サーバ構築手順書

目的

パブリッククラウド上にWordPressを構築しブログサイトを公開

使用ツール

**【サーバ】**

Amazon EC2

・インスタンスタイプ：t2.micro

　・OS　　　　　　　　:Amazon Linux 2023

**【ソフトウェア】**

Apache

PHP

MariaDB

WordPress

**【実行環境】**

Microsoft Windows [Version 10.0]

Tera Term

手順

1.インスタンス作成

1-1.EC2でインスタンスを作成

　AWSへログイン後、サービス＞コンピューティング＞EC2を順に選択し、EC2のメニューへ移動。「インスタンスを起動する」押下し、インスタンスの作成を開始する。

1-2.OS・インスタンスタイプを選択

　AMI(Amazon Machine Image)はAmazon Linuxを選択する。

　インスタンスタイプはt2.microを選択。

1-3.キーペアを作成、選択

　「新しいキーペアの作成」を押下し、作成したインスタンスにログインするために用いる認証鍵を作成します。キーペアの各設定値は下記項目を選択後、「キーペアを作成」を押下。

　　・キーペア名　　　　　　　　　：任意の名前

　　・キーペアのタイプ　　　　　　：RSA

　　・プライベートキーファイル形式：.pem

　RSAでの認証は、v4.xxxのような古いバージョンのTera Termに対応していないため、v5.0以降を使用できることを確認すること。

キーファイルは、インスタンス作成後Tera Termを用いてssh接続を行うため、pem形式を選択する。

1-4.ネットワークの設定

　ファイアウォールの欄で、「セキュリティグループを作成」を選択し、SSHとHTTPプロトコルによるトラフィックを許可する。

1-5.インスタンスを起動する

　上記設定完了後、概要タブの一番下にある、「インスタンスを起動」を押下。

以上でインスタンスの作成は完了。

2.Wordpress構築

2-1.EC2インスタンスに接続

　作成したEC２インスタンスにTera Termで接続する。ホストには、EC2インスタンスのパブリックIPv4アドレスを入力。サービスはSSHを選択し「OK」を押下。認証方法は下記の通り。

　　・ユーザー名：ec2-user

　　・認証方式　：「RSA/CSA/ECDSA/ED25519鍵を使う」を選択し、秘密鍵には作成したpemファイルを指定

2-2.ApacheとPHPをインストール、セットアップ

接続成功後、ApacheとPHPパッケージをインストールする。

|  |
| --- |
| **[ec2-user~]$ sudo dnf install -y httpd wget php-fpm php-mysqli php-json php php-devel** |

「Complete!」の文字を確認し、EC2インスタンス起動。サーバ起動と同時にApacheも自動で起動する設定を行う。

|  |
| --- |
| **[ec2-user~]$ sudo systemctl start httpd**  **[ec2-user~]$ sudo systemctl enable httpd.service** |

実行後、Apacheのインストールがされていること、自動起動が有効化されていることを確認する。

テキスト

自動的に生成された説明

続いてec2-userアカウントがApacheドキュメントルート(var/www/html)に格納されるファイルのアクセス権と所有権を持つよう設定する。今回は、ec2-userをApacheグループに追加し、var/wwwディレクトリの所有権をApacheグループに付与する。

始めに、ec2-userをApacheグループに追加。

|  |
| --- |
| **[ec2-user~]$ sudo usermod -a -G apache ec2-user** |

実行後、一度ログアウトし再度ログインしてec2-userが所属するグループにApacheが追加されていることを確認する。

|  |
| --- |
| **[ec2-user~]$ groups**  **ec2-user adm wheel apache systemd-journal** |

ec2-userがApacheグループに所属していることを確認し、/var/wwwディレクトリのコンテンツのグループ所有権をApacheに変更する。

|  |
| --- |
| **[ec2-user~]$ sudo chown -R ec2-user:apache /var/www** |

続いて新規ファイルを/var/wwwディレクトリ下に所属させ、ecc2-userが読み取りや書き込み権限を保持するよう設定。

|  |
| --- |
| **[ec2-user~]$ sudo chmod 2775 /var/www**  **[ec2-user~]$ find /var/www -type d -exec sudo chmod 2775 {} \;**  **[ec2-user~]$ find /var/www -type f -exec sudo chmod 0664 {} \;** |

最後に、ファイルへのアクセス権限の設定が正常に行われたことをテストする。/var/www/htmlディレクトリにPHPファイルを作成し、webブラウザでPHP情報ページが表示されることを確認する。

/var/www/html配下にPHPファイルを作成。

|  |
| --- |
| **[ec2-user~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php** |

webブラウザで作成したファイルを表示。赤い部分をインスタンスのパブリックIPv4 DNSアドレスに置き換え入力すれば下記ページが表示される。

http://my.public.dns.amazonaws.com/phpinfo.php

テーブル

自動的に生成された説明

テスト後、セキュリティ保護のためPHP情報ページは削除する。

|  |
| --- |
| **[ec2-user~]$ rm /var/www/html/phpinfo.php** |

2-3. MariaDBをインストール、セットアップ

　続いてMariaDBのインストールを行う。

|  |
| --- |
| **[ec2-user~]$ sudo dnf install mariadb105-server** |

インストール完了後、Apache同様インスタンス起動時にMariaDBも自動起動する設定を行う。

|  |
| --- |
| **[ec2-user~]$ sudo systemctl start mariadb**  **[ec2-user~]$ sudo systemctl enable mariadb** |

自動起動の設定が反映されていることも確認する。

テキスト

自動的に生成された説明

データベースにパスワードを設定し、不必要なユーザやリモートログインを無効化しセキュリティを強化する。

|  |
| --- |
| **[ec2-user~]$ sudo mysql\_secure\_installation** |

Ⅰ)Enterを押下

Ⅱ)「Y」と入力し新しくパスワードを設定

Ⅲ)「Y」と入力し匿名ユーザを削除

Ⅳ)「Y」と入力しリモートログインを無効化

Ⅴ)「Y」と入力しテストデータベースを削除

Ⅵ)「Y」と入力し変更を保存

2-4.WordPress用データベース作成

データベースサーバにrootユーザとしてログイン。パスワードは前工程で設定したパスワードを入力する。

|  |
| --- |
| **[ec2-user~]$ mysql -u root -p** |

データベースのユーザとパスワードを新たに作成。

|  |
| --- |
| **MariaDB [(none)]> CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';** |

データベースを作成し、前工程で設定したユーザに対して権限を付与する。

|  |
| --- |
| **MariaDB [(none)]>CREATE DATABASE `wordpress-db`;**  **MariaDB [(none)]>GRANT ALL PRIVILEGES ON `wordpress-db`.\* TO "user"@"localhost";** |

変更を反映するためデータベース権限をフラッシュし、データベースからログアウトする。

|  |
| --- |
| MariaDB [(none)]>FLUSH PRIVILEGES;  MariaDB [(none)]>exit |

2-5.構成ファイルの設定変更

続いてWordPressが正常に動作するよう、WordPressがデータベースの接続情報とセキュリティキーを持つように設定を変更する。

まず始めに、変更する構成ファイルのバックアップを作成。

|  |
| --- |
| **[ec2-user~]$** **cp wordpress/wp-config-sample.php wordpress/wp-config.php** |

テキストエディタ「nano」を指定し、vimファイルの編集に移行する。

|  |
| --- |
| **[ec2-user~]$ nano wordpress/wp-config.php** |

「DB\_NAME」の行を探し、2-4で作成したデータベース名に書き換える。

テキスト

自動的に生成された説明

「DB\_USER」の行を探し、2-4で作成したデータベースユーザに書き換える。

テキスト

自動的に生成された説明

「DB\_PASSWORD」の行を探し、2-4で作成したデータベースに設定したパスワードに書き換える。

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

「Authentication Unique Keys and Salts」セクションの各種値を設定。

今回は下記リンクを用い、ランダムに生成された文字列を使用する。

<https://api.wordpress.org/secret-key/1.1/salt/>

テキスト

自動的に生成された説明

「Ctrl + O」を押下、設定を保存し、「Ctrl + X」で編集を終了する。

2-6.WordPressインストール

　インターネットからWordPressの最新のインストールパッケージをダウンロードし、パッケージを解凍する。

|  |
| --- |
| **[ec2-user~]$ wget https://wordpress.org/latest.tar.gz**  **[ec2-user~]$ tar -xzf latest.tar.gz** |

ドキュメントルートでWordPressを実行するため、WordPressのインストールパッケージを下記の通りにコピーする。

|  |
| --- |
| **[ec2-user~]$ cp -r wordpress/\* /var/www/html/** |

webブラウザでパブリックIPv4アドレスを入力。各設定値を入力しWordpressをインストールする。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明