**User Manual for Cloud Management System**

**Overview**

The Cloud Management System (CMS) is a Python-based application with a graphical user interface (GUI) designed for managing virtual machines and Docker containers. The system allows users to create, configure, and manage virtual machines (VMs) using QEMU, as well as manage Docker resources, such as images and containers.

**System Requirements**

**Software Requirements:**

1. Python 3.x
2. Required Python Libraries:
   * tkinter
   * docker
   * os
   * subprocess
   * json
   * datetime
   * logging
   * time
3. External Dependencies:
   * QEMU binaries
   * Docker installed and running

**Hardware Requirements:**

* A system capable of running Python and Docker (Linux, macOS, or Windows with WSL2).

**Installation Guide**

1. Install Python 3.x from [Python.org](https://www.python.org/).
2. Install the required Python libraries:
3. pip install docker
4. Ensure QEMU is installed and added to the system PATH.
5. Install Docker and ensure it is running.
6. Download the cloud\_management\_system.py script and execute it:
7. python cloud\_management\_system.py

**Features and Instructions**

**Main Menu**

After launching the application, the main menu will appear, providing buttons to access various features:

1. **Create VM**
2. **Create Dockerfile**
3. **Build Docker Image**
4. **List Docker Images**
5. **List Running Containers**
6. **Stop a Container**
7. **Search for Docker Image**
8. **Search DockerHub**
9. **Download Docker Image**
10. **Run a New Container**

**1. Create VM**

**Purpose:**

Create and configure virtual machines using QEMU.

**Instructions:**

1. Click **Create VM** in the main menu.
2. A new window will appear with two options:
   * **Use Existing Disk Image**: Browse for an existing QCOW2 image.
   * **Create New Disk Image**: Specify CPU, memory, and disk size, and browse for a boot ISO.
3. For new image creation, a dialog will open prompting the user to select a save location for the new disk image.
4. Fill in the required fields and click **Create VM** to start the VM.

**2. Create Dockerfile**

**Purpose:**

Create and save Dockerfiles.

**Instructions:**

1. Click **Create Dockerfile** in the main menu.
2. A text editor window will appear to write Dockerfile content.
3. Click **Save** to save the file in the desired directory.

**3. Build Docker Image**

**Purpose:**

Build Docker images from Dockerfiles.

**Instructions:**

1. Click **Build Docker Image** in the main menu.
2. Select the directory containing the Dockerfile.
3. Provide an image name and tag.
4. Click **Build Image** to start the build process.

**4. List Docker Images**

**Purpose:**

View all locally available Docker images.

**Instructions:**

1. Click **List Docker Images** in the main menu.
2. A list of image names and tags will appear.

**5. List Running Containers**

**Purpose:**

Display all currently running Docker containers.

**Instructions:**

1. Click **List Running Containers** in the main menu.
2. A list of container IDs and names will appear.

**6. Stop a Container**

**Purpose:**

Stop a selected running container.

**Instructions:**

1. Click **Stop a Container** in the main menu.
2. Select the container from the displayed list.
3. Click **Stop** to terminate the container.

**7. Search for Docker Image**

**Purpose:**

Search for Docker images locally.

**Instructions:**

1. Click **Search for Docker Image** in the main menu.
2. Enter the name or partial name of the image.
3. Matching images will be displayed.

**8. Search DockerHub**

**Purpose:**

Search for Docker images on DockerHub.

**Instructions:**

1. Click **Search DockerHub** in the main menu.
2. Enter the name of the image.
3. Results from DockerHub will be displayed.

**9. Download Docker Image**

**Purpose:**

Download images from DockerHub.

**Instructions:**

1. Click **Download Docker Image** in the main menu.
2. Enter the image name and tag.
3. The image will be pulled to your local system.

**10. Run a New Container**

**Purpose:**

Run a Docker container from an image.

**Instructions:**

1. Click **Run a New Container** in the main menu.
2. Select the image from the list of available images.
3. Specify optional configurations (e.g., port mapping, environment variables).
4. Click **Run Container** to start the container.

**Error Handling and Logs**

The application uses logging to record errors and user actions. If an error occurs, check the log file (cloud management system.log) in the application directory for details.

**Troubleshooting**

1. **Dependencies Missing**:
   * Ensure all prerequisites are installed.
2. **Docker Not Running**:
   * Verify that the Docker service is running.
3. **Invalid Inputs**:
   * Double-check entered data (e.g., numeric fields for CPU or memory).