ssh ec2-user@52.91.11.67 -i "C:\Users\ktc\Downloads\kadaikey.pem"

**Amazon Linux 2 でのvimのインストール方法**

sudo yum install vim -y

vim ~/.vimrc を開いて

set number

set expandtab

set tabstop=2

set shiftwidth=2

set autoindent

**screenのインストール**

sudo yum install screen -y

**screen起動**

screen

**.screenrc**

vim ~/.screenrc

**中身**

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

**Dockerインストール方法**

sudo yum install -y docker

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -a -G docker ec2-user

**Docker Composeインストール方法**

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

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

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

**ディレクトリの作成**

mkdir dockertest

cd dockertest

**設定ファイル**

vim compose.yml

中身

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:

**nginxでファイル配信**

mkdir nginx

mkdir nginx/conf.d

**設定ファイルを作成**

vim nginx/conf.d/default.conf

**中身**

server {

listen 0.0.0.0:80;

server\_name \_;

charset utf-8;

root /var/www/public;

location ~ \.php$ {

fastcgi\_pass php:9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

location /image/ {

root /var/www/upload;

}

}

**配信するディレクトリを作る**

mkdir public

**配信するファイルを作る**

vim public/kadai.php

**中身**

<?php

// ===== ここより前に空白/改行/文字を絶対に置かない（BOMなしUTF-8で保存）=====

// ===== DB接続 =====

$dsn = 'mysql:host=mysql;dbname=example\_db;charset=utf8mb4';

$user = 'root';

$pass = '';

$options = [

PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,

];

$dbh = new PDO($dsn, $user, $pass, $options);

// ===== セキュリティヘッダ（必ず出力前に）=====

header("X-Content-Type-Options: nosniff");

header("Referrer-Policy: no-referrer-when-downgrade");

header("Content-Security-Policy: default-src 'self'; img-src 'self' data:; style-src 'self' 'unsafe-inline';");

// ===== 画像保存先（実ディスク）=====

$UPLOAD\_DIR = '/var/www/upload/image';

if (!is\_dir($UPLOAD\_DIR)) {

mkdir($UPLOAD\_DIR, 0755, true);

}

// ===== MIME→拡張子の安全マップ =====

function ext\_from\_mime(string $mime): ?string {

static $map = [

'image/jpeg' => 'jpg',

'image/png' => 'png',

'image/gif' => 'gif',

'image/webp' => 'webp',

];

$mime = strtolower($mime);

return $map[$mime] ?? null;

}

// ===== MIME判定（finfo優先、無ければ mime\_content\_type フォールバック）=====

function detect\_mime(string $tmpPath): ?string {

if (class\_exists('finfo')) {

$finfo = new finfo(FILEINFO\_MIME\_TYPE);

return $finfo->file($tmpPath) ?: null;

}

if (function\_exists('mime\_content\_type')) {

return @mime\_content\_type($tmpPath) ?: null;

}

return null; // 最悪の場合は null

}

// ===== POST（新規投稿）=====

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['body'])) {

$body = trim($\_POST['body']);

// 本文バリデーション

if ($body === '' || mb\_strlen($body) > 2000) {

header('HTTP/1.1 302 Found');

header('Location: ./kadai.php?err=body');

exit;

}

$image\_filename = null;

// 画像が来ていれば検証

if (isset($\_FILES['image']) && is\_uploaded\_file($\_FILES['image']['tmp\_name'])) {

// 5MB制限（サーバ側）

$max = 5 \* 1024 \* 1024; // 5MB

if ($\_FILES['image']['size'] > $max) {

header('HTTP/1.1 302 Found');

header('Location: ./kadai.php?err=size');

exit;

}

// MIMEチェック（拡張子は信用しない）

$mime = detect\_mime($\_FILES['image']['tmp\_name']);

$ext = $mime ? ext\_from\_mime($mime) : null;

if ($ext === null) {

header('HTTP/1.1 302 Found');

header('Location: ./kadai.php?err=mime');

exit;

}

// 衝突しないファイル名

$image\_filename = sprintf('%s.%s', bin2hex(random\_bytes(16)), $ext);

$dest = rtrim($UPLOAD\_DIR, '/').'/'.$image\_filename;

if (!move\_uploaded\_file($\_FILES['image']['tmp\_name'], $dest)) {

header('HTTP/1.1 302 Found');

header('Location: ./kadai.php?err=move');

exit;

}

@chmod($dest, 0644);

}

// INSERT（SQLi対策：prepared）

$sql = "INSERT INTO bbs\_entries (body, image\_filename) VALUES (:body, :image\_filename)";

$stmt = $dbh->prepare($sql);

$stmt->execute([

':body' => $body,

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

]);

// 二重送信防止リダイレクト

header('HTTP/1.1 302 Found');

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

exit;

}

// ===== 一覧取得 =====

$stmt = $dbh->prepare('SELECT id, body, image\_filename, created\_at FROM bbs\_entries ORDER BY created\_at DESC');

$stmt->execute();

$entries = $stmt->fetchAll();

?>

<!doctype html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>掲示板</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- 互換性重視CSS（color-mix等は未使用） -->

<style>

:root{

--bg:#ffffff; --fg:#111827; --sub:#6b7280; --line:#e5e7eb;

--accent:#2563eb; --accent-weak:#e8f0fe; --danger:#b91c1c;

--radius:12px; --space:16px; --space-sm:10px; --space-lg:22px;

--container:960px; --shadow:0 8px 24px rgba(0,0,0,.06);

}

@media (prefers-color-scheme: dark){

:root{

--bg:#0b0f14; --fg:#e5e7eb; --sub:#9ca3af; --line:#1f2937;

--accent:#60a5fa; --accent-weak:#0f2744; --danger:#fca5a5;

}

}

\*{box-sizing:border-box;}

html,body{height:100%;}

body{

margin:0; background:var(--bg); color:var(--fg); line-height:1.65;

font-family:ui-sans-serif,system-ui,-apple-system,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans JP","Hiragino Kaku Gothic ProN","Yu Gothic",Meiryo,sans-serif;

-webkit-text-size-adjust:100%;

}

body>\*{max-width:min(100vw - 24px, var(--container)); margin:0 auto;}

h1,h2{letter-spacing:.02em; margin:var(--space-lg) 0 var(--space);}

h1{font-size:clamp(20px,5vw,28px);} h2{font-size:clamp(18px,4.2vw,22px);}

.err{

background:#fde8e8; border:1px solid #f5b5b5; color:var(--danger);

padding:var(--space-sm) var(--space); border-radius:var(--radius);

}

form{

border:1px solid var(--line); background:#fafafa;

padding:var(--space); border-radius:var(--radius); box-shadow:var(--shadow);

margin:var(--space-lg) 0;

}

label{display:inline-block; font-size:.95rem; color:var(--sub); margin-bottom:6px;}

textarea{

width:100%; min-height:9rem; padding:12px; border:1px solid var(--line);

border-radius:10px; background:var(--bg); color:var(--fg); resize:vertical; outline:none;

}

textarea:focus{border-color:var(--accent); box-shadow:0 0 0 3px rgba(37,99,235,.2);}

input[type="file"]{

width:100%; display:block; padding:10px; border:1px dashed var(--line);

border-radius:8px; background:#f3f7ff;

}

input[type="file"]:focus{outline:2px solid var(--accent); outline-offset:2px;}

button[type="submit"]{

appearance:none; border:none; padding:12px 18px; border-radius:9999px;

background:var(--accent); color:#fff; font-weight:600; letter-spacing:.02em;

cursor:pointer; width:100%; margin-top:var(--space);

transition:transform .05s ease, opacity .2s ease;

}

button[type="submit"]:hover{opacity:.95;}

button[type="submit"]:active{transform:translateY(1px) scale(.995);}

.post{border-bottom:1px solid var(--line); padding:14px 0;}

.meta{color:var(--sub); font-size:.92rem; margin-bottom:6px; display:flex; gap:8px; flex-wrap:wrap;}

.content{font-size:1rem; word-wrap:break-word; overflow-wrap:anywhere;}

.content img{max-width:100%; height:auto; margin-top:8px; border-radius:10px; border:1px solid var(--line);}

button,input,textarea{touch-action:manipulation;}

::selection{background:var(--accent-weak);}

:target{scroll-margin-top:80px;}

@media (min-width:640px){

form{padding:20px 22px;}

button[type="submit"]{width:auto; padding-inline:20px;}

}

@media (min-width:960px){

.post{padding:18px 0;} .content{font-size:1.03rem;}

}

</style>

</head>

<body>

<h1>掲示板</h1>

<?php if (!empty($\_GET['err'])): ?>

<div class="err">

<?php

$map = [

'body' => '本文は1〜2000文字で入力してください。',

'size' => '画像は5MB以下にしてください。',

'mime' => '画像は jpg / png / gif / webp のみ対応です。',

'move' => '画像の保存に失敗しました。'

];

echo htmlspecialchars($map[$\_GET['err']] ?? 'エラーが発生しました。', ENT\_QUOTES, 'UTF-8');

?>

</div>

<?php endif; ?>

<!-- 自分自身へPOST -->

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

<label for="body">本文（1〜2000文字）</label>

<textarea id="body" name="body" maxlength="2000" required></textarea>

<div style="margin:.75rem 0;">

<label for="imageInput">画像（任意・5MBまで / jpg, png, gif, webp）</label><br>

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

</div>

<button type="submit">送信</button>

</form>

<h2>投稿一覧</h2>

<?php foreach ($entries as $entry): ?>

<div class="post">

<div class="meta">

#<?= (int)$entry['id'] ?> /

<?= htmlspecialchars($entry['created\_at'], ENT\_QUOTES, 'UTF-8') ?>

</div>

<div class="content">

<?= nl2br(htmlspecialchars($entry['body'], ENT\_QUOTES, 'UTF-8')) ?>

<?php if (!empty($entry['image\_filename'])): ?>

<img src="/image/<?= rawurlencode($entry['image\_filename']) ?>" alt="attached">

<?php endif; ?>

</div>

</div>

<?php endforeach; ?>

<script>

// クライアント側 5MB 警告（サーバ側チェックと二重化）

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

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

if (!imageInput) return;

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

if (imageInput.files.length < 1) return;

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

alert("5MB以下のファイルを選んでください。");

imageInput.value = "";

}

});

});

</script>

</body>

</html>

**Dockerfile作成**

vim Dockerfile

**中身**

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 echo -e "post\_max\_size = 5M\nupload\_max\_filesize = 5M" >> ${PHP\_INI\_DIR}/php.ini

**テーブル作成**

docker compose exec mysql mysql example\_db

**中身**

**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**

**);**