

```
FROM denoland/deno:2.1.7
```

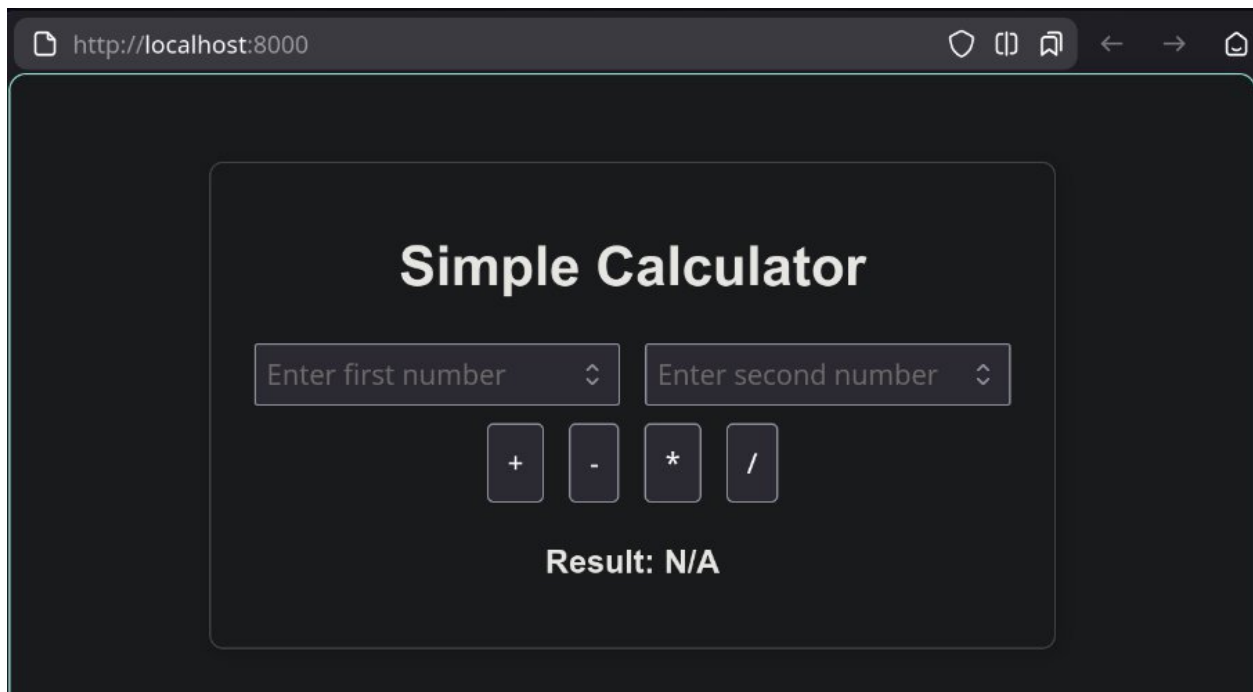
```
EXPOSE 8000
```

```
WORKDIR /app
```

```
COPY . /app
```

```
CMD ["deno", "run", "start"]
```

The Dockerfile uses the deno base docker container, it exposes the port 8000, sets the working directory to `/app` and copies everything from the host current working directory into it, and then finally the command to run the application is `deno run start`, which is a task defined in my `deno.json` file, this command is the same as `deno -R -W -N -E ./src/main.ts`.



Here is the simple html site being served by the container.

```
>docker logs deno
Task start deno -R -W -N -E ./src/main.ts
Download https://deno.land/std@0.197.0/http/server.ts
Download https://deno.land/x/emit@0.31.4/mod.ts
Download https://deno.land/std@0.197.0/async/mod.ts
Download https://deno.land/x/emit@0.31.4/emit.generated.js
Download https://deno.land/x/emit@0.31.4/_utils.ts
Download https://deno.land/x/deno_cache@0.5.2/mod.ts
Download https://deno.land/std@0.197.0/async/abortable.ts
Download https://deno.land/std@0.197.0/async/deadline.ts
Download https://deno.land/std@0.197.0/async/debounce.ts
Download https://deno.land/std@0.197.0/async/deferred.ts
Download https://deno.land/std@0.197.0/async/delay.ts
Download https://deno.land/std@0.197.0/async/mux_async_iterator.ts
Download https://deno.land/std@0.197.0/async/pool.ts
Download https://deno.land/std@0.197.0/async/tee.ts
Download https://deno.land/std@0.197.0/async/retry.ts
Download https://deno.land/x/wasmbuild@0.15.1/loader.ts
Download https://deno.land/x/wasmbuild@0.15.1/cache.ts
Download https://deno.land/std@0.186.0/path/mod.ts
Download https://deno.land/x/deno_cache@0.5.2/cache.ts
Download https://deno.land/x/deno_cache@0.5.2/deps.ts
Download https://deno.land/x/deno_cache@0.5.2/deno_dir.ts
Download https://deno.land/x/deno_cache@0.5.2/file_fetcher.ts
Download https://deno.land/std@0.197.0/assert/assert.ts
Download https://deno.land/x/dir@1.5.1/data_local_dir/mod.ts
Download https://deno.land/std@0.186.0/_util/os.ts
Download https://deno.land/std@0.186.0/path/win32.ts
Download https://deno.land/std@0.186.0/path/posix.ts
Download https://deno.land/std@0.186.0/path/common.ts
Download https://deno.land/std@0.186.0/path/separator.ts
Download https://deno.land/std@0.186.0/path/_interface.ts
Download https://deno.land/std@0.186.0/path/glob.ts
Download https://deno.land/std@0.140.0/fs/ensure_dir.ts
Download https://deno.land/std@0.140.0/fmt/colors.ts
Download https://deno.land/std@0.140.0/path/mod.ts
Download https://deno.land/std@0.140.0/streams/conversion.ts
Download https://deno.land/x/deno_cache@0.5.2/disk_cache.ts
Download https://deno.land/x/deno_cache@0.5.2/dirs.ts
Download https://deno.land/x/deno_cache@0.5.2/http_cache.ts
Download https://deno.land/x/deno_cache@0.5.2/util.ts
Download https://deno.land/x/deno_cache@0.5.2/auth_tokens.ts
Download https://deno.land/std@0.197.0/assert/assertion_error.ts
Download https://deno.land/std@0.186.0/path/_constants.ts
Download https://deno.land/std@0.186.0/path/_util.ts
Download https://deno.land/std@0.186.0/_util/asserts.ts
Download https://deno.land/std@0.140.0/fs/_util.ts
Download https://deno.land/std@0.140.0/_util/os.ts
Download https://deno.land/std@0.140.0/path/win32.ts
Download https://deno.land/std@0.140.0/path/posix.ts
Download https://deno.land/std@0.140.0/path/common.ts
Download https://deno.land/std@0.140.0/path/separator.ts
Download https://deno.land/std@0.140.0/path/_interface.ts
Download https://deno.land/std@0.140.0/path/glob.ts
Download https://deno.land/std@0.140.0/io/buffer.ts
Download https://deno.land/x/deno_cache@0.5.2/lib/deno_cache_dir.generated.js
Download https://deno.land/std@0.140.0/path/_constants.ts
Download https://deno.land/std@0.140.0/path/_util.ts
Download https://deno.land/std@0.140.0/_util/assert.ts
Download https://deno.land/std@0.140.0/bytes/bytes_list.ts
Download https://deno.land/std@0.140.0/bytes/mod.ts
Download https://deno.land/x/deno_cache@0.5.2/lib/snippets/deno_cache_dir-77bed54ace8005e0/fs.js
Download https://deno.land/std@0.140.0/bytes/equal.ts
Server running on http://localhost:8000
Listening on http://localhost:8000/
```

These are the logs of the docker container.