

# AI Developer API Gateway by Code Creator

## Customer Launch and First Use Instructions

Also known as **Private AI Developer Gateway by Code Creator**

### 1. Product Overview

AI Developer API Gateway by Code Creator is a ready to launch self hosted AI API gateway for developers, teams, and businesses that want OpenAI compatible API access, provider routing, virtual keys, spend visibility, and developer integration tools from an AWS instance.

The server is built on Ubuntu 24.04 and includes LiteLLM, PostgreSQL, Redis, Docker, Docker Compose, and Nginx. The AMI generates fresh credentials on first boot and prepares a customer specific FIRST\_LOGIN file with the correct public IP address.

**Important:** This AMI does not include your OpenAI, Anthropic, Bedrock, Azure, or other provider API keys. Add your own provider keys after launch.

Item	Value
Operating system	Ubuntu 24.04 LTS
Default OS user	ubuntu
Web front door	Nginx on TCP 80
Gateway application	LiteLLM running internally on 127.0.0.1:4000
Database	PostgreSQL container
Cache	Redis container
Public API base	http://YOUR_PUBLIC_IP/v1
Admin UI	http://YOUR_PUBLIC_IP/ui/

### 2. AWS Security Group Requirements

Open only the ports required for SSH and web based access.

Item	Value
TCP 22	SSH access. Restrict to your trusted administrator IP address when possible.
TCP 80	Landing page, LiteLLM Admin UI, API docs, and OpenAI compatible API access through Nginx.

**Important:** Do not open TCP 4000, TCP 5432, or TCP 6379 to the public internet. LiteLLM is bound to localhost, and PostgreSQL and Redis are internal container services.

### 3. Connect to the Instance

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

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

**Note:** Replace /path/to/your-key.pem with your actual Amazon private key path and replace YOUR\_PUBLIC\_IP with the public IP address shown for your EC2 instance.

### 4. Read the First Login File

After connecting by SSH, read the first login instructions generated for your specific instance.

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

This file includes the current public IP based URLs, generated admin credentials, API key, helper commands, and security notes.

### 5. Main URLs

Item	Value
Landing page	http://YOUR_PUBLIC_IP/
LiteLLM Admin UI	http://YOUR_PUBLIC_IP/ui/
API docs	http://YOUR_PUBLIC_IP/docs
OpenAI compatible API base	http://YOUR_PUBLIC_IP/v1

**Note:** This product defaults to public IP based setup so a domain name is not required. You may add a domain and TLS later using your own preferred reverse proxy or certificate workflow.

### 6. Get Admin Credentials

Use the credential helper to display the generated admin username, password, and API key.

```
codecreator-ai-gateway-credentials
```

**Important:** Store these credentials securely. The password and API key are generated uniquely on first boot for each customer launch.

### 7. Log In to the Admin UI

- Open [http://YOUR\\_PUBLIC\\_IP/ui/](http://YOUR_PUBLIC_IP/ui/) in your browser.
- Use the username: **admin**
- Use the password shown by codecreator-ai-gateway-credentials or in [/home/ubuntu/FIRST\\_LOGIN.txt](/home/ubuntu/FIRST_LOGIN.txt).
- After login, review models, virtual keys, usage, and gateway settings from the LiteLLM Admin UI.

## 8. Helpful Management Commands

Item	Value
Show URLs	codecreator-ai-gateway-url
Show status	codecreator-ai-gateway-status
Show credentials	codecreator-ai-gateway-credentials
Show logs	codecreator-ai-gateway-logs
Restart services	codecreator-ai-gateway-restart
Show examples	codecreator-ai-gateway-examples
Add OpenAI API key	codecreator-ai-gateway-set-openai-key

Common commands:

**codecreator-ai-gateway-url**  
**codecreator-ai-gateway-status**  
**codecreator-ai-gateway-examples**  
**codecreator-ai-gateway-logs**  
**codecreator-ai-gateway-restart**

## 9. Add Your OpenAI API Key

The gateway is installed and running immediately, but model calls to OpenAI require your own OpenAI API key. To add the key, run:

**codecreator-ai-gateway-set-openai-key**

The command prompts for your API key, stores it in the gateway environment file, and restarts the LiteLLM container so the new key is loaded.

**Important:** Do not paste your provider API keys into public tickets, screenshots, or shared logs.

## 10. Test the OpenAI Compatible API

List configured models from inside the instance:

```
/opt/codecreator-ai-gateway/examples/curl-models.sh
```

After adding a valid provider key, test a chat completion request:

```
/opt/codecreator-ai-gateway/examples/curl-chat-openai-compatible.sh
```

The local API base from inside the instance is:

```
http://127.0.0.1:4000/v1
```

The public API base for applications is:

```
http://YOUR_PUBLIC_IP/v1
```

## 11. Connect an Application to the Gateway

Use the gateway as an OpenAI compatible API endpoint by changing the API base URL in your application.

```
# Example public API base
```

```
http://YOUR_PUBLIC_IP/v1
```

```
# Example Authorization header
```

```
Authorization: Bearer YOUR_GATEWAY_API_KEY
```

A Python example client is included on the instance:

```
python3 /opt/codecreator-ai-gateway/examples/python-openai-compatible-client.py
```

**Note:** The Python example requires a valid provider key before it can complete a paid provider backed model request.

## 12. Developer Examples

Developer examples are stored here:

```
/opt/codecreator-ai-gateway/examples
```

Read the local examples guide:

```
cat /opt/codecreator-ai-gateway/examples/README.txt
```

## 13. Basic Operations and Troubleshooting

Check the overall system status:

**codecreator-ai-gateway-status**

Check Docker containers directly:

**cd /opt/codecreator-ai-gateway && sudo docker compose ps**

Check LiteLLM readiness:

**curl -s http://127.0.0.1:4000/health/readiness | jq**

View LiteLLM logs:

**codecreator-ai-gateway-logs**

Restart the gateway stack and Nginx:

**codecreator-ai-gateway-restart**

## 14. Security Notes

- Restrict SSH access on TCP 22 to trusted administrator IP addresses when possible.
- Keep TCP 4000 closed to the public internet. Nginx is the intended public front door on TCP 80.
- Keep PostgreSQL and Redis internal. Do not expose TCP 5432 or TCP 6379.
- Store generated admin credentials and API keys securely.
- Rotate provider API keys according to your organization policy.
- Consider adding a domain name and HTTPS certificate for production use.

## 15. Recommended Instance Sizes and Storage

Item	Value
Small testing	t3.large
Recommended default	m6i.large
Heavier team usage	m6i.xlarge or larger
Root volume	40 GB gp3 minimum, 60 GB gp3 recommended

## 16. What Is Included

- Ubuntu 24.04 LTS base operating system.
- Docker and Docker Compose based deployment.
- LiteLLM gateway container pinned to the tested build image.
- PostgreSQL and Redis containers.
- Nginx reverse proxy and landing page.
- First boot automation that generates fresh customer credentials and public IP based instructions.
- Helper commands for URLs, credentials, status, logs, restart, examples, and OpenAI key setup.
- Developer examples for model listing, chat completions, and Python OpenAI compatible client usage.

## 17. Support Notes

When requesting support, include the output of these commands, but remove any passwords, API keys, tokens, or provider credentials before sharing:

**`codecreator-ai-gateway-status`**

**`codecreator-ai-gateway-logs 200`**

**`cd /opt/codecreator-ai-gateway && sudo docker compose ps`**