



Remoder.com

# **Al Agent Deployment — Project 2**

- The first version of this document is available on my posts, please check it out to understand what changed.
- This version adds Security Enhancements & is part of ensuring the path of <u>Building Responsible Al</u>
   <u>Workloads</u>

# The Challenge

The Problem We Solved

How do you take a powerful AI model and make it accessible, secure, and easy to deploy anywhere?

Traditional setups are:

- Slow to deploy
- 🕨 Insecure for production use 🔓
- Difficult to manage and scale

#### Our Solution



We built a secure, self-contained AI API using:

Docker: To package our application and all its dependencies into a single, portable container.

ollama: A powerful, easy-to-use LLM server to run models like Mistral and LLaMA3.

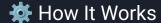
Nginx: An industry-standard reverse proxy to handle traffic and add a crucial layer of security.

#### Features

#### Multi-Model AI Agent powered by:

- / \*\*API Key Authentication\*\* Protects your endpoint from unauthorized access.
- \*\*Runs as Non-Root User\*\* Enhances security by running all processes with limited privileges.
- #\*Dynamic Configuration via Environment Variables\*\* Change models, API keys, and SSL details at runtime without rebuilding.
- Pull Portability: Runs flawlessly on any cloud platform
- Scalability: The foundation for scaling out to handle heavy traffic loads.
- V \*\*Self-Signed SSL Certificate\*\* Encrypts traffic between the client and the server.
- \*\*Container-Native Logging\*\* Nginx logs are properly redirected to `stdout` and `stderr` for easy access with `docker logs`.
- ▼ \*\*Health Check Endpoint\*\* A simple `/health` endpoint to verify service status.
- V \*\*Ollama Al Model Server\*\* Run powerful Al models like Mistral, LLaMA3, and more.
- 🔽 \*\*Dockerized for Portability\*\* Works anywhere with Docker, perfect for cloud platforms like Runpod.

### The Architecture



A simple, powerful flow:

- 1. Your Client sends a secure request.
- 2. Nginx validates the request and API key.
- 3. Nginx proxies the request to the Ollama container.
- 4. Ollama processes the request using the Al model.
- 5. The Response is sent back, securely.

Visually, imagine a simple diagram: Client 👉 Nginx Reverse Proxy 👉 Ollama Server 👉 Al Model

### The Tech Stack

#### Our Toolkit

- Docker: The containerization powerhouse.
- Ollama: Our choice for a lightweight, versatile LLM server.
- Nginx: For robust and secure reverse proxying.
- Ubuntu 22.04: A stable and efficient base image for our container.
- Bash Scripting: To orchestrate the startup process and pre-load models.

# A Simple Path to Al Mastery

Al isn't complicated. It's a journey of a thousand small steps.

The world of AI can seem overwhelming, but the most important lesson is to simplify. Focus on the core principles and build your skills one piece at a time. This project is a perfect example of that philosophy in action—we took a big problem and solved it with small, manageable components.

Here's how you can make AI easy:

- Start with the Basics: Learn fundamental concepts like models, APIs, and data. Don't try to understand everything at once.
- — Mail Small Projects: Apply your knowledge to a single, concrete task. A simple API or a small script is a huge victory!
- Connect the Dots: See how each new piece of knowledge fits with what you already know.
- Celebrate Progress: Acknowledge every small step forward. Each line of code and every successful run is a win.

## QUESTIONS???

Contact us & come learn with us  $9(9_{\circ})^{\circ}$ 

- https://www.linkedin.com/company/remoder
- https://www.linkedin.com/in/sanjars/
- https://www.youtube.com/@remoder-inc
- remoder.com
- Full walkthrough of this project is available on our "Master Al Deployment " course