

## User Manual

### Overview

This manual describes how to use the provided Python scripts to manage Docker operations, search and pull Docker images, and create virtual machines interactively. The tools provide a GUI for ease of use, built using tkinter.

---

### 1. Docker Management App (app.py)

#### Features:

1. **Create a Dockerfile**
  - Generates a Dockerfile for either Python or Java applications.
  - Prompts the user for the desired programming language and saves the file to a specified directory.
2. **Build a Docker Image**
  - Builds a Docker image from a specified directory containing a Dockerfile.
  - Allows naming the image and specifying a tag.
3. **Create a Container**
  - Creates and starts a container from an existing Docker image.
  - Prompts for the image name and container name.
4. **List Docker Images**
  - Displays all Docker images available on the local system.
5. **List Running Containers**
  - Lists all active containers and provides an option to stop them.
6. **Stop a Container**
  - Stops a container by specifying its ID.

#### How to Use:

1. Run the script: `python app.py`.
  2. Interact with the GUI to perform Docker-related tasks.
- 

### 2. Docker Image Search App (images.py)

#### Features:

### 1. Search DockerHub

- Searches DockerHub for images matching the entered name.
- Displays a list of repositories found and provides an option to pull them.

### 2. Search Local Docker Images

- Searches for Docker images stored locally.

### 3. Pull Docker Images

- Downloads Docker images from DockerHub.

#### How to Use:

1. Run the script: `python images.py`.
  2. Enter the image name in the input field.
  3. Choose one of the options:
    - Search DockerHub
    - Search Local Storage
    - Pull Image
  4. Follow the on-screen instructions to proceed.
- 

## 3. Virtual Machine Management App (VM.py)

#### Features:

### 1. Create Virtual Machines

- Provides a GUI to enter details for a VM (name, memory size, disk size) Memory: 2 GB of RAM: Enter 2048 (in MB) / 4 GB of RAM: Enter 4096 (in MB) / 6 GB of RAM: Enter 6144 (in MB)
- Disk Size: Enter 10240 for 10 GB of disk space / 20480 for 20 GB of disk space / 30720 for 30 GB of disk space.
- Creates a virtual disk and launches the VM using QEMU.
- Requires the path to QEMU binaries and an ISO file.

### 2. Interactive GUI

- Simplified form for inputting VM specifications.

#### Prerequisites:

- Install QEMU.

- Specify the paths to qemu-system-x86\_64.exe and qemu-img.exe in the script.

### How to Use:

1. Run the script: python VM.py.
2. Enter the required VM details in the form.
3. Ensure the specified ISO file exists.
4. Submit the details to create and boot the VM.

---

### Technical Notes

- **Python Version:** Ensure Python 3.9+ is installed.
- **Dependencies:** Install required libraries (tkinter, docker, requests) using:

bash

Copy code

```
pip install docker requests
```

---

For any errors or issues encountered:

1. Ensure Docker and QEMU are correctly installed and configured.
2. Verify network connectivity for DockerHub operations.
3. Consult the error messages displayed in the GUI or the terminal.