Kadaiページ表示方法

1.EC2インスタンスの作成

2.EC2インスタンスへPCからSSHでログイン

ssh ec2-user@{IPアドレス} -i {秘密鍵ファイルのパス}

3.sudo yum install vim -y

4.vim ~/.vimrc（vimの設定を変更）

“set number

set expandtab

set tabstop=2

set shiftwidth=2

set autoindent

“

5.sudo yum install screen -y

6.screen1(のちにdocker compose,php編集画面,sql入力画面とタブを分けるのがおすすめ)

7.vim ~/.screenrc（screenの設定変更）

hardstatus alwayslastline "%{= bw}%-w%{= wk}%n%t\*%{-}%+w"

8.sudo yum install -y docker（dockerのインストール）

9.sudo systemctl start docker

10.sudo systemctl enable docker

11.sudo usermod -a -G docker ec2-user

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

13.sudo curl -SL <https://github.com/docker/compose/releases/download/v2.36.0/docker>-compose-linux-x86\_64 -o /usr/local/lib/docker/cli-plugins/docker-compose

13.sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose

14.docker compose version

15.mkdir dockertest

16.cd dockertest

17.vim compose.yml

“

services:

web:

image: nginx:latest

ports:

- 80:80

volumes:

- ./nginx/conf.d/:/etc/nginx/conf.d/

- ./public/:/var/www/public/

- image:/var/www/upload/image/

depends\_on:

- php

php:

container\_name: php

build:

context: .

target: php

volumes:

- ./public/:/var/www/public/

- image:/var/www/upload/image/

mysql:

container\_name: mysql

image: mysql:8.4

environment:

MYSQL\_DATABASE: example\_db

MYSQL\_ALLOW\_EMPTY\_PASSWORD: 1

TZ: Asia/Tokyo

volumes:

- mysql:/var/lib/mysql

command: >

mysqld

--character-set-server=utf8mb4

--collation-server=utf8mb4\_unicode\_ci

--max\_allowed\_packet=4MB

volumes:

mysql:

image:

”

18.mkdir nginx

19.mkdir nginx/conf.d

20.vim nginx/conf.d/default.conf

“

server {

listen 0.0.0.0:80;

server\_name \_;

charset utf-8;

root /var/www/public;

client\_max\_body\_size 6M;

location ~ \.php$ {

fastcgi\_pass php:9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME /var/www/public$fastcgi\_script\_name;

include fastcgi\_params;

}

location /image/ {

root /var/www/upload;

}

}

“

21.mkdir public

22.vim Dockerfile

“

FROM php:8.4-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 docker-php-ext-install fileinfo && docker-php-ext-enable fileinfo

COPY uploads.ini /usr/local/etc/php/conf.d/uploads.ini

“

23.docker compose exec mysql mysql example\_db

“

CREATE TABLE `example\_db`.`bbs\_entries` (

`id` INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

`body` TEXT NOT NULL,

`image\_filename` VARCHAR(255),

`created\_at` DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP

);

“

24.vim public/kadai.php

‘

<?php

$dbh = new PDO('mysql:host=mysql;dbname=example\_db', '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

$image\_filename = null;

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) {

// アップロードされたものが画像ではなかった場合処理を強制的に終了

header("HTTP/1.1 302 Found");

header("Location: ./kadai.php");

return;

}

// 元のファイル名から拡張子を取得

$pathinfo = pathinfo($\_FILES['image']['name']);

$extension = $pathinfo['extension'];

// 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;

move\_uploaded\_file($\_FILES['image']['tmp\_name'], $filepath);

}

// insertする

$insert\_sth = $dbh->prepare("INSERT INTO bbs\_entries (body, image\_filename) VALUES (:body, :image\_filename)");

$insert\_sth->execute([

':body' => $\_POST['body'],

':image\_filename' => $image\_filename,

]);

// 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

header("HTTP/1.1 302 Found");

header("Location: ./kadai.php");

return;

}

// いままで保存してきたものを取得

$select\_sth = $dbh->prepare('SELECT \* FROM bbs\_entries ORDER BY created\_at DESC');

$select\_sth->execute();

?>

<!-- フォームのPOST先はこのファイル自身にする -->

<form method="POST" action="./kadai.php" enctype="multipart/form-data">

<textarea name="body" required></textarea>

<div style="margin: 1em 0;">

<input type="file" accept="image/\*" name="image" id="imageInput">

</div>

<button type="submit">送信</button>

</form>

<hr>

<?php foreach($select\_sth as $entry): ?>

<dl style="margin-bottom: 1em; padding-bottom: 1em; border-bottom: 1px solid #ccc;">

<dt>ID</dt>

<dd><?= $entry['id'] ?></dd>

<dt>日時</dt>

<dd><?= $entry['created\_at'] ?></dd>

<dt>内容</dt>

<dd>

<?= nl2br(htmlspecialchars($entry['body'])) // 必ず htmlspecialchars() すること ?>

<?php if(!empty($entry['image\_filename'])): // 画像がある場合は img 要素を使って表示 ?>

<div>

<img src="/image/<?= $entry['image\_filename'] ?>" style="max-height: 10em;">

</div>

<?php endif; ?>

</dd>

</dl>

<?php endforeach ?>

<script>

document.addEventListener("DOMContentLoaded", () => {

const imageInput = document.getElementById("imageInput");

imageInput.addEventListener("change", () => {

if (imageInput.files.length < 1) {

// 未選択の場合

return;

}

if (imageInput.files[0].size > 5 \* 1024 \* 1024) {

// ファイルが5MBより多い場合

alert("5MB以下のファイルを選択してください。");

imageInput.value = "";

}

});

});

</script>

‘

25.docker compose up –build（一度sshログインしなおしたほうが良いかも）