

Laravel AI Dev Workbench by Code Creator

User Guide

This guide explains the normal customer workflow for launching, accessing, and using the instance, including how to enable live AI responses with your own OpenAI API key.

Quick Start

1. Launch the EC2 instance from the AWS Marketplace AMI and assign a public IPv4 address.
2. Connect to your instance using your Amazon private key and the **ubuntu** user.
3. Open the Laravel landing page in your browser using your instance public IP address.
4. If you want live AI answers in Workbench Assistant or Open WebUI, configure your own OpenAI API key first.
5. Use the built in links to reach Open WebUI, Adminer, and Mailpit.

What You Need Before You Begin

- An AWS account so you can subscribe to the AMI and launch an EC2 instance.
- A public IPv4 address on the instance so you can open the web pages in your browser.
- Your Amazon private key pair for SSH access.
- An OpenAI API account with active API billing or credits only if you want live AI responses.
- No domain name is required for normal use.

Important: A normal ChatGPT subscription is not the same as OpenAI API billing. Workbench Assistant and Open WebUI need an OpenAI API key with active API usage if you want live AI responses.

Step 1: Connect to the Server

Connect to your instance using your Amazon private key and the ubuntu user.

Example command:

```
ssh -i /path/to/your-key.pem ubuntu@YOUR_PUBLIC_IP
```

After login, the server displays a banner with the main browser URLs and points you to the helpful file below:

```
/home/ubuntu/FIRST_LOGIN.txt
```

Step 2: Configure Live AI Responses with OpenAI

Do this first if you want Workbench Assistant and Open WebUI to return real AI answers.

You only need this step if you want live AI features. The Laravel landing page, Adminer, Mailpit, and the general server launch process do not require an OpenAI API key.

Run this command on the instance:

```
sudo /usr/local/bin/codecreator-laravel-ai-configure-openai.sh
```

When prompted:

- For the model, press Enter to accept the default, or type **openai/gpt-4.1-mini**.
- For the API key, paste your own OpenAI API key.

After the script finishes, try the assistant again in your browser.

If you see an error like “Application rate limited by AI provider [openai],” your OpenAI API account usually needs billing, credits, or higher usage limits. The AMI itself is working when you see that message.

Step 3: Open the Main Pages in Your Browser

Replace YOUR_PUBLIC_IP with your instance public IP address.

```
http://YOUR_PUBLIC_IP/  
http://YOUR_PUBLIC_IP/assistant  
http://YOUR_PUBLIC_IP:8080/  
http://YOUR_PUBLIC_IP:8081/  
http://YOUR_PUBLIC_IP:8025/
```

- Laravel landing page: main front page for the workbench.
- Workbench Assistant: built in Laravel AI assistant page.
- Open WebUI: browser based AI workspace.
- Adminer: database web interface for MySQL.
- Mailpit: safe email preview inbox for development.

Step 4: Normal Customer Workflow

6. Launch the instance and confirm it has a public IPv4 address.
7. SSH into the server as ubuntu and review /home/ubuntu/FIRST_LOGIN.txt.
8. Open the Laravel landing page in your browser.
9. If you want live AI responses, run the OpenAI configuration script.
10. Open Workbench Assistant and ask a question.
11. Use Adminer when you want to inspect the database.
12. Use Mailpit when you want to preview application emails during development.
13. Use Open WebUI if you want a browser based AI chat workspace connected to the same provider path.

Step 5: Adminer Database Login

The first boot process creates a fresh database password for the new instance. The current password is shown in this file:

```
/home/ubuntu/FIRST_LOGIN.txt
```

Use these values in Adminer:

- System: **MySQL**
- Server: **mysql**
- Username: **sail**
- Password: the generated password shown in **/home/ubuntu/FIRST_LOGIN.txt**
- Database: **laravel**

Step 6: Use the AI Features

Workbench Assistant is available inside the Laravel app at /assistant. Open WebUI is available on port 8080.

Both use the local LiteLLM gateway. To get live responses, your own OpenAI API key must be configured on the instance.

If you do not configure an OpenAI API key, the pages still load, but live AI answers will not work.

Helpful Commands

Use these commands when you want to check service status or restart the stack.

docker ps

sudo systemctl status codecreator-laravel-ai-stack.service --no-pager

sudo systemctl status codecreator-laravel-ai-firstboot.service --no-pager

cd /opt/codecreator-laravel-ai/app

docker compose -f compose.yaml up -d

docker compose -f compose.ai.yaml up -d --force-recreate

Troubleshooting

- Workbench Assistant returns 401: your OpenAI API key is missing, incorrect, revoked, or not configured correctly.
- Application rate limited by AI provider [openai]: your OpenAI API project is out of credits, has no billing, or hit provider usage limits.
- Open WebUI page opens but does not answer: configure the OpenAI API key with the setup script first.
- Adminer login fails: recheck the generated database password inside `/home/ubuntu/FIRST_LOGIN.txt`.
- A page does not open in the browser: confirm your EC2 security group allows the required port and that the instance has a public IPv4 address.

Required Security Group Ports

- TCP 22 for SSH
- TCP 80 for Laravel landing page and Workbench Assistant
- TCP 8080 for Open WebUI
- TCP 8081 for Adminer
- TCP 8025 for Mailpit

Accounts and Keys Summary

- AWS account: required
- Amazon private key pair: required
- Public IPv4 address: strongly recommended
- Domain name: not required
- OpenAI API account and API key: optional, but required for live AI answers
- Outbound internet access: required if you want live OpenAI responses

Support Paths

Main application path:

`/opt/codecreator-laravel-ai/app`

Helpful first boot note:

`/home/ubuntu/FIRST_LOGIN.txt`