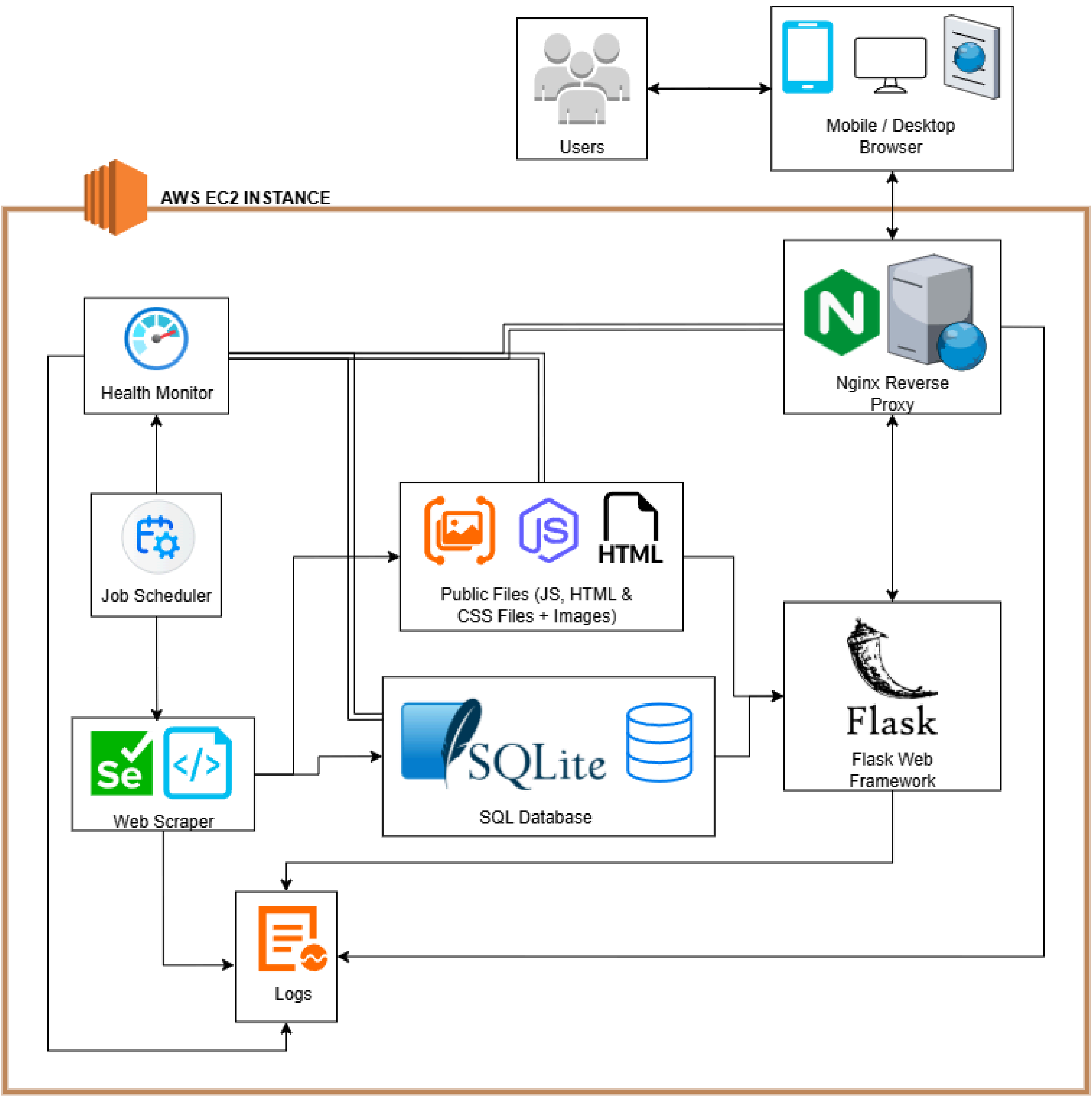
PopMart Scraper - System Architecture Overview



Executive Summary

The PopMart Scraper is a complete web application that automatically collects data about PopMart blind box collections and presents it through a user-friendly website. The system runs entirely on a single AWS server and operates 24/7 with minimal human intervention.

System Overview

The application consists of three main components that work together:

1. Data Collection System - Automatically scrapes product information from the web
2. Database System - Stores and organizes all the collected data
3. Web Server - Presents the data to users through a website

---

1. Data Collection System (Web Scraper)

What It Does

The scraper automatically visits PopMart's website and collects information about different blind box series and their individual products. Think of it like a digital assistant that visits websites and copies down product information.

How It Works

- Series Discovery: The scraper first finds all available PopMart series by scanning the main website
- Image Collection: For each series, it downloads product images and stores them locally
- Data Processing: It analyzes the images and text to identify which products are regular vs. secret/rare items
- Automated Scheduling: The scraper runs automatically every day at 2:00 AM to check for new products

Technical Details

- Tools Used: Python with Selenium (automated web browser) and BeautifulSoup (web page parsing)
- Storage: Images are saved to the server's file system in organized folders

- Data Output: Product information is stored in a structured database

---

2. Database System

What It Does

The database acts as the central storage system, organizing all the collected data in a structured way that makes it easy to search and retrieve information.

How It Works

- Series Table: Stores information about each PopMart series (name, ID, image URL, etc.)
- Products Table: Stores details about individual products within each series
- Relationships: Links products to their parent series
- Status Tracking: Keeps track of which series are currently active vs. discontinued

Technical Details

- Database Type: SQLite (lightweight, file-based database)
- Location: Stored as a single file on the server
- Size: Currently about 0.24 MB (very small and efficient)
- Backup: Database is automatically backed up daily

---

3. Web Server System

What It Does

The web server presents all the collected data to users through a website that anyone can visit. It provides both a user interface and an API for programmatic access.

How It Works

- User Interface: Serves HTML pages that display the PopMart collections
- API Endpoints: Provides data access for the website's JavaScript functionality

- Static File Serving: Delivers images and other assets to users
- Search Functionality: Allows users to search through series and products

Technical Details

- Framework: Flask (Python web framework)
- Server: Gunicorn (production web server)
- Reverse Proxy: Nginx (handles incoming requests and serves static files)
- SSL/HTTPS: Secure connections using Let's Encrypt certificates

---

4. AWS Hosting Infrastructure

Server Setup

- Instance Type: t3.micro (small, cost-effective server)
- Operating System: Ubuntu Linux
- Location: AWS US East (Ohio) region
- Cost: Free for first year, then ~$4-5/month

How Everything Runs Together

1. 24/7 Operation: All components run continuously as background services
2. Automatic Restart: If any component crashes, it automatically restarts
3. Logging: All activities are logged for monitoring and debugging
4. Health Monitoring: Built-in health checks ensure everything is working

Security & Access

- HTTPS Encryption: All web traffic is encrypted
- Domain Name: Custom domain (radahnmiku.dev) with subdomain support
- Firewall: AWS security groups control network access
- SSH Access: Secure remote access for maintenance

Request Flow

1. User visits website → Request goes to AWS
2. Nginx receives request → Processed by Flask application

   a. File requests (Image, JS Scripts, HTML & CSS Files) → Retrieve requested information
   b. Database queries → Retrieve requested information through DB Handler
3. Response → Sent back to user's browser through Nginx

---

5. Cost Structure

Current Costs

- AWS EC2: $0/month (free tier) → $4-5/month after first year
- Domain Name: ~$12/year
- SSL Certificate: Free (Let's Encrypt)
- Total Monthly Cost: ~$5-6/month

Cost Optimization

- Efficient Resource Usage: Minimal server resources required
- Automated Operations: Reduces manual maintenance costs
- Free Tier Utilization: Maximizes AWS free tier benefits

Technical Architecture Summary

All components run on a single AWS EC2 instance, making it simple to deploy, maintain, and scale. The architecture is designed for reliability, with automatic restarts, comprehensive logging, and built-in error handling.This design allows the system to operate autonomously while providing a professional, fast, and reliable service to users interested in PopMart blind box collections.