 つ使ったり、最後の文字をハイフンにしたりすることはできません。

▼ 認証情報の設定

マスターユーザー名 情報

DB インスタンスのマスターユーザーのログイン ID を入力します。

admin

3

1~16 文字の英数字。1 文字目は文字である必要があります

パスワードの自動生成

Amazon RDS がパスワードを生成するか、お客様がご自身でパスワードを指定することができます

マスターパスワード 情報

4

制約事項: 表示可能な ASCII 文字で 8 文字以上で入力してください。次の文字を含めることはできません: / (スラッシュ)、* (二重引用符)、および @ (アットマーク)。

パスワードを確認 情報

5

[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

DB インスタンスクラス 情報
処理能力とメモリの要件に合った DB インスタンスクラスを選択します。以下の DB インスタンスクラスオプションは、上記で選択したエンジンでサポートされているものに制限されます。

標準クラス (m クラスを含む)

メモリ最適化クラス (r クラスと x クラスを含む)

バースト可能クラス (t クラスを含む) 1

db.t2.micro 2
1 vCPUs 1 GiB RAM Not EBS Optimized

以前の世代のクラスを含める

ストレージ

ストレージタイプ 情報
汎用 (SSD)

ストレージ割り当て
20 GiB

(最小: 20 GiB、最大: 16384 GiB) より高い割り当て済みストレージは、IOPS のパフォーマンスを改善する場合があります。

可用性と耐久性

マルチ AZ 配置 情報

スタンバイインスタンスを作成する (本稼働環境向けに推奨)
データの冗長性を提供し、I/O のフリーズを防ぎ、システムバックアップの間のレイテンシーの急上昇を最小限に抑えるために、別のアベイラビリティゾーン (AZ) にスタンバイを作成します。

スタンバイインスタンスを作成しないでください 3

[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 🔄

Virtual Private Cloud (VPC) 情報
この DB インスタンスの仮想ネットワーク環境を定義する VPC。

handson-user1 (vpc-0219c5e2bc2073785) ▼ 1

対応する DB サブネットグループがある VPC のみが表示されます。

📌 データベースの作成後に、VPC の選択を変更することはできません。

▼ 追加の接続設定 2

サブネットグループ 情報
選択した VPC で DB インスタンスが使用できるサブネットと IP 範囲を定義する DB サブネットグループ。

db subnet user1 ▼ 3

パブリックアクセス可能 情報

あり
VPC 外部の Amazon EC2 インスタンスとデバイスがお客様のデータベースに接続できます。データベースに接続できる VPC 内の EC2 インスタンスおよびデバイスを指定する 1 つ以上の VPC セキュリティグループを選択します。

なし 4
パブリック IP アドレスをデータベースに割り当てません。VPC 内部の Amazon EC2 インスタンスとデバイスのみをお客様のデータベースに接続できます。

VPC セキュリティグループ
RDS セキュリティグループを 1 つ以上選択し、データベースへのアクセスを許可します。セキュリティグループのルールで EC2 インスタンスと VPC 外のデバイスからの着信トラフィックが許可されていることを確認します (セキュリティグループはパブリックにアクセス可能なデータベースに必要です)。

既存の選択
既存の VPC セキュリティグループの選択

新規作成
新しい VPC セキュリティグループを作成

既存の VPC セキュリティグループ
VPC セキュリティグループを選択します

default 5 X

6

既存の VPC セキュリティグループ

VPC セキュリティグループを選択します

db-user1

アベイラビリティゾーン 情報

ap-northeast-1a ▼ 7

データベースポート 情報
データベースがアプリケーションの接続に使用する TCP/IP ポート。

3306

[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、「×」を外します。
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

データベースオプション、バックアップが有効、バックトラックが無効、拡張モニタリングが有効、メンテナンス、CloudWatch Logs、削除保護が無効

データベースの選択肢

最初のデータベース名 [情報](#)

データベース名を指定しないと、Amazon RDS はデータベースを作成しません。

DB パラメータグループ [情報](#)

default.mysql5.7
▼

オプショングループ [情報](#)

default:mysql-5-7
▼

バックアップ

データベースのポイントインタイムスナップショットを作成します

自動バックアップの有効化
 バックアップを有効にすると、特定の時間帯でデータベースのバックアップが自動的に作成されます。

⚠️ 自動バックアップは現在 InnoDB ストレージエンジンでのみサポートされていることに注意してください。MyISAM を使用している場合、詳細についてはこちらを参照してください。

バックアップ保持期間 [情報](#)

このインスタンスの自動バックアップを RDS が保存する日数を選択します。

0 日間
▼
2

バックアップウィンドウ [情報](#)

Amazon RDS によって作成されるデータベースの自動バックアップの期間を選択します。

キャンセル
データベースの作成
3

[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。

The screenshot shows the Amazon RDS console for an instance named 'redmine-user1'. The left sidebar contains navigation options like 'ダッシュボード', 'データベース', 'Query Editor', etc. The main content area shows the instance details under the '接続とセキュリティ' (Connections and Security) tab. A red box highlights the 'エンドポイントとポート' (Endpoint and Port) section, which contains the following information:

エンドポイント
redmine-user1.cizpucnfhf8.ap-northeast-1.rds.amazonaws.com

Below the endpoint, the port is listed as 3306. Other sections visible include 'ネットワーク' (Network) and 'セキュリティ' (Security).

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redmine1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

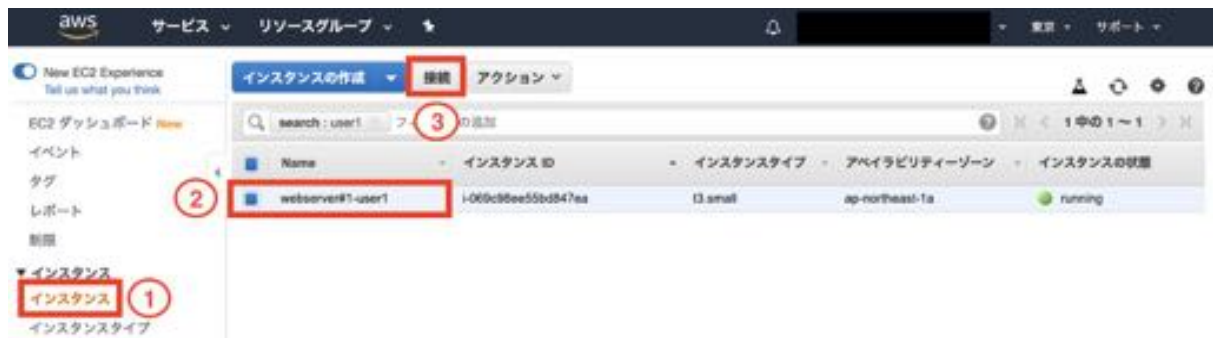
ステップ 2-5-1: Redmineにダミーデータを登録

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 ×

接続方法

- スタンドアロン SSH クライアント ⓘ
- セッションマネージャー ⓘ 1
- EC2 Instance Connect (ブラウザベースの SSH 接続) ⓘ

セッションマネージャーの使用

- SSH キーまたは踏み台ホストなしでインスタンスに接続します。
- セッションは、AWS Key Management Service キーを使用して保護されます。
- セッションコマンドと詳細を Amazon S3 バケットまたは CloudWatch Logs ロググループに記録できます。
- セッションマネージャー [設定](#) ページでセッションを設定します。

詳細については、[セッションマネージャーの開始方法](#) を参照してください。

2
接続

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。

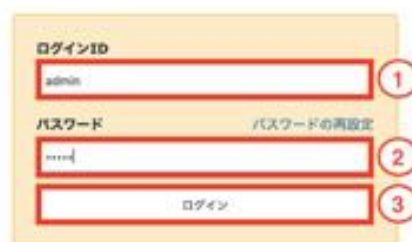
```
RAILS_ENV=production bundle exec rake db:fixtures:load
```

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。



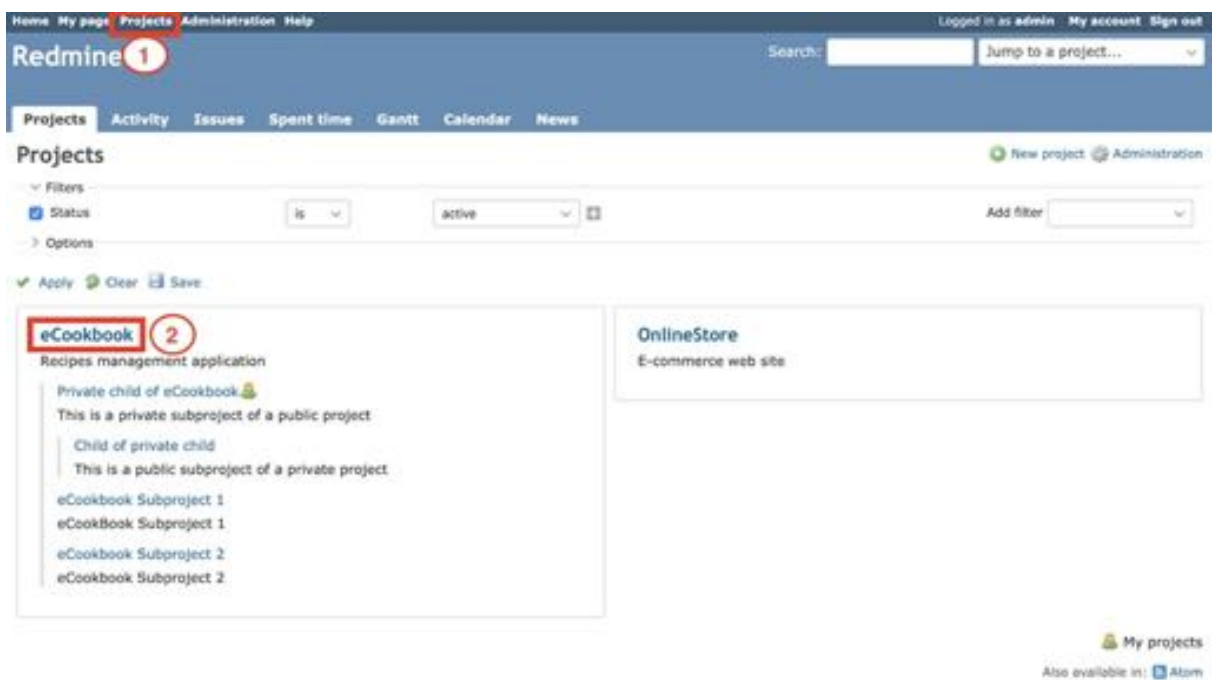
先ほど fixtures:load を実行しダミーのユーザが作成されているため、admin でログインします。

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。

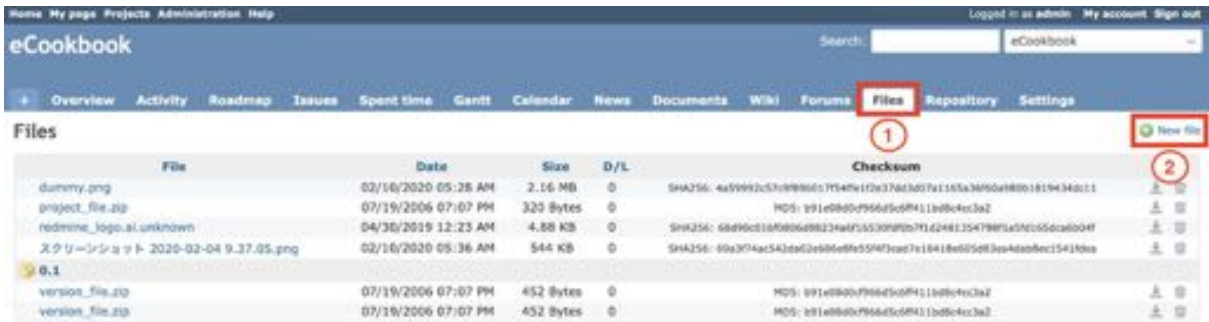


Homeが表示されたらログイン成功です。



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。

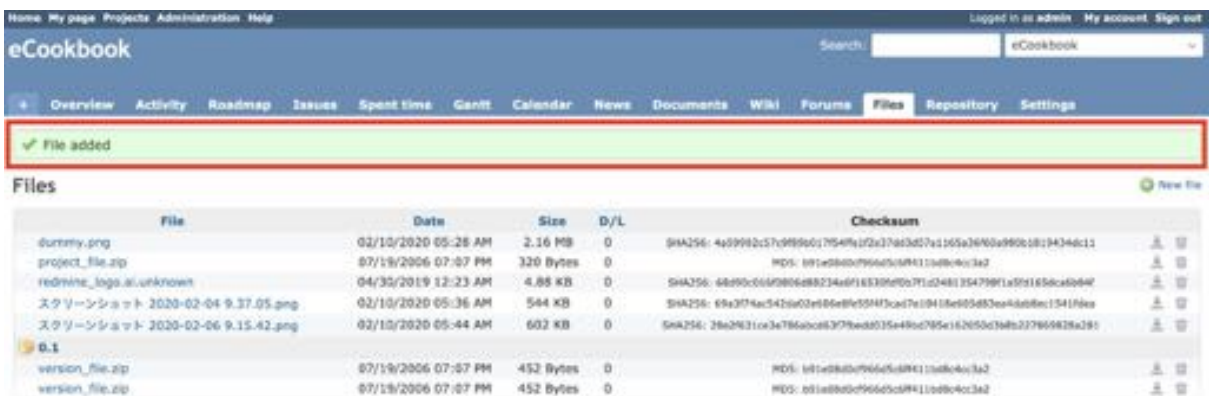
1. 「projects」をクリックします。
2. 「eCookbook」をクリックします。



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。

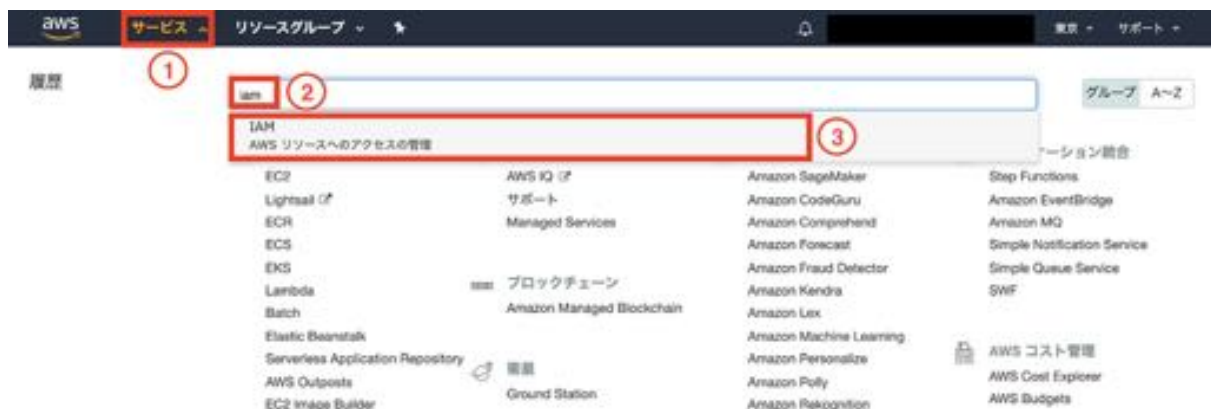
ステップ 2-5-3: ファイルのアップロード確認

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。

ユーザーを追加



ユーザー詳細の設定

同じアクセスの種類とアクセス権限を使用して複数のユーザーを一度に追加できます。 [詳細はこちら](#)

ユーザー名* 1

[別のユーザーの追加](#)

AWS アクセスの種類を選択

これらのユーザーから AWS にアクセスする方法を選択します。アクセスキーと自動生成パスワードは前のステップで提供されています。 [詳細はこちら](#)

- アクセスの種類*
- プログラムによるアクセス**
AWS API、CLI、SDK などの開発ツールの **アクセスキー ID** と **シークレットアクセスキー** を有効にします。 2
 - AWS マネジメントコンソールへのアクセス**
ユーザーに AWS マネジメントコンソールへのサインインを許可するための **パスワード** を有効にします。

3

キャンセル

次のステップ: アクセス権限

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。

ユーザーを追加

1 2 3 4 5

▼ アクセス許可の設定

1

ユーザーをグループに追加 アクセス権限を既存のユーザーからコピー 既存のポリシーを直接アタッチ

ポリシーの作成

ポリシーのフィルタ Q AmazonS3FullAccess 2 1件の結果を表示中

ポリシー名	タイプ	次として使用
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS による管理	Permissions policy (9)

3

4

キャンセル 戻る 次のステップ: タグ

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。

ユーザーを追加

1 2 3 4 5

タグの追加 (オプション)

IAM タグは、ユーザー に追加できるキーと値のペアです。タグには、E メールアドレスなどのユーザー情報を含めるか、役職などの説明文とすることができます。タグを使用して、このユーザー のアクセスを整理、追跡、制御できます。 [詳細はこちら](#)

キー	値 (オプション)	削除
<input type="text" value="Name"/> 1	<input type="text" value="iam-user1"/> 2	✕
<input type="text" value="新しいキーを追加"/>	<input type="text"/>	

さらに 49 個のタグを追加できます。

3

キャンセル 戻る 次のステップ: 確認

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。

ユーザーを追加

1 2 3 4 5

確認

選択内容を確認します。ユーザーを作成した後で、自動生成パスワードとアクセスキーを確認してダウンロードできます。

ユーザー詳細

ユーザー名	s3access-20200228
AWS アクセスの種類	プログラムによるアクセス - アクセスキーを使用
アクセス権限の境界	アクセス権限の境界が設定されていません

アクセス権限の概要

次のポリシー例は、上記のユーザーにアタッチされます。

タイプ	名前
管理ポリシー	AmazonS3FullAccess

タグ

新しいユーザーは次のタグを受け取ります

キー	値
Name	iam-user1

キャンセル

戻る

1 ユーザーの作成

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。

ユーザーを追加

1 2 3 4 5

成功

以下に示すユーザーを正常に作成しました。ユーザーのセキュリティ認証情報を確認してダウンロードできます。AWS マネジメントコンソールへのサインイン手順を E メールでユーザーに送信することもできます。今回が、これらの認証情報をダウンロードできる最後の機会です。ただし、新しい認証情報はいつでも作成できます。

AWS マネジメントコンソールへのアクセス権を持つユーザーは「<https://533384410763.signin.aws.amazon.com/console>」でサインインできます

1 .csv のダウンロード

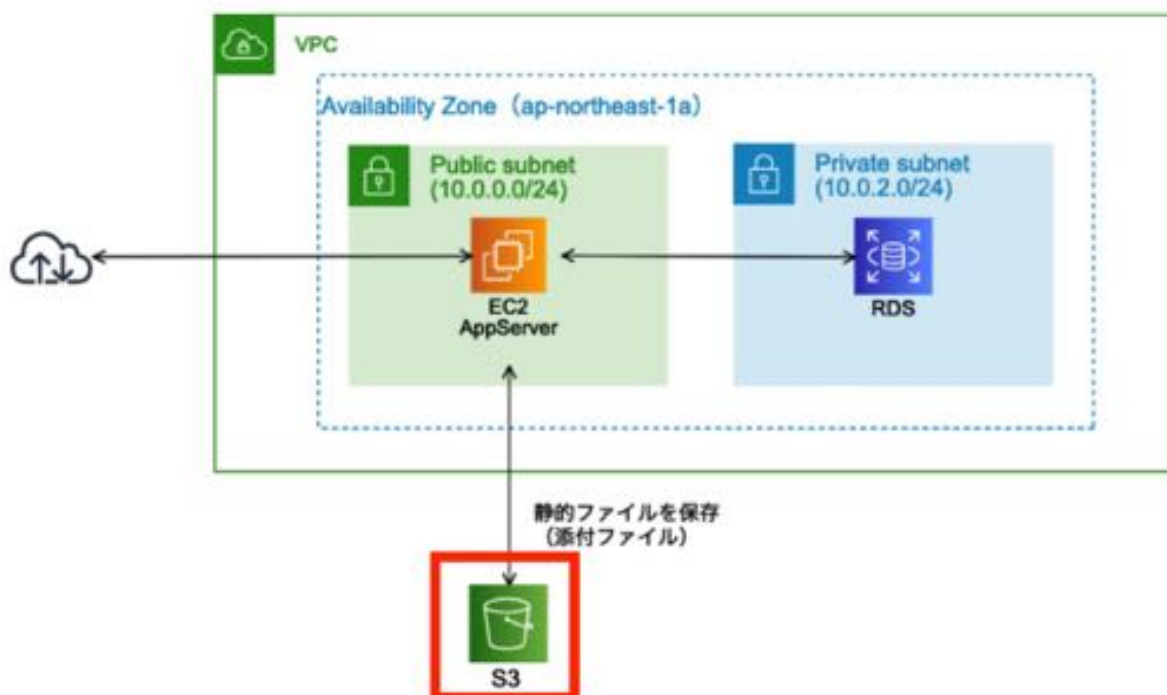
ユーザー	アクセスキー ID	シークレットアクセスキー
▶ s3access-20200228	AKIAXYMBQG2FV5OYN2XD	***** 表示

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 1

Projects Activity Issues Spent time Gantt Calendar News

Projects New project Administration

Filters

Status is active Add filter

Options

Apply Clear Save

eCookbook 2

Recipes management application

Private child of eCookbook

This is a private subproject of a public project

Child of private child

This is a public subproject of a private project

eCookbook Subproject 1

eCookbook Subproject 1

eCookbook Subproject 2

eCookbook Subproject 2

OnlineStore

E-commerce web site

My projects

Also available in: Atom

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。

eCookbook

Overview Activity Roadmap Issues Spent time Gantt Calendar News Documents Wiki Forums **Files** Repository Settings

Files New file

File	Date	Size	D/L	Checksum
dummy.png	02/16/2020 05:28 AM	2.16 MB	0	SHA256: 4a59992c57b99866577540e172b376c3037e1185a3650a18801819436d11
project_file.zip	07/19/2006 07:07 PM	320 Bytes	0	MD5: b91e0800f966d5c0f411b0b4cc3a2
redmine_logo.ai.unknown	04/30/2019 12:23 AM	4.88 KB	0	SHA256: 68d9c818f086d88234a0f5530905b71d2481254798f1a5f1c050ca6004f
スクリーンショット 2020-02-04 9:37:05.png	02/16/2020 05:36 AM	544 KB	0	SHA256: 00a3074ac543d032e99e6e9f559f3ead7e18418e05e83ee4a0b8ec1541fde
0.1				
version_file.zip	07/19/2006 07:07 PM	452 Bytes	0	MD5: b91e0800f966d5c0f411b0b4cc3a2
version_file.zip	07/19/2006 07:07 PM	452 Bytes	0	MD5: b91e0800f966d5c0f411b0b4cc3a2

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

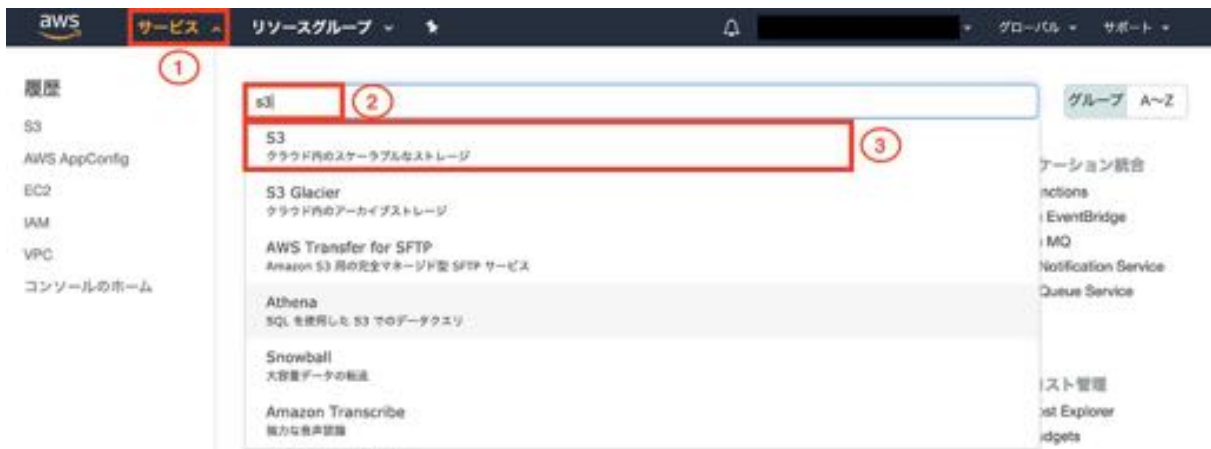


3. 「ファイルを選択」をクリックして、アップロードするファイルを選択します。
4. 「Add」をクリックします。

ステップ 2-5-9: S3アップロード確認

redmine1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。

The screenshot shows the Amazon S3 console interface. At the top, there is a dark blue banner with white text: "S3 オブジェクトロックを使用すれば、事前に定義した保持期間内に S3 オブジェクトが削除されるのを防ぐことができます。詳細はこちら" and a "ドキュメント" link. Below this, the "S3 バケット" section is visible. A search bar contains the text "user1" and is marked with a red circle and the number "1". To the right of the search bar is a dropdown menu for "すべてのアクセスタイプ". Below the search bar are several buttons: "+ バケットを作成する", "パブリックアクセス設定を編集する", "空にする", and "削除". To the right of these buttons, it says "1 リージョン" and "1 バケット". Below this is a table with columns: "バケット名", "アクセス", "リージョン", and "作成日". The first row in the table is highlighted with a red box and marked with a red circle and the number "2". The row contains: a checkbox, "redmine-user1-20200228", "バケットとオブジェクトは非公開", "アジアパシフィック (東京)", and "2月 10, 2020 11:29:54 午前 GMT+0900".

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。

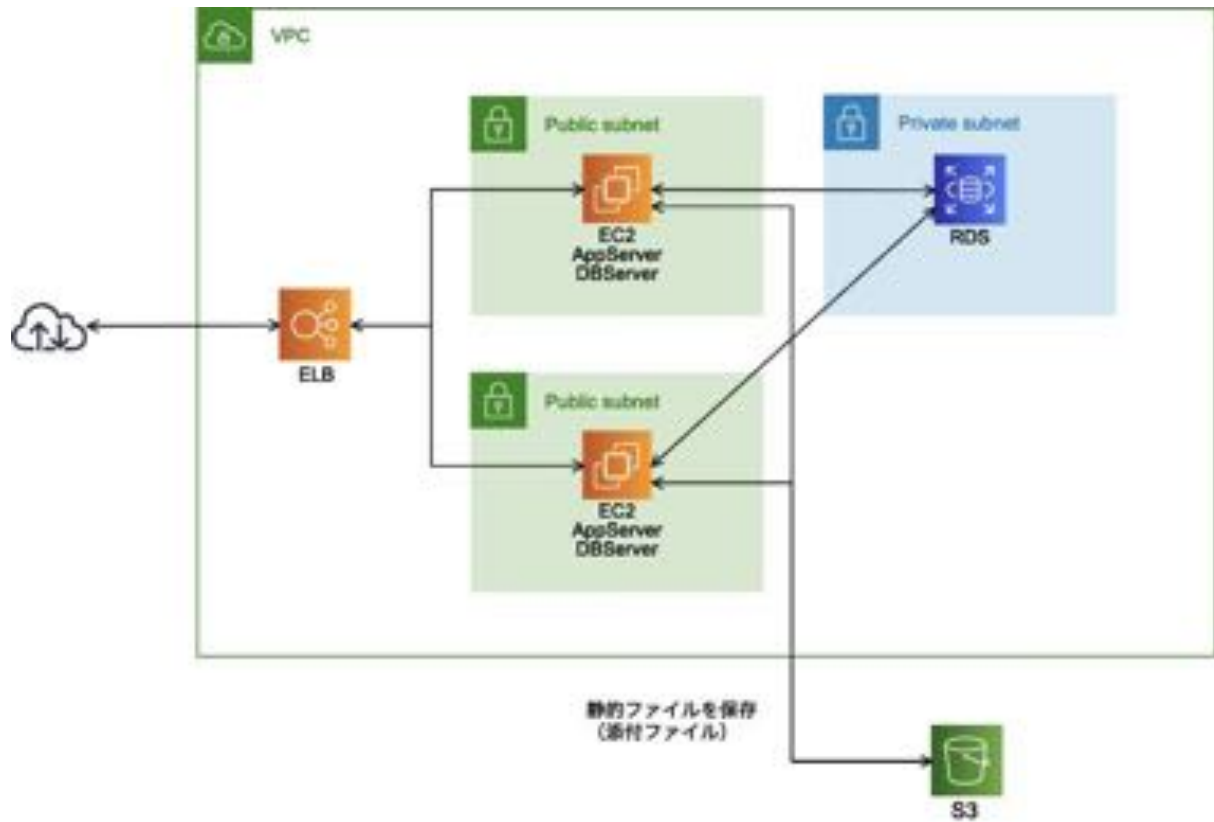
1. 自分の名前などで検索します。(日付でも○)
2. フェーズ2-5-5で作成したバケット名をクリックします。

The screenshot shows the Amazon S3 console interface for a specific bucket. The breadcrumb navigation at the top reads: "Amazon S3 > redmine-user1-20200228 > username > files > 2020 > 02". Below this, the bucket name "redmine-user1-20200228" is displayed. There is a "概要" tab. Below the tab is a search bar with the text: "プレフィックスを入力し、Enter キーで検索します。ESC を押してクリアします。". Below the search bar are several buttons: "アップロード", "+ フォルダの作成", "ダウンロード", and "アクション". To the right of these buttons, it says "アジアパシフィック (東京)" and "表示中 1 ~ 3". Below this is a table with columns: "名前", "最終更新日時", "サイズ", and "ストレージクラス". The first row in the table is highlighted with a red box. The row contains: a checkbox, "200210052815_dummy.png", "2月 10, 2020 2:28:16 午後 GMT+0900", "2.2 MB", and "スタンダード".

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
-

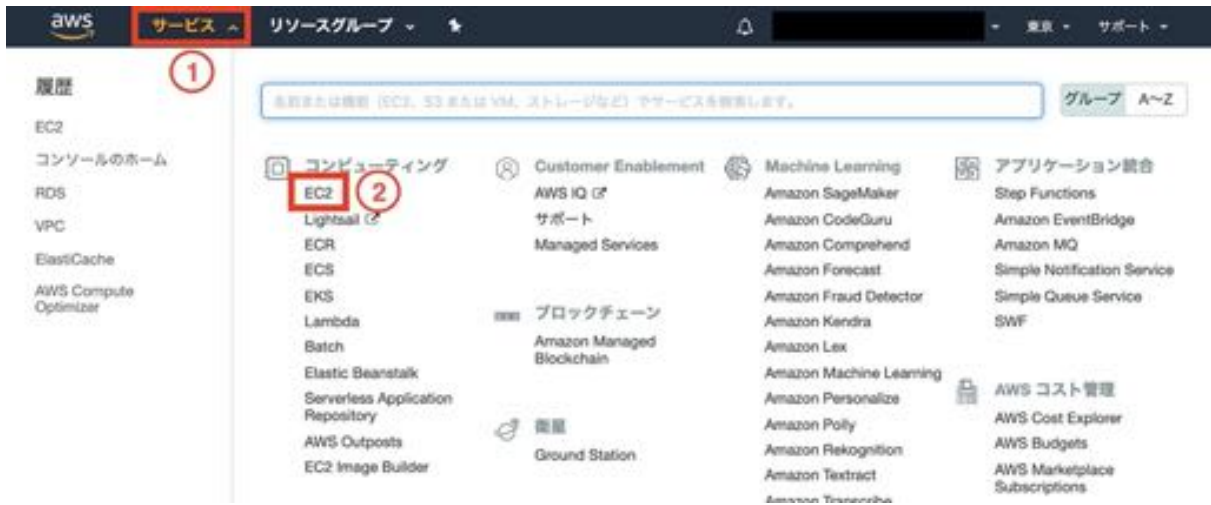
[フェーズ3]

～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. 「webservice-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。

イメージの作成 ×

インスタンス ID ⓘ i-Oe57c47d906af0edd

イメージ名 ⓘ 1

イメージの説明 ⓘ

再起動しない ⓘ

インスタンスボリューム

ボリュームタイプ ⓘ	デバイス ⓘ	スナップショット ⓘ	サイズ (GiB) ⓘ	ボリュームタイプ ⓘ	IOPS ⓘ	スループット (MB/秒) ⓘ	終了時に削除 ⓘ	暗号化済み ⓘ
ルート	/dev/sda1	snap-0a1027278e666fa9	10	汎用 SSD (gp2)	100 / 3000	該当なし	<input checked="" type="checkbox"/>	暗号化なし

新しいボリュームの追加

EBS ボリュームの合計サイズ: 10 GiB
EBS イメージを作成すると、上の各ボリュームの EBS スナップショットも作成されます。

キャンセル
イメージの作成 2

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。

イメージの作成 ×

✔
イメージの作成リクエストを受け取りました
1

保留中のイメージ ami-091204285810b0d7 の表示

イメージの作成が正常に完了すると、新しい EBS イメージをバックアップするスナップショットを、[スナップショット画面](#) で管理できるようになります。

閉じる

「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。



The screenshot shows the AWS Management Console interface for AMIs. The search bar contains 'ami-09120f42858f0b0d7'. The table below shows one AMI with the status 'available' highlighted in a red box.

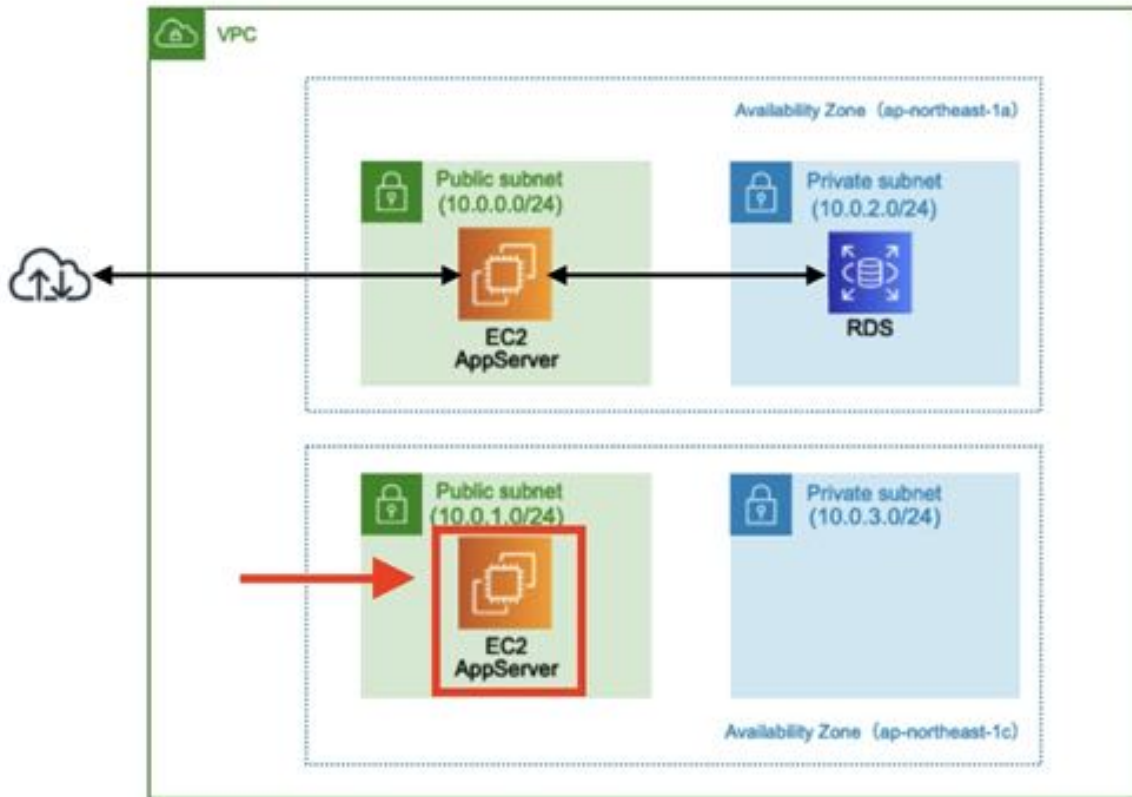
Name	AMI 名	AMI ID	ソース	所有者	可視性	ステータス	作成日
wordpress user1	ami-09...	ami-09120f42858f0b0d7	533384410763/...	533384410763	プライベート	available	2020年1月28日 9

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。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 2: インスタンスタイプの選択

<input type="checkbox"/>	汎用	t3.nano	2	0.5	EBS のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	汎用	t3.micro	2	1	EBS のみ	はい	最大 5 ギガビット	はい
<input checked="" type="checkbox"/>	汎用	t3.small	2	2	EBS のみ	はい	最大 5 ギガビット	はい
<input type="checkbox"/>	汎用	t3.medium	2	4	EBS のみ	はい	最大 5 ギガビット	はい

キャンセル 戻る **確認と作成** 次のステップ: インスタンスの詳細の設定

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。

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 3: インスタンスの詳細の設定

要件に合わせてインスタンスを設定します。同じ AMI からの複数インスタンス作成や、より低料金を実現するためのスポットインスタンスのリクエスト、インスタンスへのアクセス管理ロール割り当てなどを行うことができます。

インスタンス数 ① 1 Auto Scaling グループに作成する ②

購入のオプション ① スポットインスタンスのリクエスト

ネットワーク ① vpc-0e941f74723ca26f2 | handson-user1 ① ③ ④ ⑤
C 新しい VPC の作成

サブネット ① subnet-0d9815dd778e81356 | パブリックサブネット ②
249 個の IP アドレスが 100 個可能
C 新しいサブネットの作成

自動割り当てパブリック IP ① 有効 ③

配置グループ ① インスタンスをプレイズメントグループに追加します。

キャパシティの予約 ① 閉く ④
C 新しいキャパシティ予約の作成

IAM ロール ① session-manager-20200228 ④
C 新しい IAM ロールの作成

キャンセル 戻る **確認と作成** 次のステップ: ストレージの追加

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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 4: ストレージの追加

インスタンスは次のストレージデバイス設定を使用して作成されます。インスタンスに追加の EBS ボリュームやインスタンスストアボリュームをアタッチするか、ルートボリュームの設定を編集することができます。また、インスタンスを作成してから追加の EBS ボリュームをアタッチすることもできますが、インスタンスストアボリュームはアタッチできません。Amazon EC2 のストレージオプションに関する [詳細](#) はこちらをご覧ください。

ボリュームタイプ	デバイス	スナップショット	サイズ (GiB)	ボリュームタイプ	IOPS	スループット (MB/秒)	終了時に削除	暗号化
ルート	/dev/xvda	snap-00e37af63af76d77c	8	汎用 SSD (gp2)	100 / 3000	該当なし	<input checked="" type="checkbox"/>	暗号化

新しいボリュームの追加

キャンセル 戻る 確認と作成 **次のステップ: タグの追加**

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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 5: タグの追加

タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、キーに「Name」、値に「Webserver」を使用してタグを定義することができます。タグのコピーは、ボリューム、インスタンス、またはその両方に適用できます。タグは、すべてのインスタンスとボリュームに適用されます。Amazon EC2 リソースのタグ付けに関する [詳細](#) はこちら。

キー (最大 128 文字)	値 (最大 256 文字)	インスタンス	ボリューム
Name	webserver#2-user1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

タグの追加 (最大 50 個のタグ)

キャンセル 戻る 確認と作成 **次のステップ: セキュリティグループの設定**

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。

1. AMIの選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 6: セキュリティグループの設定

セキュリティグループは、インスタンスのトラフィックを制御するファイアウォールのルールセットです。このページで、特定のトラフィックに対してインスタンスへの到達を許可するルールを追加できます。たとえば、ウェブサーバーをセットアップして、インターネットトラフィックにインスタンスへの到達を許可する場合、HTTP および HTTPS ポートに無制限のアクセス権限を与えます。新しいセキュリティグループを作成するか、次の既存のセキュリティグループから選択することができます。Amazon EC2 セキュリティグループに関する詳細はこちら。

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

既存のセキュリティグループを選択する 1

セキュリティグループ ID	名前	説明	アクション
<input type="checkbox"/> sg-0aabd2572a015f888	db-user1	RDS for MySQL	コピーして新規作成
<input type="checkbox"/> sg-024a52f0671ad71	default	default VPC security group	コピーして新規作成
<input checked="" type="checkbox"/> sg-00f3d0e5d3fc12c43	web-user1	web-user1	コピーして新規作成 2

sg-00f3d0e5d3fc12c43 に関するインバウンドのルール (選択したセキュリティグループ: sg-00f3d0e5d3fc12c43)

タイプ (i)	プロトコル (i)	ポート範囲 (i)	ソース (i)	説明 (i)
HTTP	TCP	80	0.0.0.0/0	

3

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 ×

▲ **警告**
AMI では、アクセスを可能にするためにポート 22 を開く必要があるため、このインスタンスに接続できません。現在のセキュリティグループでは、ポート 22 が開いていません。

1

2

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

1. AMI の選択 2. インスタンスタイプの選択 3. インスタンスの設定 4. ストレージの追加 5. タグの追加 6. セキュリティグループの設定 7. 確認

ステップ 7: インスタンス作成の確認

インスタンスの作成に関する詳細を確認してください。各セクションの変更を行うことができます。[作成] をクリックして、インスタンスにキーペアを割り当て、作成処理を完了します。

⚠ インスタンスのセキュリティを強化してください。セキュリティグループ `web-user1` は世界に向けて開かれています。このインスタンスには、どの IP アドレスからもアクセスできる可能性があります。セキュリティグループのルールを更新して、既知の IP アドレスからのみアクセスできるようにすることをお勧めします。また、セキュリティグループの追加ポートを開いて、実行中のアプリケーションやサービスへのアクセスを容易にすることもできます。たとえば、ウェブサーバー用に HTTP (80) を開きます。 [セキュリティグループの編集](#)

AMI の詳細

AMI の編集

wordpress user1 - ami-09120f42858f0b0d7
ルートデバイスタイプ: ebs 仮想化タイプ: hvm

インスタンスタイプ

インスタンスタイプの編集

インスタンスタイプ	ECU	vCPU	メモリ (GiB)	インスタンス ストレージ (GiB)	EBS 最適化利用	ネットワークパフォーマンス
t2.micro	可変	1	1	EBS のみ	-	Low to Moderate

キャンセル 戻る **起動**

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 3-2-2: キーペアを選択する

既存のキーペアを選択するか、新しいキーペアを作成します。 ×

キーペアは、AWS が保存するパブリックキーとユーザーが保存するプライベートキーファイルで構成されます。組み合わせて使用することで、インスタンスに安全に接続できます。Windows AMI の場合、プライベートキーファイルは、インスタンスへのログインに使用されるパスワードを取得するために必要です。Linux AMI の場合、プライベートキーファイルを使用してインスタンスに SSH で安全に接続できます。

注: 選択したキーペアは、このインスタンスに対して権限がある一連のキーに追加されます。「パブリック AMI から既存のキーペアを削除する」の詳細情報をご覧ください。

② ①
 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。 ③

キャンセル **インスタンスの作成**

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



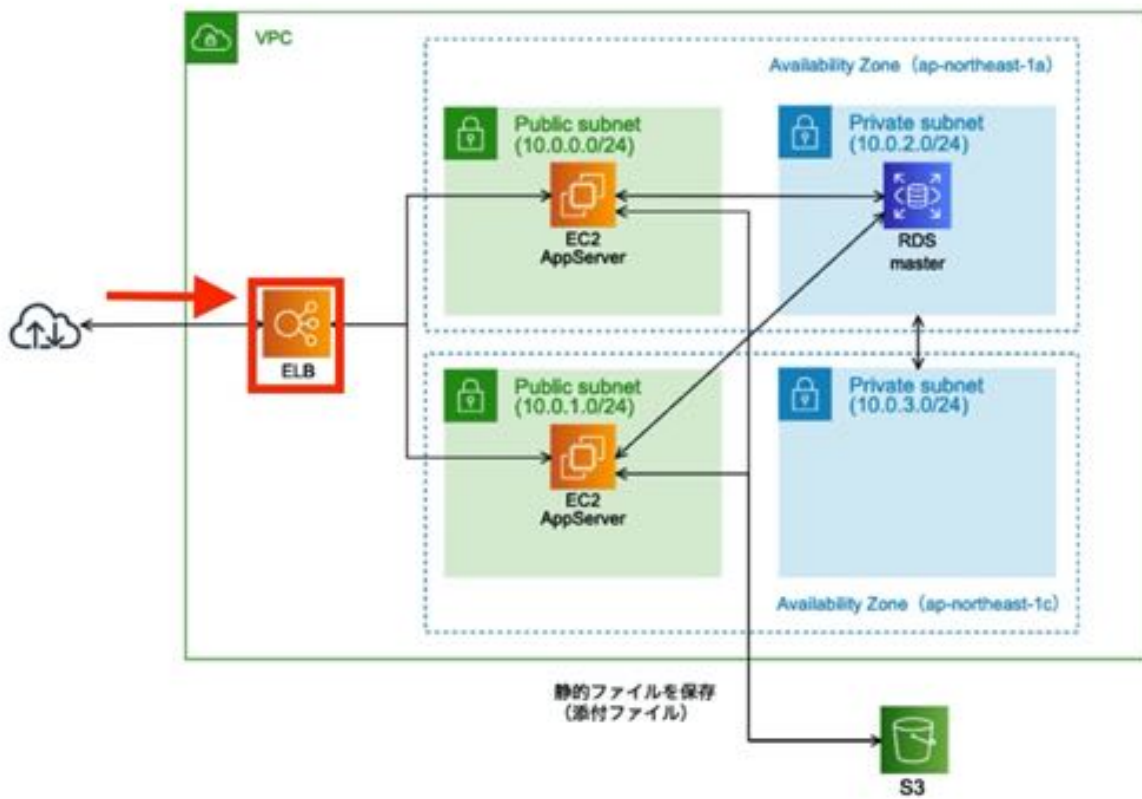
インスタンスの作成が完了するのに数分間かかります。

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一名を入れて絞り込んでください。

webserver2-ユーザ一名、ap-northeast-1c に作成されていることを確認してください。

▼ 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

ロードバランサーの種類の選択

Elastic Load Balancing は 3 種類のロードバランサー (Application Load Balancer、Network Load Balancer (新規)、および Classic Load Balancer) をサポートします。お客様のニーズに合うロードバランサーの種類を選択してください。お客様に最適なロードバランサーの詳細

Application Load Balancer

作成

HTTP および HTTPS トラフィックを使用するウェブアプリケーション用に柔軟性の高い機能セットが必要な場合は、Application Load Balancer を選択します。Application Load Balancer はリクエストレベルで動作し、マイクロサービスとコンテナを含む、アプリケーションアーキテクチャを対象とした高度なルーティングおよび可視性機能を提供します。

[詳細はこちら >](#)

Network Load Balancer

作成

非常に高いパフォーマンス、大規模な TLS のオフロード、証明書デプロイの一元管理、UDP のサポート、およびアプリケーションの静的 IP アドレスが必要な場合は、Network Load Balancer を選択します。Network Load Balancer は接続レベルで動作し、非常に低いレイテンシーを維持しながら、1 秒あたり数百万のリクエストを確実に処理することができます。

[詳細はこちら >](#)

Classic Load Balancer

以前の世代
HTTP、HTTPS、および TCP

1

作成

EC2-Classical ネットワークで既存のアプリケーションを実行している場合は、Classic Load Balancer を選択します。

[詳細はこちら >](#)

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. ロードバランサーの定義 2. セキュリティグループの割り当て 3. セキュリティ設定の構成 4. ヘルスチェックの設定 5. EC2 インスタンスの追加 6. タグの追加 7. 確認

手順 1: ロードバランサーの定義

基本的な設定

このウィザードを使用すると、新しいロードバランサーを設定できます。作成する可能性がある他のロードバランサーと区別できるように、まず新しいロードバランサーに一意の名前を指定します。ロードバランサーのポートとプロトコルも設定する必要があります。クライアントからのトラフィックは、ロードバランサーの任意のポートから EC2 インスタンスの任意のポートにルーティングできます。デフォルトでは、標準の Web サーバーにポート 80 を使用するようにロードバランサーが設定されています。

ロードバランサー名: ①

ロードバランサーを作成する場所: ②

内部向けロードバランサーの作成: (説明)

高度な VPC 設定の有効化:

リスナーの設定:

ロードバランサーのプロトコル	ロードバランサーのポート	インスタンスのプロトコル	インスタンスのポート
HTTP	80	HTTP	80

追加

- 「elb-ユーザ名と入力」と入力します。
例) elb-user1

- 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。

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



- 「パブリックサブネット」のみであることを確認します。
- 「Availability ゾーン」が 2 種類あることを確認します。
- 「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. ロードバランサーの定義 2. セキュリティグループの割り当て 3. セキュリティ設定の構成 4. ヘルスチェックの設定 5. EC2 インスタンスの追加 6. タグの追加 7. 確認

手順 2: セキュリティグループの割り当て

Elastic Load Balancer を VPC 内に配置するオプションを選択しました。この場合、ロードバランサーにセキュリティグループを割り当てることができます。このロードバランサーに割り当てるセキュリティグループを選択してください。これはいつでも変更できます。

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 ①
 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: ②
説明:

タイプ ①	プロトコル ①	ポート範囲 ①	ソース ④
<input checked="" type="radio"/> HTTP ③	TCP	80	<input checked="" type="radio"/> 任意の場所 ④ 0.0.0.0/0

⑤

キャンセル 戻る ⑤ 次の手順: セキュリティ設定の構成

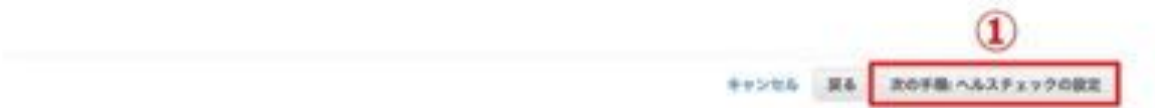
1. 「新しいセキュリティグループを作成する」を選択します。
2. 「elb-ユーザ名」と名前を入力します。
例) elb-user1
3. 「HTTP」を選択します。
4. 「任意の場所」を選択します。
5. 「次の手順: セキュリティ設定の構成」をクリックします。

ステップ 3-3-5: ELB を作成(3)

1. ロードバランサーの定義 2. セキュリティグループの割り当て 3. セキュリティ設定の構成 4. ヘルスチェックの設定 5. EC2 インスタンスの追加 6. タグの追加

ステップ 3: セキュリティ設定の構成

⚠ ロードバランサーのセキュリティの向上。ロードバランサーは、いずれのセキュアリスナーも使用していません。
ロードバランサーへのトラフィックを保護する必要がある場合は、フロントエンドで HTTPS、または、SSL プロトコルのいずれかをお使いください。
高度な設定 セッションのセキュアリスナーの追加設定のために、最初のステップに戻ることができます。または既存の設定のまま実行することもできます。



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)

1. ロードバランサーの定義 2. セキュリティグループの割り当て 3. セキュリティ設定の構成 4. ヘルスチェックの設定 5. EC2 インスタンスの追加 6. タグの追加 7. 確認

手順 4: ヘルスチェックの設定

ロードバランサーは EC2 インスタンスで自動的にヘルスチェックを実行し、ヘルスチェックに合格したインスタンスにのみトラフィックをルーティングします。インスタンスがヘルスチェックに合格しない場合、そのインスタンスはロードバランサーから自動的に削除されます。お使いの要件に合わせてヘルスチェックをカスタマイズしてください。

ping プロトコル

ping ポート

ping パス **1**

高度な詳細

応答タイムアウト 秒 **2**

間隔 秒 **2**

非正常のしきい値

正常のしきい値

キャンセル 戻る **次の手順: EC2 インスタンスの追加** **3**

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)

1. ロードバランサーの定義 2. セキュリティグループの割り当て 3. セキュリティ設定の構成 4. ヘルスチェックの設定 5. EC2 インスタンスの追加 6. タグの追加 7. 確認

手順 5: EC2 インスタンスの追加

以下の表は、実行中のすべての EC2 インスタンスの一覧です。これらのインスタンスをこのロードバランサーに追加するには、[選択] 列のチェックボックスをオンにします。

VPC vpc-0219c5e2bc2073785 (10.0.0.0/16) | handson-user1

選択	インスタンス	名前	状態	セキュリティグループ	ゾーン	サブネット ID	サブネット CIDR
<input checked="" type="checkbox"/>	i-024cdb70e6f117c82	webserver#2-user1	running	web-user1	ap-northe...	subnet-018727f...	10.0.1.0/24
<input checked="" type="checkbox"/>	i-033c85f6ec88dc486	webserver#1-user1	running	web-user1	ap-northe...	subnet-004c450...	10.0.0.0/24

①

アベイラビリティゾーンの分散
 ap-northeast-1a にある 1 個のインスタンス
 ap-northeast-1c にある 1 個のインスタンス

クロスゾーン負荷分散の有効化 ⓘ ⓘ

Connection Drainingの有効化 ⓘ ⓘ 300 秒

キャンセル 戻る **次の手順: タグの追加**

②

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. ロードバランサーの定義 2. セキュリティグループの割り当て 3. セキュリティ設定の構成 4. ヘルスチェックの設定 5. EC2 インスタンスの追加 6. タグの追加 7. 確認

手順 6: タグの追加

リソースを整理、識別しやすいように、リソースにタグを適用します。

タグは、大文字と小文字が区別されるキーと値のペアから構成されます。たとえば、キーに「Name」、値に「Webserver」を使用してタグを定義することができます。Amazon EC2 リソースへのタグ付けに関する [詳細はこちら](#)。

キー	値
Name	elb-user1

タグの作成

キャンセル 戻る **確認と作成**

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。

The screenshot shows the AWS Management Console interface for configuring an ELB instance. The search bar at the top contains 'user1'. Below it, a table lists ELB instances, with 'elb-user1' selected. The configuration details for 'elb-user1' are shown below, including its name, DNS name (highlighted with a red box and circled '3'), creation time, host zone, status, and VPC ID.

名前	elb-user1	作成時刻	2020年1月28日 14:04:58 UTC+9
* DNS 名	elb-user1-236872948.ap-northeast-1.elb.amazonaws.com (A レコード)	ホストゾーン	Z14GRHDCWA56QT
種類	Classic (今すぐ移行)	ステータス	2個のうち0個のインスタンスが実行中です
スキーム	internet-facing	VPC	vpc-0219c5e2bc2073785
アベイラビリティゾーン	subnet-018727121dfd31004 - ap-northeast-1c, subnet-024c450a811ced4fd - ap-northeast-1a		

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。

The screenshot shows the AWS Management Console interface for configuring an ELB instance. The search bar at the top contains 'user1'. Below it, a table lists ELB instances, with 'elb-user1' selected. The configuration details for 'elb-user1' are shown below, including its name, DNS name, creation time, host zone, status, and VPC ID. The 'インスタンス' (Instances) tab is active, showing a table of instances with 'mService' highlighted.

インスタンス ID	名前	アベイラビリティゾーン	状態	アクション
i-113e73c134160cc85	webserver-elb-user1	ap-northeast-1c	mService (i)	ロードバランサーから削除
i-02a2267a13030bc	webserver-elb-user1	ap-northeast-1a	mService (i)	ロードバランサーから削除

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. ユーザー名で絞り込みます。
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。

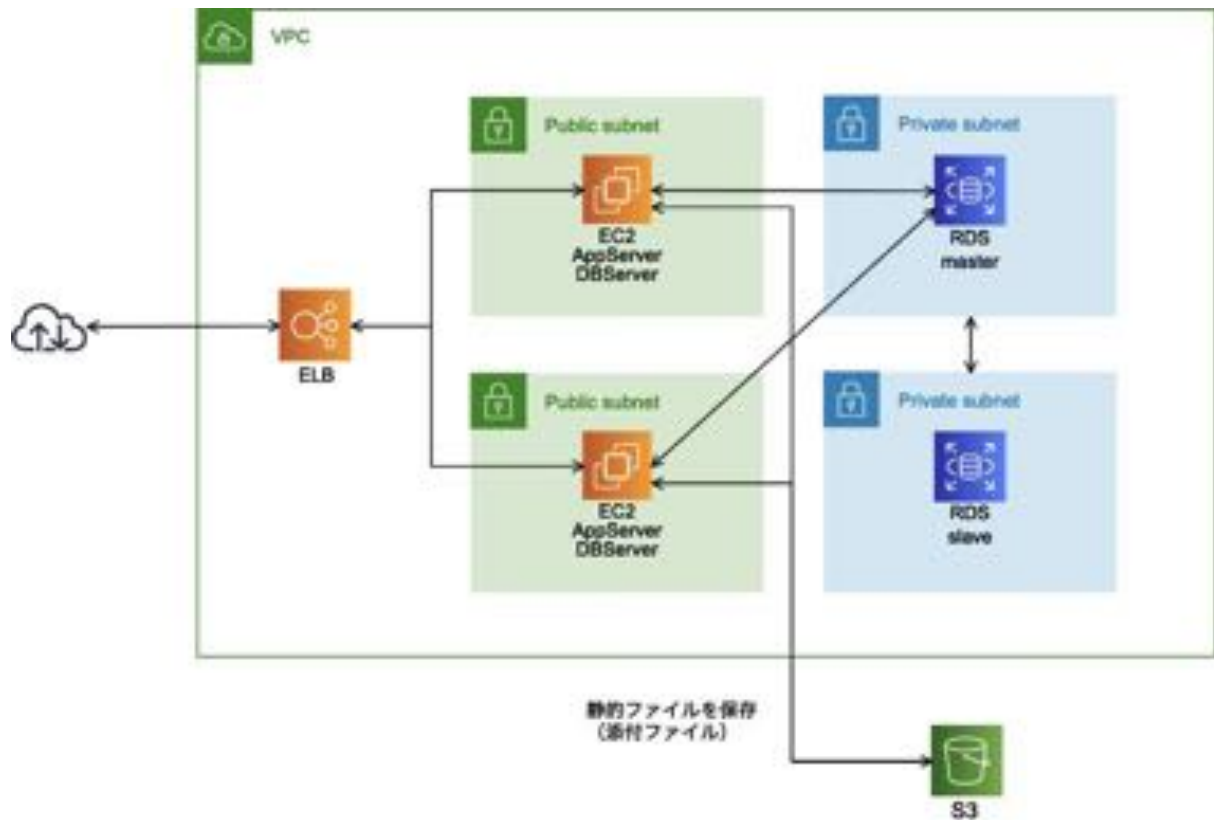


1. 「elb」と入力して候補を表示させます。

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

RDS > データベース > 変更

DB インスタンスの変更: wp-user1

インスタンスの仕様

DB エンジンのバージョン
このインスタンスに使用するデータベースエンジンのバージョン番号。

MySQL 5.7.22 (デフォルト) ▼

DB インスタンスのクラス
DB インスタンスのコンピューティング能力とメモリ容量が含まれます。

db.t2.micro — 1 vCPU, 1 GiB RAM ▼

マルチ AZ 配置
DB インスタンスのスタンバイを別のアベイラビリティゾーンに配置するかどうかを指定します。

はい 1

いいえ

ストレージタイプ

汎用 (SSD) ▼

ストレージ割り当て

GiB

このインスタンスは、20~16384 GiB の複数のストレージ範囲をサポートします。 [すべて表示](#)

キャンセル 2 次へ

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。

RDS > データベース > 変更

DB インスタンスの変更: wp-user1

変更の概要

以下の変更を送信しようとしています。変更される値のみが表示されます。変更をよく確認してから、[DB インスタンスの変更] をクリックしてください。

属性	現在の値	新しい値
マルチ AZ 配置	いいえ	はい

⚠ Potential performance impact
 You may experience a significant performance impact when converting this database instance to Multi-AZ configuration. This impact will be more noticeable on database instances with large amounts of storage and write-intensive workloads.

変更のスケジュール

変更を適用する時間

次に予定されるメンテナンスウィンドウ中に適用します 1
最新のメンテナンスウィンドウ: thu:18:34-thu:19:04

すぐに適用
 このデータベースインスタンスのメンテナンスウィンドウ設定に関わらず、このリクエストの変更とすべての保留中の変更はできるだけ早く非同期に適用されます。

⚠ 予期されないダウンタイムの可能性
 変更の即時適用を選択した場合、保留中の変更キューにあるすべての変更も同様に適用されます。ダウンタイムを必要とする保留中の変更がある場合、即時適用を選択すると予想外のダウンタイムが発生することがあります。

2

キャンセル

「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。

The screenshot displays the Amazon RDS console for a MySQL instance named 'redmine-user1'. The left sidebar shows navigation options like 'ダッシュボード', 'データベース', and 'Query Editor'. The main content area is titled 'redmine-user1' and includes a '概要' (Summary) section with a table of instance details. A red box highlights the '情報' (Info) section, which shows '利用可能' (Available) with a circled '1'. Below this, the '接続とセキュリティ' (Connections and Security) section is visible, containing details for endpoints, ports, network, and security groups.

概要			
DB 識別子 redmine-user1	CPU 2.67%	情報 利用可能	クラス db.t2.micro
ロール インスタンス	現在のアクティビティ 2 接続	エンジン MySQL Community	リージョンと AZ ap-northeast-1a

接続とセキュリティ		
エンドポイントとポート	ネットワーク	セキュリティ
エンドポイント redmine-user1.cizpucnnhfj8ap-northeast-1.rds.amazonaws.com	アベイラビリティゾーン ap-northeast-1a	VPC セキュリティグループ db-user1 (sg-099eb01756122c893) (アクティブ)
ポート 3306	VPC handson-user1 (vpc-0a941f74723ca26f2)	パブリックアクセス なし
	サブネットグループ db subnet user1	認証機関 rds-ca-2019
	サブネット subnet-05dae220b74f2a8c1 subnet-082ba9fb5ee8ec5a86	証明機関の日付 Aug 23rd, 2024

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

The screenshot shows the Amazon RDS console interface for a MySQL instance named 'redmine-user1'. The left sidebar contains navigation options like 'ダッシュボード', 'データベース', 'Query Editor', etc. The main content area shows the instance details and configuration options. The '設定' (Settings) tab is highlighted with a red box and a circled '1'. Below it, the '可用性' (Availability) section shows 'マルチAZ' (Multi-AZ) is checked, also highlighted with a red box and a circled '2'.

概要			
DB 識別子 redmine-user1	CPU 2.67%	情報 利用可能	クラス db.t2.micro
ロール インスタンス	現在のアクティビティ 2 接続	エンジン MySQL Community	リージョンと AZ ap-northeast-1a

インスタンス			
設定	インスタンスクラス	ストレージ	Performance Insights
DB インスタンス ID redmine-user1	インスタンスクラス db.t2.micro	暗号化 有効でない	Performance Insights が 有効 なし
エンジンバージョン 5.7.22	vCPU 1	ストレージタイプ 汎用 (SSD)	
DB 名 rds_redmine	RAM 1 GB	IOPS -	
ライセンスモデル General Public License	可用性	ストレージ 20 GiB	
オプショングループ default:mysql-5-7	マスターユーザー名 admin	ストレージの自動スケール リング 有効	
ARN arn:aws:rds:ap-northeast-1:533584410763:db:redmine-user1	IAM db 認証 有効でない	最大ストレージしきい値 1000 GiB	
	マルチ AZ あり		

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も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