

Kong API Gateway and AI Gateway packaged by Code Creator

What this server does

This AMI launches a self-hosted Kong API Gateway server in your AWS account. It includes a ready-to-test demo route, a public landing page, Docker Compose management, PostgreSQL-backed Kong configuration, backup/export helpers, and an optional AI Gateway setup helper that lets you add your own AI provider key after launch.

Quick Start

1. Launch the AMI in AWS Marketplace or from your AWS account.
2. Open port 80 for the landing page and port 8000 for the Kong HTTP proxy. Open port 8443 if you want to test the HTTPS proxy.
3. SSH to the instance using the **ubuntu** user and your EC2 key pair.
4. Run `codecreator-kong-url` to show the correct public URLs for your instance.
5. Open the landing page in a browser, then test the demo route at `http://PUBLIC_IP:8000/demo`.
6. Use `codecreator-kong-ai-setup` only if you want to configure the optional AI Gateway route with your own provider API key.

Item	Value
Default SSH username	ubuntu
Landing page	<code>http://PUBLIC_IP/</code>
Working demo route	<code>http://PUBLIC_IP:8000/demo</code>
Kong HTTP proxy	<code>http://PUBLIC_IP:8000/</code>
Kong HTTPS proxy	<code>https://PUBLIC_IP:8443/</code>
Kong Admin API	Local-only by default: <code>http://127.0.0.1:8001/</code>
AI Gateway starter route after setup	<code>http://PUBLIC_IP:8000/ai/openai/chat</code>

How This Server Works

This product is designed to be easy to launch, easy to verify, and safer by default. The public internet sees the landing page and gateway proxy ports. The Kong Admin API is kept local to the instance so customers manage it through SSH or helper commands rather than exposing the admin interface publicly.

Component	Plain-English Role
Ubuntu 26.04 LTS	The operating system used by the AMI.
Docker and Docker Compose	Runs Kong and PostgreSQL as managed containers.

Kong Gateway 3.9.1 community edition	Receives API requests, matches routes, and proxies traffic to upstream services.
PostgreSQL 16 container	Stores Kong services, routes, plugins, and configuration.
Nginx	Serves the simple public landing page on port 80 and hosts the local demo upstream response.
First boot automation	Detects the new EC2 public IP and rewrites the landing page, first-login notes, and demo payload for each customer instance.
AI Gateway helper	Optional setup command that creates an OpenAI-compatible AI proxy route after the customer enters their own provider API key.

Important behavior

If `http://PUBLIC_IP:8000/` or `https://PUBLIC_IP:8443/` says "no Route matched with those values," that is normal. Kong is running, but no default root route exists. The included proof route is `/demo`.

First Login and URL Discovery

Connect with SSH using the EC2 key pair selected at launch. The username is `ubuntu`.

```
ssh -i your-key.pem ubuntu@PUBLIC_IP
```

After login, run:

```
codecreator-kong-url
cat /home/ubuntu/FIRST_LOGIN.txt
codecreator-kong-status
```

The URL helper displays the live public IP for your instance, the demo route, gateway proxy URLs, AI Gateway starter route, local Admin API information, and the most useful helper commands.

Browser Tests

Use these tests after the instance has had a few minutes to complete first boot. Replace `PUBLIC_IP` with the public IPv4 address of your EC2 instance.

URL	Expected Result
<code>http://PUBLIC_IP/</code>	Code Creator landing page opens.
<code>http://PUBLIC_IP:8000/demo</code>	JSON response with status ok and product name.
<code>http://PUBLIC_IP:8000/</code>	Kong message: no Route matched with those values. This is normal.
<code>https://PUBLIC_IP:8443/</code>	Browser certificate warning may appear. After accepting, Kong may show no Route matched. This is normal.
<code>http://PUBLIC_IP:8001/</code>	Should not work publicly. Kong Admin API is local-only by default.

Command-Line Tests

From the instance, run:

```
codecreator-kong-status  
codecreator-kong-demo-test
```

Expected signs of a healthy system:

- Docker is active.
- Nginx is active.
- codecreator-kong.service is enabled and active.
- codecreator-kong and codecreator-kong-database containers are healthy.
- Demo route reports ok.
- Kong version is 3.9.1.
- Kong edition is community.
- AI route says not yet configured unless the customer has run the AI setup helper.

Recommended AWS Security Group Rules

Port	Protocol	Recommended Source	Purpose
22	TCP	Your IP only	SSH administration.
80	TCP	0.0.0.0/0 or trusted users	Public landing page.
8000	TCP	0.0.0.0/0 or trusted clients	Kong HTTP proxy/API traffic.
8443	TCP	0.0.0.0/0 or trusted clients	Kong HTTPS proxy/API traffic.
8001	TCP	Do not open publicly	Kong Admin API is local-only on the instance.
8444	TCP	Do not open publicly	Kong HTTPS Admin API is local-only on the instance.

Security default

The Kong Admin API is intentionally bound to localhost on the EC2 host. Manage Kong from SSH using helper commands or use an SSH tunnel. Do not expose the Admin API directly to the public internet.

Managing Kong Safely

The Kong Admin API is available locally on the instance:

```
http://127.0.0.1:8001/  
https://127.0.0.1:8444/
```

To reach the Admin API from your workstation without exposing it publicly, use an SSH tunnel:

```
ssh -i your-key.pem -L 8001:127.0.0.1:8001 ubuntu@PUBLIC_IP
```

Then open this URL on your local computer:

```
http://127.0.0.1:8001/
```

Adding Your Own API Route

Kong routes requests by matching a path and forwarding the request to an upstream service. For example, the included /demo route forwards to a small local Nginx-hosted JSON file. You can add your own service and route using the local Admin API.

Before adding production routes

Plan your upstream URL, authentication plugin, rate limiting, and logging requirements. For production use, avoid placing sensitive APIs behind unauthenticated public routes.

Example: add a route named my-api that forwards /my-api traffic to an upstream URL:

```
curl -X PUT http://127.0.0.1:8001/services/my-api \
  --data "url=https://httpbin.org/anything"

curl -X PUT http://127.0.0.1:8001/services/my-api/routes/my-api-route \
  --data "paths[]={/my-api}" \
  --data "strip_path=true"

curl http://PUBLIC_IP:8000/my-api
```

Optional AI Gateway Setup

This AMI includes helper commands for creating an AI Gateway starter route with Kong's bundled AI Proxy plugin. No AI provider key is included in the AMI. The customer supplies their own key after launch.

Command	Purpose
codecreator-kong-ai-setup	Prompts for your OpenAI API key and model, then creates /ai/openai/chat.
codecreator-kong-ai-test	Sends a small OpenAI-style chat request through Kong.
codecreator-kong-ai-remove	Removes the AI route and AI Proxy plugin from Kong.

To configure the AI route:

```
codecreator-kong-ai-setup
```

After setup, the route is:

```
http://PUBLIC_IP:8000/ai/openai/chat
```

To test it:

```
codecreator-kong-ai-test
```

AI provider cost warning

Requests sent through the AI Gateway route may create charges with your AI provider. The customer is responsible for their own provider key, model choice, usage, and billing.

Helper Commands Reference

Command	What It Does
<code>codecreator-kong-url</code>	Shows the current public URLs, local Admin API information, and helper commands.
<code>codecreator-kong-status</code>	Shows OS, Docker, Nginx, Kong containers, demo route, AI plugin availability, version, and edition.
<code>codecreator-kong-demo-test</code>	Tests the local upstream, local Kong proxy, and public /demo route.
<code>codecreator-kong-ai-setup</code>	Creates the optional AI Gateway route after you enter your own AI provider key.
<code>codecreator-kong-ai-test</code>	Tests the optional AI Gateway route.
<code>codecreator-kong-ai-remove</code>	Removes the optional AI Gateway route and plugin.
<code>codecreator-kong-restart</code>	Restarts the Kong Docker Compose stack and waits for Kong to become ready.
<code>codecreator-kong-logs</code>	Shows logs for the stack. Can also show kong, database, nginx, or firstboot logs.
<code>codecreator-kong-backup</code>	Creates a backup archive with the Kong database dump and local product files.
<code>codecreator-kong-list-backups</code>	Lists available backup and export files.
<code>codecreator-kong-export-config</code>	Exports Kong services, routes, plugins, and consumers to JSON.
<code>codecreator-kong-firstboot-status</code>	Shows first boot status, marker, recent first boot log, and main status.

Backup and Export

Use the built-in backup helper before making major changes to services, routes, plugins, or AI Gateway configuration.

```
codecreator-kong-backup
codecreator-kong-list-backups
```

Backups are stored here:

```
/opt/codecreator-kong/backups
```

To export only Kong Admin API configuration as JSON:

```
codecreator-kong-export-config
```

Troubleshooting

Problem	What It Usually Means	What To Do
Landing page does not load immediately	First boot may still be running.	Wait 1-3 minutes and refresh. Then run <code>codecreator-kong-firstboot-status</code> .
/demo does not work	Kong may not be ready or port 8000 may not be open.	Run <code>codecreator-kong-status</code> and check the AWS Security Group for port 8000.
<code>http://PUBLIC_IP:8000/</code> says no Route matched	Normal Kong behavior. No root route is configured.	Use /demo or create your own route.

Public Admin API does not open	This is intentional.	Use SSH and local helper commands, or create an SSH tunnel.
AI test fails	AI route not configured, invalid API key, provider error, or network issue.	Run codecreator-kong-ai-setup again, verify provider key and model, then run codecreator-kong-ai-test.
Containers unhealthy	Kong or PostgreSQL may still be starting or may need restart.	Run codecreator-kong-restart and check codecreator-kong-logs.

System Updates and Maintenance

For routine Ubuntu package updates, use:

```
sudo apt-get update  
sudo DEBIAN_FRONTEND=noninteractive apt-get -y upgrade
```

If a reboot is required, reboot during a maintenance window:

```
sudo reboot
```

After reconnecting, verify the gateway:

```
codecreator-kong-status  
codecreator-kong-demo-test
```

Technical Summary

Category	Value
Operating system	Ubuntu 26.04 LTS
Gateway	Kong 3.9.1 community edition
Database	PostgreSQL 16 Alpine container
Runtime	Docker and Docker Compose
Landing page	Nginx on port 80
Public gateway ports	8000 HTTP, 8443 HTTPS
Admin API ports	127.0.0.1:8001 and 127.0.0.1:8444 only
Product directory	/opt/codecreator-kong
First login notes	/home/ubuntu/FIRST_LOGIN.txt
First boot log	/var/log/codecreator-kong-firstboot.log

Appendix: Suggested Product Positioning

Short customer-facing summary:

Suggested summary

Launch a self-hosted Kong API Gateway and AI Gateway starter server in your AWS account with Docker Compose, a working demo route, first-boot public IP automation, backup/export helpers, and localhost-only Admin API security by default.

Three key highlights:

- Self-hosted Kong Gateway on Ubuntu 26.04 LTS with Docker Compose and PostgreSQL.
- Ready-to-test public demo route plus helper commands for status, logs, backup, export, restart, and AI setup.
- Security-focused default design with public proxy ports and localhost-only Kong Admin API.