

Docker Workspace setup

TABLE OF CONTENTS

- 1 [Setting up your development environment](#)
- 2 [Docker Troubleshooting](#)

We have prepared a Docker Workspace that is preconfigured with all the development tools necessary to run and test your code for this class. Later in the course, we will discuss virtual machines; for now, you can think of it as a software version of actual hardware.

Setting up your development environment

To begin your setup, you **must** follow the instructions on this [Docker Workspace setup](#) page before moving onto [Workspace Configuration](#).

If you require access to a computer, please consider reaching out to [STEP](#).

Docker Troubleshooting

When opening Docker, you might encounter an error message saying, `Error: required compatibility check: Directory permissions error`, this means that you will need to change the ownership of certain files from `root` to your user. To fix this, in your terminal, run the command `chown` on the `.docker`, `./docker/contexts`, and `./docker/contexts/meta` files (i.e. `chown user_name .docker`).

After running `docker-compose up -d`, you may encounter an error stating `kex_exchange_identification: Connection closed by remote host`. This can be fixed by running `docker-compose down` within your terminal and then following the steps to run the workspace again.

If the Workspace only contains `binutils-2.40.tar.gz` after SSH-ing into it, do not worry and give the Workspace some time to finish loading. It can take a while before you see the `code` directory inside the Workspace. If the `code` directory has not appeared after a very long time, from your terminal delete the `.workspace` directory found in the `docker` directory (`docker/.workspace`). After deleting the directory, in your terminal run `docker-compose up -d`.

If you run into an issue saying that you need to use `unminimize` to access `man` pages. You can just run `sudo unminimize`.
