

Cloud Management System (CMS) - User Manual

1. Overview

The Cloud Management System (CMS) is a hybrid cloud orchestration tool designed to run on WSL (Windows Subsystem for Linux). It serves as a unified dashboard for administrators to manage:

- IaaS (Infrastructure as a Service): Virtual Machines using QEMU/KVM.
- CaaS (Container as a Service): Docker containers and images using the Docker Engine.

The system offers two modes of operation: a lightweight Command Line Interface (CLI) and a modern Graphical User Interface (GUI).

2. Prerequisites & Setup

Before running the application, ensure the following are installed and active on your system:

1. **Operating System:** Windows 10/11 with WSL 2 (Ubuntu 20.04 or later).
2. **Container Engine:** Docker Desktop for Windows (ensure "WSL 2 Integration" is enabled in settings).
3. **Python Environment:** Python 3.8+ with the following libraries:
 - docker
 - customtkinter
 - packaging
4. **Virtualization Tools:** QEMU and KVM installed inside Ubuntu:
`sudo apt update && sudo apt install qemu-kvm`

Part 1: Using the Command Line Interface (CLI)

The CLI version provides a fast, text-based method for managing resources without graphical overhead.

How to Launch

1. Open your Ubuntu terminal.
2. Navigate to the project folder.
3. Run the following command:
4. `python3 main.py`
5. Menu Navigation

The system displays a numbered menu. Enter the corresponding number to select an option:

- Option 1 (VM Operations): Access creation and management of Virtual Machines.
- Option 2 (Docker Operations): Access container and image management tools.
- Option 0 (Exit): Close the application.

Part 2: Using the Graphical User Interface (GUI)

The GUI version provides a professional dashboard experience with real-time system logs, visual file browsers, and advanced project management tools.

How to Launch

1. Open your Ubuntu terminal.
2. Run the following command:
3. `python3 gui_main.py`
- 4.

Section A: VM Workstation

Located on the left sidebar, click "VM Workstation".

1. Manual Launch Use this for quick, custom VM creation.

- **VM Name:** Enter a unique name for the virtual machine.
- **RAM (MB):** Allocate memory (e.g., 1024 for 1GB).

- **CPU Cores:** Assign processor cores (e.g., 2).
- **Disk Size (GB):** Define the virtual hard drive size.
- **ISO Image:** (Optional) Click "Browse" to select an OS installer file (.iso).
- **Action:** Click the green "CREATE & LAUNCH" button to boot the VM.

2. Automated Launch (Config Mode) Use this for reproducible infrastructure.

- **Action:** Click the orange "LAUNCH FROM CONFIG FILE" button.
- **Behavior:** The system reads the **vm_config.json** file in the project directory and automatically boots a VM with the pre-defined settings.

Section B: Docker Operations

Located on the left sidebar, click "Docker Operations". This section contains four specialized tabs.

Tab 1: Manage

Monitor and control active resources.

- **List All Images:** Displays a detailed table of all local Docker images, including Repository, Tag, ID, and Size.
- **List Running Containers:** Shows active containers and their status.
- **Run New Container:** Launch an instance of an image directly from the dashboard.
 - **Image Name:** Enter the repository name (e.g., nginx or alpine).
 - **Container Name:** (Optional) Assign a custom name (e.g., web-server).
 - **Action:** Click the green "RUN" button to start the container in the background.
- **Stop Container:** Enter a Container ID (e.g., a1b2c3d4) and click the red "STOP CONTAINER" button to halt the process immediately.

Tab 2: Create File

A built-in editor for defining Infrastructure as Code.

1. **Project Name:** Enter a name (e.g., MyWeb). A dedicated folder will be created inside Docker_Projects.
2. **Editor:** Type or paste your Dockerfile instructions (e.g., FROM python:3.9).
3. **Action:** Click "SAVE DOCKERFILE". The system saves the file automatically to the correct path.

Tab 3: Build Image

Compiles your Dockerfile into a runnable image.

1. **Image Tag:** Define the name and version (e.g., myweb:v1).
2. **Directory Path:** Enter the path to your project (e.g., Docker_Projects/MyWeb).
3. **Action:** Click "BUILD IMAGE". The system logs will display the build progress in real-time.

Tab 4: Search

Find resources locally or on the web.

- **Search Docker Hub:** Enter a keyword (e.g., nginx) and click the Blue Button to search the global Docker registry.
- **Search Local Images:** Enter a keyword (e.g., myweb) and click the Purple Button to filter your local library.
- **Pull Image:** Enter a specific image name (e.g., ubuntu:latest) to download it to your machine.

3. Troubleshooting Guide

Error Message / Issue	Solution
"Cannot connect to the Docker daemon"	Ensure Docker Desktop is open in Windows and the "WSL 2 Integration" is turned on in Docker settings.
"Permission Denied" (deleting folders)	Windows cannot modify files created by Linux root. Use (Sudo rm -rf <filename>) inside the Ubuntu terminal.
VM Window Freezes or Hangs	The VM may have too much RAM allocated for your system. Close the window and try reducing RAM to 1024MB or 512MB.
GUI Buttons not appearing	Ensure the application window is maximized. The layout is responsive but requires sufficient screen space.