Miami HackWeek 2022 Quick Start Guide

The following directions contain information on how to clone the github repository, setup your development environment with a docker container, and how to access data from S3.

|  |  |
| --- | --- |
| Clone git repository | Assuming you have git installed on your computer, run the following command on your terminal:  git clone <https://github.com/Channel-Logistics/MHW-2022.git>  Next, please create your own git branch with your name to develop on:  git branch <name>  git checkout <name> |
| Downloading docker desktop | Go to the following link and follow the download instructions for docker desktop -> <https://www.docker.com/products/docker-desktop>  You need this software to build and run docker containers on your computer. If you’re on windows, you will need to enable Hyper-V. Click the following link for instructions -> <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>  After downloading docker desktop and completing additional configuration, you may need to restart your computer. |
| Building the docker image | Open your project folder in your selected IDE, then start a terminal process in the project folder. Assuming you already have docker installed on your computer, run the following command:  docker build -t <image\_name> .  This command will build a docker image with the instructions specified in the provided dockerfile. The -t flag allows you to “tag” the image with a specified name. |
| Running the docker container | After building the docker image, you use the docker run command to deploy a container from the target image. Use the following command to create a container:  docker run -it –name <name of container> -v <absolute path of local src folder>:<target directory> <image name>  The -it flags run an interactive terminal on the docker container. The -v flags allows you to create a volume between your computer and the containers; it essentially allows file sharing between a local directory and a container directory. In this case, I specified the absolute path to my src folder of my project directory as the source directory and then /src as the target directory. |
| Reading data from S3 | If you’re using the docker container, this is fairly simple with pandas:  df = pd.read\_parquet(<s3\_uri>)  df = pd.read\_parquet("s3://mhw-2022/preprocessed/ais/ihs/position-history/5\_min/default/date=2021-10-20/20211020000100\_data.parquet.gzip")  If you’re not using the docker container, you can access the files programmatically with either boto3 or pandas.   If you want to use pandas, install the ffspec and s3fs python packages and setup your access key and secret access key in the credentials file under the .aws directory (for linux -> ~/.aws; for windows -> C:/Users/<User>/.aws).   For boto3, you also need to setup the credentials file. You can grab files (tabular files) with the following command:  s3\_client = boto3.client(‘s3’)  s3\_client.get\_object(Bucket=”mhw-2022”, Key=”<s3\_key>”)[“Body”].read().decode(“utf-8”) |
| Fixing gdal import error | Sometimes if you import gdal, you may receive the following error:    This is because the LD\_LIBRARY\_PATH environment variable is not set. Run the following command to correct the error:  export LD\_LIBRARY\_PATH=/usr/local/lib |