Prerequisites

- **Docker Desktop**: Installed on your computer with virtualization enabled in your BIOS. Ensure the "Windows Subsystem for Linux" feature is enabled if using Windows.
- **Trainer Resources**: Access to the folder containing `Containerfile`, `README.md`, `build.sh`, and `test_runner_script.py`

Step-by-Step Instructions

Step 1: Set Up the Project Folder Structure

1. **Create the Parent Folder**:
   - Open a terminal and create a directory named `ms-novel-code`:

```
mkdir ~/ms-novel-code
cd ~/ms-novel-code
```

2. **Create Subfolders and Copy Files**:
   - Inside `ms-novel-code`, create a `sandbox` folder and copy the `Containerfile`, `README.md`, and `build.sh` from the Trainer Resources folder:

```
mkdir sandbox
cp /path/to/Trainer/Resources/Containerfile ~/ms-novel-code/sandbox/
cp /path/to/Trainer/Resources/README.md ~/ms-novel-code/sandbox/
cp /path/to/Trainer/Resources/build.sh ~/ms-novel-code/sandbox/
```

   - Create a `host_tasks` folder and copy `test_runner_script.py`:

```
mkdir host_tasks
cp /path/to/Trainer/Resources/test_runner_script.py ~/ms-novel-code/
   host_tasks/
```

   - Ensure `task-1` and `task-2` folders (with `main.py` and `tests.py`) are inside `host_tasks`. For example:

```
mkdir ~/ms-novel-code/host_tasks/task-1
cp /path/to/Trainer/Resources/task-1/main.py ~/ms-novel-code/host_
   tasks/task-1/
cp /path/to/Trainer/Resources/task-1/tests.py ~/ms-novel-code/host_
   tasks/task-1/
```

Step 2: Build the Docker Image

1. **Navigate to the Sandbox Directory**:
   - Change to the `sandbox` directory:

```
cd ~/ms-novel-code/sandbox
```

2. **Build the Docker Image**:
   - Build the image using the `Containerfile`:

```
docker build --progress=plain -t ms-novel-code-sandbox:latest -f
    Containerfile .
```

   - This process downloads the base `alpine` image, installs dependencies (e.g., Python 3.12, `uv`), and sets up `/tasks`. It may take several minutes. Wait for the process to complete.
   - Verify the image:

```
docker images
```

   - You should see `ms-novel-code-sandbox` with tag `latest` and a size around 531MB.

Step 3: Run the Container

1. **Stop and Remove Existing Containers**:
   - Remove any existing containers with conflicting names:

```
docker stop ms-novel-code-sandbox
docker rm ms-novel-code-sandbox
```

2. **Run a New Container**:
   - Start a new container with the `host_tasks` volume mounted to `/tasks`:

```
docker run -d --name ms-novel-code-sandbox -v "/Users/dev/Desktop/ms-
    novel-code/host_tasks:/tasks" --rm ms-novel-code-sandbox:latest
```

   - The `--rm` flag ensures the container is removed when stopped, providing a fresh environment. Note the container ID (e.g., `40b816cd3b44d5f55074e158a9f76253c60a67225d`

Step 4: Access the Container and Run Tests

1. **Open a Shell Inside the Container**:
   - Access the container:

```
docker exec -it ms-novel-code-sandbox bash
```

2. **Navigate to the Task Directory**:
   - Change to the `task-1` directory:

```
cd /tasks/task-1
```

3. **Run the Test Runner Script**:
   - Execute the tests:

```
python3 ../test_runner_script.py --task-dir /tasks/task-1
```

   - Review the output for pass/fail results. If errors occur (e.g., `RecursionError`), edit `main.py` in `/Users/dev/Desktop/ms-novel-code/host_tasks/task-1/` on the host to add input validation (e.g., raise `ValueError` for negative numbers or non-integers) and rerun.

Step 5: Install Additional Packages (e.g., `pandas`)

1. **Create a Virtual Environment**:
   - Set up a virtual environment in the task folder:

```
uv venv
```

   - This creates a `.venv` directory in `/tasks/task-1/`.

2. **Install `pandas`**:
   - Install the `pandas` package:

```
uv pip install pandas
```

   - This installs `pandas` and its dependencies (e.g., `numpy`, `python-dateutil`).

3. **Save Dependencies**:
   - Generate a `requirements.txt` file:

```
uv pip freeze > requirements.txt
```

4. **Activate the Virtual Environment and Verify**:

   - Activate the virtual environment:

   ```
   source .venv/bin/activate
   ```

   - Check the `pandas` version:

   ```
   python3 -c "import␣pandas;␣print(pandas.__version__)"
   ```

   - Expected output: A version like `2.3.0`.
   - Alternatively, use `uv run` without activation:

   ```
   uv run python3 -c "import␣pandas;␣print(pandas.__version__)"
   ```

5. **Troubleshoot**:

   - If `ModuleNotFoundError` occurs, ensure the virtual environment is activated or use `uv run`. Reinstall if needed:

   ```
   uv venv
   uv pip install pandas
   ```

Step 6: Maintain and Troubleshoot

- **Edit Code**: Modify `main.py` or `tests.py` in `/Users/dev/Desktop/ms-novel-code/host_tasks/` on the host. Changes are reflected in `/tasks/task-1/` due to the volume mount.
- **Switch Test Runner**: If `test_runner_script_ng.py` is required (per guidelines but not present), copy it from Trainer Resources:

```
cp /path/to/Trainer/Resources/test_runner_script_ng.py ~/ms-novel-code/
   host_tasks/
```

  Then run:

```
python3 ../test_runner_script_ng.py --task-dir /tasks/task-1
```

- **Stop the Container**: It auto-removes due to `--rm` when you exit the shell (`exit`).