# WEB アプリケーション説明書

21d2030006

カンシン

# 目次

	WE	EB ~	<u> </u>	ジ	の糸	引介	٠ ١	•	•		•	•	•	•	•	•	• (		•	•		•	•	•		•	•	•	•	• ;	3	
	作成	<b>文手</b> 师	<b>頁•</b>	•		•	• •	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	4	
	EC2	2サ-	ーノヾ	`-	の1	乍月	戏	•	•	•	•		•		•	•	•		•	•		•	•	•		•	•		•	•	4	
	SSH	[接網	売 <b>・</b>	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	8	
	Doc	ker		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	9	
Νę	ginx			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1	1
PΗ	IP•		•	•	•	•	•	•	•	•	•	• •	•	• •	• •	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• 1	2
M	YSQ	L·			•		•			•		•		•	•		•	•	•	•	•	•			•	•				•	1	4

## WEB ページ紹介

#### ページ説明

誰でも自由にテキストや画像を投稿できる掲示板の WEB サイト

### 構築要素

AWS EC2

Docker

Nginx

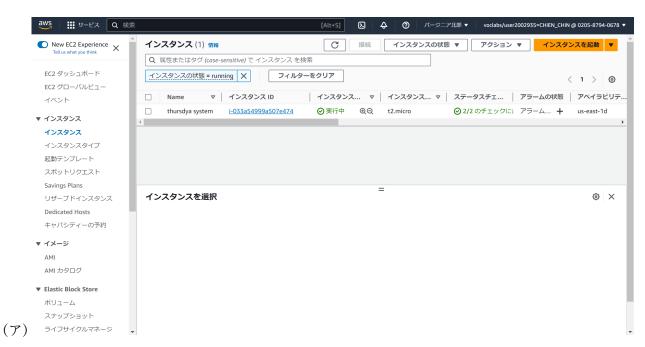
MY SQL

#### 効能

- 投稿者が自由にテキストを投稿できること
  - ▶ 内容は MySQL に保存すること
  - ➤ XSS および SQL インジェクションの対策ができていること
- 投稿それぞれに自動で投稿日時が付与されていること
- 投稿それぞれに自動で連番が付与されていること
- 投稿者が自由に画像を投稿できること
  - ▶ 5MB以上の画像をアップロードできないように
  - ➤ 画像をブラウザ側(JavaScript で実装)で 5MB 以下に自動縮小しア ップロードするように (これができたら+15 点)

# 作成手順

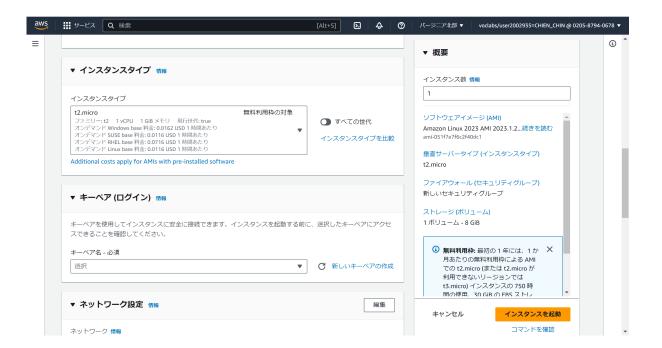
### 1. EC2 サーバー構築



インスタンスを起動を選択



好きな名前を入力する



### キーペアを作成



#### キーペア名

キーペアを使用すると、インスタンスに安全に接続できます。

#### キーペア名を入力

名前には最大 255 文字の ASCII 文字を使用できます。先頭または末尾にスペースを含めることはできま

#### キーペアのタイプ

#### RSA

RSA で暗号化されたプライベートとパ ブリックのキーペア

O ED25519

ED25519 で暗号化されたプライベート キーとパブリックキーのペア

#### プライベートキーファイル形式

o .pem OpenSSH で使用する場合

O .ppk

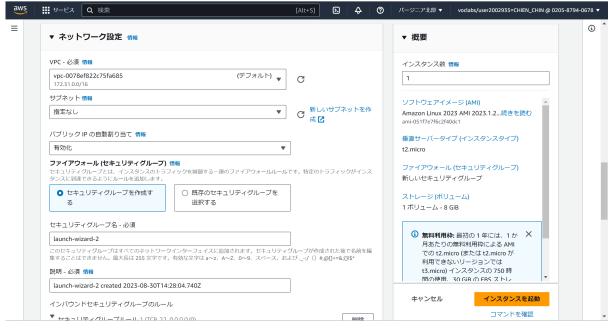
PuTTY で使用する場合

↑ プロンプトが表示されたら、コンピュータの安全でアクセス可能な場所にプラ イベートキーを保存してください。 後でインスタンスに接続するときに必要に なります。 詳細はこちら 🔼

キャンセル

キーペアを作成

キーペア名を入力して図のように選択する





セキュリティグループを図のように設定する

設定が完了したらインスタンスを起動する

# 2. SSH 接続

Powershell を開いて

SSH で接続するコマンドを打ちます

ssh ec2-user@{EC2 インスタンスのアドレス} -i {.pem ファイルのパス}

{EC2 インスタンスのアドレス}は AWS の EC2 インスタンスで確認することができる {.pem ファイルのパス}は先ほど作ったキーペア

### 3. Docker

Docker とは

仮想的な環境を簡単に作るための技術です。仮想的な環境は簡単に作成と削除を行うことができ,面倒なミドルウェアのインストール・設定を毎回行う必要がなくなります。

仮想的な環境のことをコンテナと呼びます。一つのサーバー(コンピューター)内に沢山作ることができ、また同時に動かすことができます。

(ア) Docker インストール方法 & 自動起動化

コマンドを打ち込みます

sudo yum install -y docker sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -a -G docker ec2-user

(イ) Docker Compose インストール方法

複数の Docker のコンテナを簡単に扱うためのツール(Docker コマンドのサブコマンド)です。 コマンドを打ち込みます

sudo mkdir -p /usr/local/lib/docker/cli-plugins/

sudo curl -SL https://github.com/docker/compose/releases/download/v2.2.2/docker-compose-linux-x86\_64 -o /usr/local/lib/docker/cli-plugins/docker-compose sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose

(ウ) Docker Compose は、カレントディレクトリにある compose.yml という設定ファイルを参照します。(もちろんオプションで他の場所にある設定ファイルを参照することもできます。)

まずは作業用のディレクトリを作ってその中に移動しましょう。

mkdir dockertest cd dockertest

そのため、まずは設定ファイルを書きましょう。

vim compose.yml

```
services:
          image: nginx:latest
         ports:
           - 80:80
         volumes:
           - ./nginx/conf.d/:/etc/nginx/conf.d/
           - ./public/:/var/www/public/
            - image:/var/www/upload/image/
10
          depends_on:
11
           - php
       php:
12
13
          container_name: php
14
         build:
15
           context: .
16
           target: php
17
         volumes:
18
           - ./public/:/var/www/public/
19
            - image:/var/www/upload/image/
20
       mysql:
21
          container_name: mysql
22
         image: mysql:8.0
23
         environment:
          MYSQL DATABASE: techc
24
25
           MYSQL_ALLOW_EMPTY_PASSWORD: 1
26
            TZ: Asia/Tokyo
27
         volumes:
28
           - mysql:/var/lib/mysql
29
        command: >
           mysqld
31
            --character-set-server=utf8mb4
32
           --collation-server=utf8mb4 unicode ci
33
            --max_allowed_packet=4MB
34
      volumes:
35
       image:
36
        driver: local
        mysql:
```

#### (エ) PHP の拡張モジュール

PHP の拡張モジュールを利用するために カレントディレクトリに Dockerfile を作成しましょう コードは以下のようになります。

```
FROM php:8.1-fpm-alpine AS php

RUN docker-php-ext-install pdo_mysql

RUN install -o www-data -g www-data -d /var/www/upload/image/

RUN echo -e "post_max_size = 5M\nupload_max_filesize = 5M" >> ${PHP_INI_DIR}/php.ini
```

# 4. Nginx

自分の作ったファイルや、プログラムの出力結果も Web に配信することがでる WEB サーバーまず設定ファイル用のディレクトリを作ります。

mkdir nginx

mkdir nginx/conf.d

設定ファイルを作成

vim nginx/conf.d/default.conf

ファイルのコードは以下のようになります

```
1
       server {
           listen
                        0.0.0.0:80;
           server_name _;
           charset
                        utf-8;
 5
           client_max_body_size 6M;
 6
 7
           root /var/www/public;
 8
 9
           location ~ \.php$ {
10
               fastcgi_pass php:9000;
               fastcgi_index index.php;
11
               fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
12
13
                include
                           fastcgi_params;
14
           }
15
           location /image/ {
16
17
               root /var/www/upload;
18
           }
19
       }
```

### 5. PHP

配信するページの実装

配信するページ用のディレクトリを作ります。

mkdir public

配信ページを作成

vim public/bbsimagetest.php

中身のコードは以下のようになります。

```
$dbh = new PDO('mysql:host=mysql;dbname=techc', 'root', '');
      if (isset($_POST['body'])) {
        // POSTで送られてくるフォームパラメータ body がある場合
       if (isset($_FILES['image']) && !empty($_FILES['image']['tmp_name'])) {
         // アップロードされた画像がある場合
         if (preg_match('/^image\//', mime_content_type($_FILES['image']['tmp_name'])) !== 1) {
10
          // アップロードされたものが画像ではなかった場合
           header("HTTP/1.1 302 Found");
           header("Location: ./bbsimagetest.php");
13
14
        // 元のファイル名から拡張子を取得
         $pathinfo = pathinfo($_FILES['image']['name']);
17
         $extension = $pathinfo['extension'];
18
19
         // 新しいファイル名を決める。他の投稿の画像ファイルと重複しないように時間+乱数で決める。
         $image_filename = strval(time()) . bin2hex(random_bytes(25)) . '.' . $extension;
         $filepath = '/var/www/upload/image/' . $image_filename;
21
         move_uploaded_file($_FILES['image']['tmp_name'], $filepath);
22
23
        // insertする
        $insert_sth = $dbh->prepare("INSERT INTO bbs_entries (body, image_filename) VALUES (:body, :image_filename)");
26
27
        $insert_sth->execute([
         ':body' => $_POST['body'],
         ':image_filename' => $image_filename,
```

```
31
32
        // 処理が終わったらリダイレクトする
       // リダイレクトしないと、リロード時にまた同じ内容でPOSTすることになる
33
        header("HTTP/1.1 302 Found");
34
       header("Location: ./bbsimagetest.php");
35
36
       return;
37
38
      // いままで保存してきたものを取得
39
      $select_sth = $dbh->prepare('SELECT * FROM bbs_entries ORDER BY created_at DESC');
40
41
42
43
      <head>
       <title>画像投稿できる掲示板</title>
44
45
      </head>
      <!-- フォームのPOST先はこのファイル自身にする -->
47
      <form method="POST" action="./bbsimagetest.php" enctype="multipart/form-data">
48
49
        <textarea name="body"></textarea>
       <div style="margin: 1em 0;">
        <input type="file" accept="image/*" name="image" id="imageInput">
51
       </div>
52
53
       <button type="submit">送信</button>
54
      </form>
55
56
      <hr>>
57
58
      <?php foreach($select_sth as $entry): ?>
       <dl style="margin-bottom: 1em; padding-bottom: 1em; border-bottom: 1px solid #ccc;">
59
         <dt>ID</dt>
60
         <dd><?= $entry['id'] ?></dd>
61
62
         <dt>日時</dt>
63
         <dd><?= $entry['created_at'] ?></dd>
64
         <dt>内容</dt>
65
         <dd>
           <?= nl2br(htmlspecialchars($entry['body'])) // 必ず htmlspecialchars() すること ?>
66
67
            <?php if(!empty($entry['image_filename'])): // 画像がある場合は img 要素を使って表示 ?>
68
             <img src="/image/<?= $entry['image_filename'] ?>" style="max-height: 10em;">
69
70
           </div>
71
           <?php endif; ?>
72
         </dd>
        </dl>
73
74
      <?php endforeach ?>
75
      document.addEventListener("DOMContentLoaded", () => {
77
       const imageInput = document.getElementById("imageInput");
78
79
       imageInput.addEventListener("change", () => {
80
         if (imageInput.files.length < 1) {</pre>
          // 未選択の場合
81
           return:
82
83
84
         if (imageInput.files[0].size > 5 * 1024 * 1024) {
           // ファイルが5MBより多い場合
85
           alert("5MB以下のファイルを選択してください。");
86
87
           imageInput.value = "";
88
89
       });
90
      });
91
      </script>
```

# 6. MY SQL

ALTER TABLE `bbs\_entries` ADD COLUMN image\_filename TEXT DEFAULT NULL;