Simple Docker Environment

This is a minimal and flexible Docker setup for launching isolated Linux environments easily. It can be used for development, cybersecurity work, experimentation, or any other purpose. It includes automatic cleanup, volume mounting, and reusable configurations.

How to Use

1. Place your project files inside the `src/` folder.
2. Open the project folder.
3. Run the environment using the batch file:

```
run-env.bat
```

This will:

- Build the Docker image using the Dockerfile
- Run a container using docker-compose
- Mount the `src/` folder as `/app` inside the container
- Set `/app` as the working directory
- Launch an interactive shell
- Automatically delete the container when you exit

Folder Structure Explained

```
my-docker-project/
├── Dockerfile              # The active environment definition
├── docker-compose.yml      # Docker Compose configuration
├── run-env.bat             # Batch script for launching from Windows
├── README.md               # This documentation file
├── src/                    # Shared folder with the container
│   └── (your files)
└── ready-env/              # Pre-configured environments (backups/templates)
    ├── kali-full/
    │   ├── Dockerfile
    │   └── docker-compose.yml # optional
    ├── kali-minimal/
    │   ├── Dockerfile
    │   └── docker-compose.yml # optional
    └── alpine-base/
        ├── Dockerfile
        └── docker-compose.yml # optional
```

- `ready-env/` holds different environment presets.

- Each subfolder inside `ready-env/` contains a `Dockerfile` (and optionally a `docker-compose.yml`).
- To activate a different environment, copy its Dockerfile and optionally docker-compose.yml into the root folder.

Switching Environments

When you want to switch to a saved environment:

```
copy .\ready-env\kali-full\Dockerfile .\Dockerfile
copy .\ready-env\kali-full\docker-compose.yml .\docker-compose.yml
```

Then run:

```
run-env.bat
```

File Roles

Dockerfile

Defines the base image and what is installed in the container. Split into logical layers to optimize rebuild time.

docker-compose.yml

Defines how the container is run — mapped folders, working directory, interactive shell settings, etc.

run-env.bat

Batch file for easy execution on Windows.

ready-env/

A collection of reusable Docker environments — used as backups or quick start templates.

Using Different Base Images

You can change the OS or environment by modifying the `FROM` line in your Dockerfile, or by switching to a different template from `ready-env`.

Examples:

```
FROM kalilinux/kali-rolling
FROM debian
FROM alpine
```

Cleanup

- Containers are removed automatically when exited (`--rm`).
- Files in `src/` are persistent.

To manually remove the image:

```
docker image rm pentest-dev
```

That's it. Simple and clean.