

# MAADSBML Setup and Configuration

Sebastian Maurice, PhD

April 12, 2024

# Setup

## 1. You will need to have Linux OS installed

- In Windows – you can install WSL (windows subsystem for Linux)
- In Mac – Use Terminal
- Or get a VM running with Linux Ubuntu installed
- Further Information can be found in [Appendix N in this PDF](#)

## 2. Install Docker:

- You can install Docker Desktop (Windows/Mac)
- Or in linux run: `sudo apt install docker.io`

## 3. Pull the maadsbml docker container:

- AMD64 container for [Windows/Linux is here](#): `docker pull maadsdocker/maads-batch-automl-otics`
- ARM64 container for [MAC is here](#): `docker pull maadsdocker/maads-batch-automl-otics-arm64`

# Setup Local Folders

4. Create Local File Folders in your computer – these MUST be the following:

- a) {YOUR LOCAL FOLDER PATH}/csvuploads
- b) {YOUR LOCAL FOLDER PATH}/pdfreports
- c) {YOUR LOCAL FOLDER PATH}/autofeatures
- d) {YOUR LOCAL FOLDER PATH}/outliers
- e) {YOUR LOCAL FOLDER PATH}/sqlloads
- f) {YOUR LOCAL FOLDER PATH}/networktemp
- g) {YOUR LOCAL FOLDER PATH}/networks
- h) {YOUR LOCAL FOLDER PATH}/exception
- i) {YOUR LOCAL FOLDER PATH}/staging

Where {YOUR LOCAL FOLDER PATH} is the ROOT folder on your local computer

# Setup

## 5. Run the Docker Container with the following command:

```
docker run -d -v {YOUR LOCAL FOLDER  
PATH}/csvuploads:/maads/agentfilesdocker/dist/maadsweb/csvuploads:z  
-v {YOUR LOCAL FOLDER PATH}/pdfreports:/maads/agentfilesdocker/dist/maadsweb/pdfreports:z  
-v {YOUR LOCAL FOLDER PATH}/autofeatures:/maads/agentfilesdocker/dist/maadsweb/autofeatures:z  
-v {YOUR LOCAL FOLDER PATH}/outliers:/maads/agentfilesdocker/dist/maadsweb/outliers:z  
-v {YOUR LOCAL FOLDER PATH}/sqlloads:/maads/agentfilesdocker/dist/maadsweb/sqlloads:z  
-v {YOUR LOCAL FOLDER PATH}/networktemp:/maads/agentfilesdocker/dist/maadsweb/networktemp:z  
-v {YOUR LOCAL FOLDER PATH}/networks:/maads/agentfilesdocker/networks:z  
-v {YOUR LOCAL FOLDER PATH}/exception:/maads/agentfilesdocker/dist/maadsweb/exception:z  
-v {YOUR LOCAL FOLDER PATH}/staging:/maads/agentfilesdocker/dist/staging:z  
-p 5595:5595 -p 5495:5495 -p 10000:10000 --env TRAININGPORT=5595 --env PREDICTIONPORT=5495 --env  
ABORTPORT=10000 --env COMPANYNAME=OTICS --env MAXRUNTIME=20 --env MAINHOST=127.0.0.1  
maadsdocker/maads-batch-automl-otics
```

**Note: Replace {YOUR LOCAL FOLDER PATH} with the step in 4.**

# Setup

5b. If everything went well you will see the running container:

```
seb@DESKTOP-H0DIAMM:~$ docker run -d -v /mnt/c/maads/maadsbml/csvuploads:/maads/agentfilesdocker/dist/maadsweb/
csvuploads:z -v /mnt/c/maads/maadsbml/pdfreports:/maads/agentfilesdocker/dist/maadsweb/pdfreports:z
-v /mnt/c/maads/maadsbml/autofeatures:/maads/agentfilesdocker/dist/maadsweb/autofeatures:z
-v /mnt/c/maads/maadsbml/outliers:/maads/agentfilesdocker/dist/maadsweb/outliers:z -v /mnt/c/maads/ma
adsbml/sqlloads:/maads/agentfilesdocker/dist/maadsweb/sqlloads:z -v /mnt/c/maads/maadsbml/networktemp
:/maads/agentfilesdocker/dist/maadsweb/networktemp:z -v /mnt/c/maads/maadsbml/networks:/maads/agentfi
lesdocker/networks:z -v /mnt/c/maads/maadsbml/exception:/maads/agentfilesdocker/dist/maadsweb/excepti
on:z -v /mnt/c/maads/maadsbml/staging:/maads/agentfilesdocker/dist/staging:z -p 5595:5595 -p 5495:549
5 -p 10000:10000 --env TRAININGPORT=5595 --env PREDICTIONPORT=5495 --env ABORTPORT=10000 --env COMPANYNAME=OTICS
--env MAXRUNTIME=20 --env MAINHOST=127.0.0.1 maadsdocker/maads-batch-automl-otics
a6d119d761f1c1e9488bd0baefff5153b096e31128e20647d844f4c98ffd3991
seb@DESKTOP-H0DIAMM:~$ docker ps
CONTAINER ID   IMAGE                                     COMMAND                  CREATED        STATUS        PORTS
a6d119d761f1   maadsdocker/maads-batch-automl-otics   "/bin/bash -c 'while..." 7 seconds ago   Up 5 seconds   0.0.0.0:5495->5495/tcp, :::5495->5495/tcp, 0.0.0.0:5595->5595/tcp, :::5595->5595/tcp, 0.0.0.0:10000->10000/tcp, :::10000->10000/tcp
lucid_galois
seb@DESKTOP-H0DIAMM:~$
```

RUN: **docker ps** to see the running container

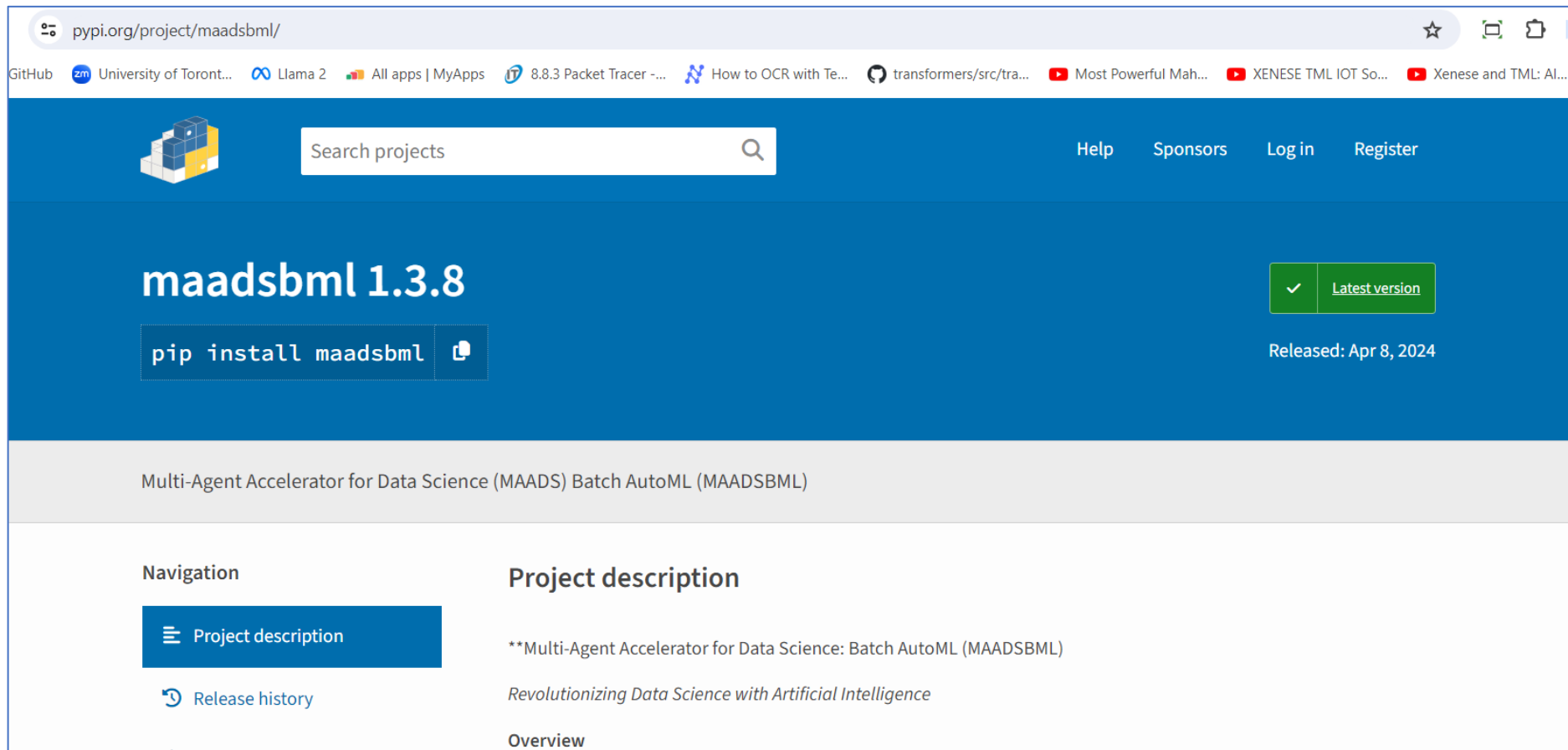
Note: if you get a docker.sock error – just do:

- Run: **sudo chmod 666 /var/run/docker.sock**
- Then Re-run the docker Run command



# Python File

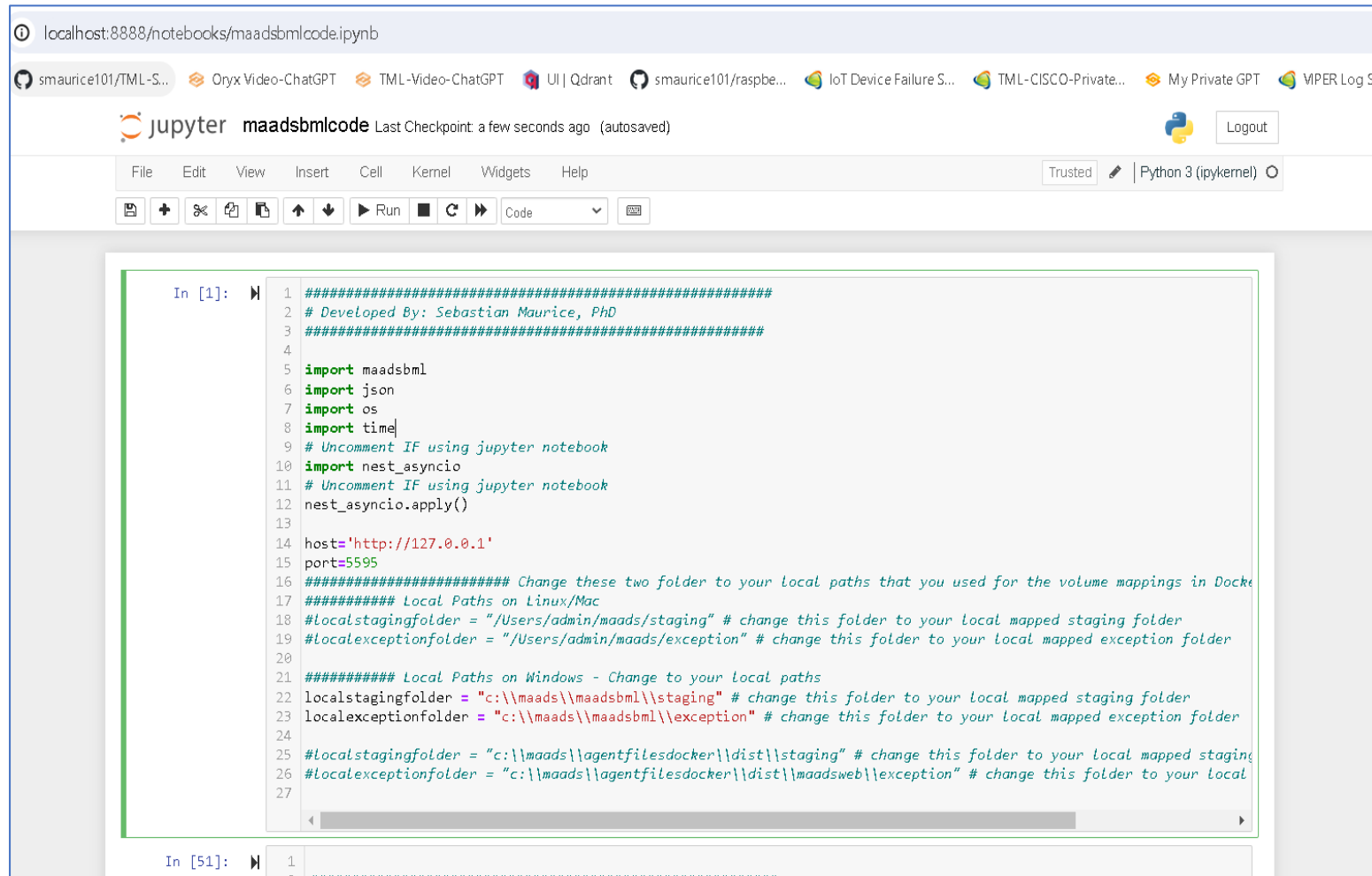
6. Pip install [maadsbml Python Library](https://pypi.org/project/maadsbml/): `pip install maadsbml`



The screenshot shows the PyPI project page for **maadsbml 1.3.8**. The page has a blue header with the PyPI logo, a search bar, and links for Help, Sponsors, Log in, and Register. Below the header, the project name **maadsbml 1.3.8** is displayed in large white text. To the right of the name is a green button with a checkmark and the text **Latest version**. Below the name is a dark blue button with the text `pip install maadsbml` and a copy icon. To the right of this button is the text **Released: Apr 8, 2024**. Below the main content area is a light gray bar with the text **Multi-Agent Accelerator for Data Science (MAADS) Batch AutoML (MAADSBML)**. At the bottom, there is a navigation section with a blue button labeled **Project description** and a link labeled **Release history**. To the right of the navigation section is the **Project description** section, which contains the text **\*\*Multi-Agent Accelerator for Data Science: Batch AutoML (MAADSBML)** and the subtitle *Revolutionizing Data Science with Artificial Intelligence*. Below the project description is the **Overview** section.

# Python File Configurations

7. Now open the MAADSBML Jupyter Python Notebook called [maadsbmlcode.ipynb](#)



```
In [1]: 1 #####
2 # Developed By: Sebastian Maurice, PhD
3 #####
4
5 import maadsbml
6 import json
7 import os
8 import time
9 # Uncomment IF using jupyter notebook
10 import nest_asyncio
11 # Uncomment IF using jupyter notebook
12 nest_asyncio.apply()
13
14 host='http://127.0.0.1'
15 port=5595
16 ##### Change these two folder to your local paths that you used for the volume mappings in Docker
17 ##### Local Paths on Linux/Mac
18 #localstagingfolder = "/Users/admin/maads/staging" # change this folder to your local mapped staging folder
19 #localexceptionfolder = "/Users/admin/maads/exception" # change this folder to your local mapped exception folder
20
21 ##### Local Paths on Windows - Change to your local paths
22 localstagingfolder = "c:\\maads\\maadsbml\\staging" # change this folder to your local mapped staging folder
23 localexceptionfolder = "c:\\maads\\maadsbml\\exception" # change this folder to your local mapped exception folder
24
25 #localstagingfolder = "c:\\maads\\agentfilesdocker\\dist\\staging" # change this folder to your local mapped staging folder
26 #localexceptionfolder = "c:\\maads\\agentfilesdocker\\dist\\maadsweb\\exception" # change this folder to your local
27
In [51]: 1 #####
2 #####
```

Make the following simple changes to

- **localstagingfolder** (line 22)
- **localexceptionfolder** (line 23)
- These MUST point to the STAGING and EXCEPTION folder paths in **STEP 4.h and 4.i**
- To test the system - run the **RUNDEMO** function. Details on the rundemo function is found [here](#)



# MAADSBML Local Output From Container

8. The MAADSBML container will store the output in the container and also on your host machine

- a) {YOUR LOCAL FOLDER PATH}/csvuploads – THIS IS WHERE YOU WRITE YOUR OWN FILE FOR PROCESSING
- b) {YOUR LOCAL FOLDER PATH}/pdfreports – THIS IS WHERE YOU WILL FIND THE MAADSBML PDF REPORT
- c) {YOUR LOCAL FOLDER PATH}/autofeatures – THIS IS WHERE YOU WILL FIND THE AUTOFEATURES
- d) {YOUR LOCAL FOLDER PATH}/outliers – THIS IS WHERE YOU WILL FIND OUTLIERS
- e) {YOUR LOCAL FOLDER PATH}/sqlloads – THIS CAN BE IGNORED – IT IS A SYSTEM FOLDER
- f) {YOUR LOCAL FOLDER PATH}/networktemp – THIS CAN BE IGNORED – IT IS A SYSTEM FOLDER
- g) {YOUR LOCAL FOLDER PATH}/networks – THIS IS WHERE THE ALGORITHMS ARE STORED
- h) {YOUR LOCAL FOLDER PATH}/exception – THIS IS THE JSON FILE FOR THE ALGORITHM OUTPUT
- i) {YOUR LOCAL FOLDER PATH}/staging – THIS CAN BE IGNORED – IT IS A SYSTEM FOLDER

# Using Your Own Data

## 8. To process your own data –

- YOU MUST STORE YOUR DATA in the {YOUR LOCAL FOLDER PATH}/csvuploads
- The data Must be CSV
- The first column of the data file Must contain a Date column
- The Date Must be in the format: M/D/YYYY
- Example of data files can be found [here](#):
  - Look at: **aesopowerdemand.csv**

1	Date	AESO_Power_Demand	Calgary_Weather	Edmonton_Weather	FtMac_Weather
2	1/1/2014	9641	-5.15	-17.92	-32.4
3	1/2/2014	9648	-0.7	-6.69	-15.45
4	1/3/2014	9979	-4.1	-5.56	-19.3
5	1/4/2014	10044	-16.5	-18.86	-30.1
6	1/5/2014	9956	-19.95	-26.64	-32
7	1/6/2014	10037	-5.55	-15.36	-24.65
8	1/7/2014	9933	-6.4	-13.25	-28.35

- Your CSV must contain column headings
- The Dependent variable MUST be contained in this file
- **ALL DATA IN YOUR CSV MUST BE NUMERIC** (with exception of column headers)

# Docker Setup

For User running WSL (Windows Subsystem for Linux)

## 1. Install wsl:

1. Open Windows Powershell
2. Type: **wsl --install**
3. Then IF THIS IS YOUR FIRST TIME INSTALLING WSL Type: **sudo apt update && sudo apt upgrade**

## 2. Now Install docker engine: **sudo apt install docker.io**

1. Install python: **sudo apt install python3-pip**

## 2. Test Docker Install Type: **docker run hello-world**

(If Docker is installed properly - Docker will pull the hello-world container and you will see a

```
seb@DESKTOP-H0DIAMM:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:03b30c6a3c320ff172b52bd68eddfde6ded08ce47e650fe52de861c5e9df46d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

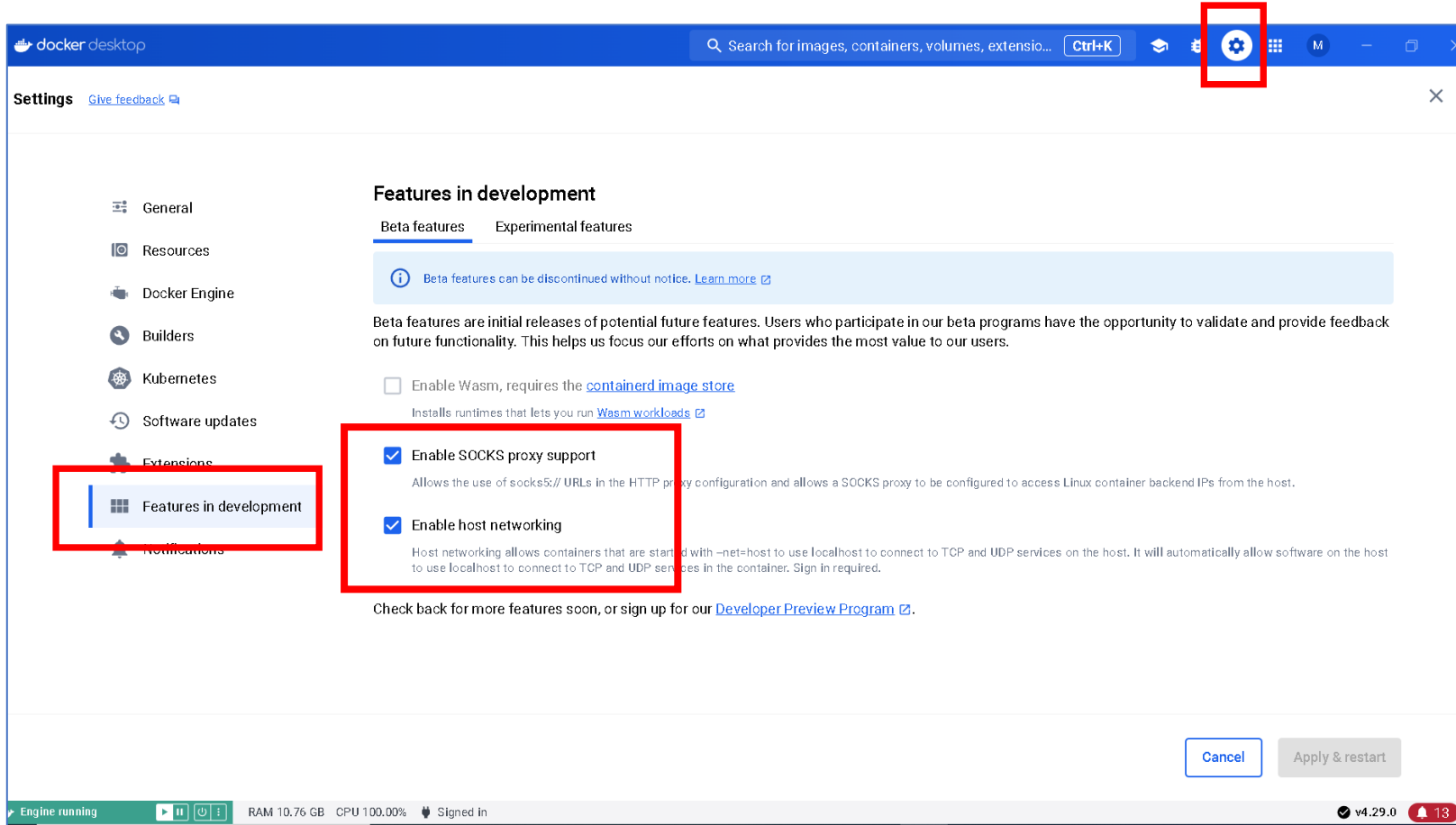
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

# Docker Desktop Setup

- Download the latest Docker Desktop for your computer

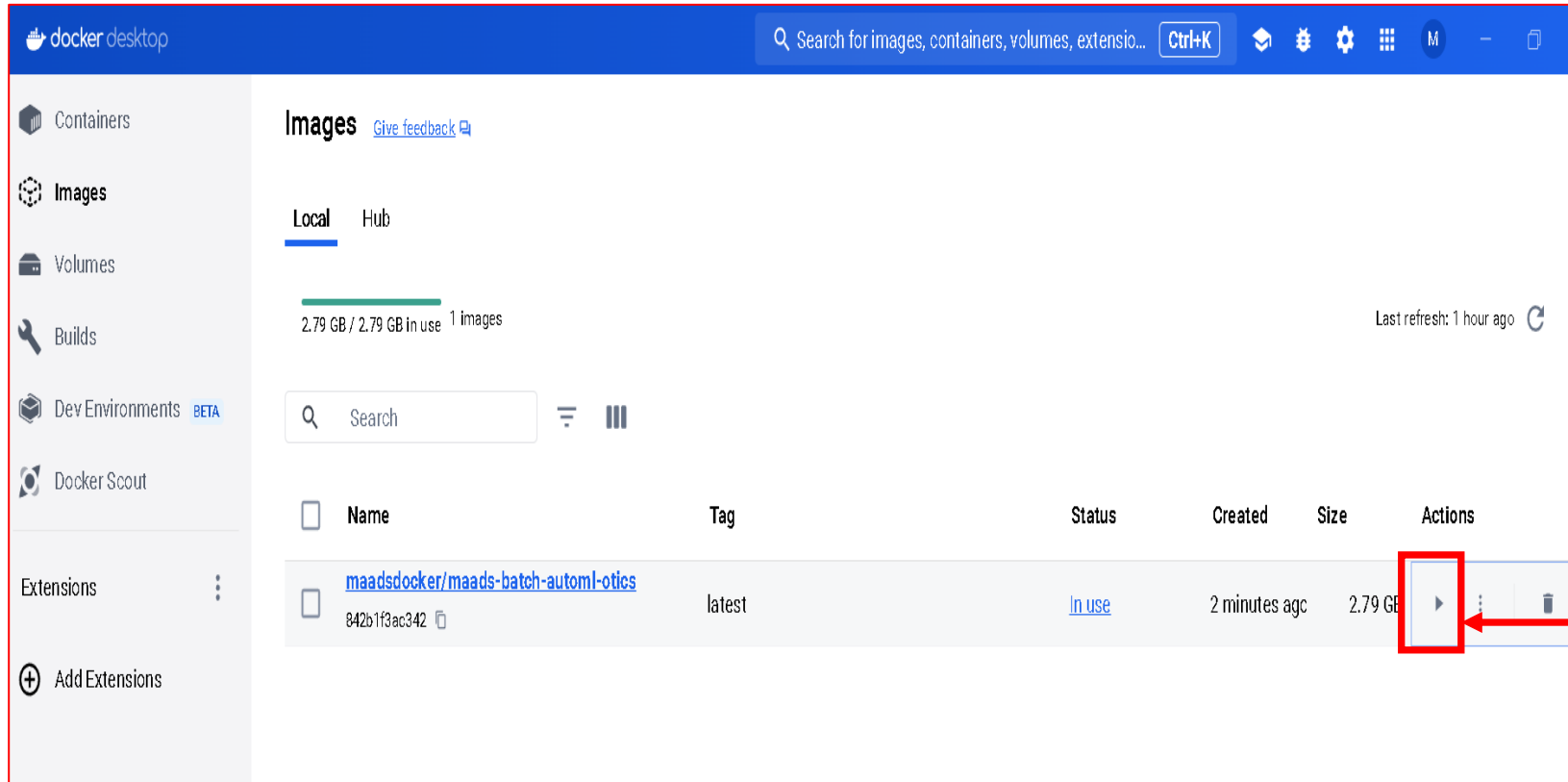


**CHECK the following options:**

1. Enable host networking
2. Enable SOCKS proxy support
3. Click APPLY and RESTART

# Docker Desktop Setup

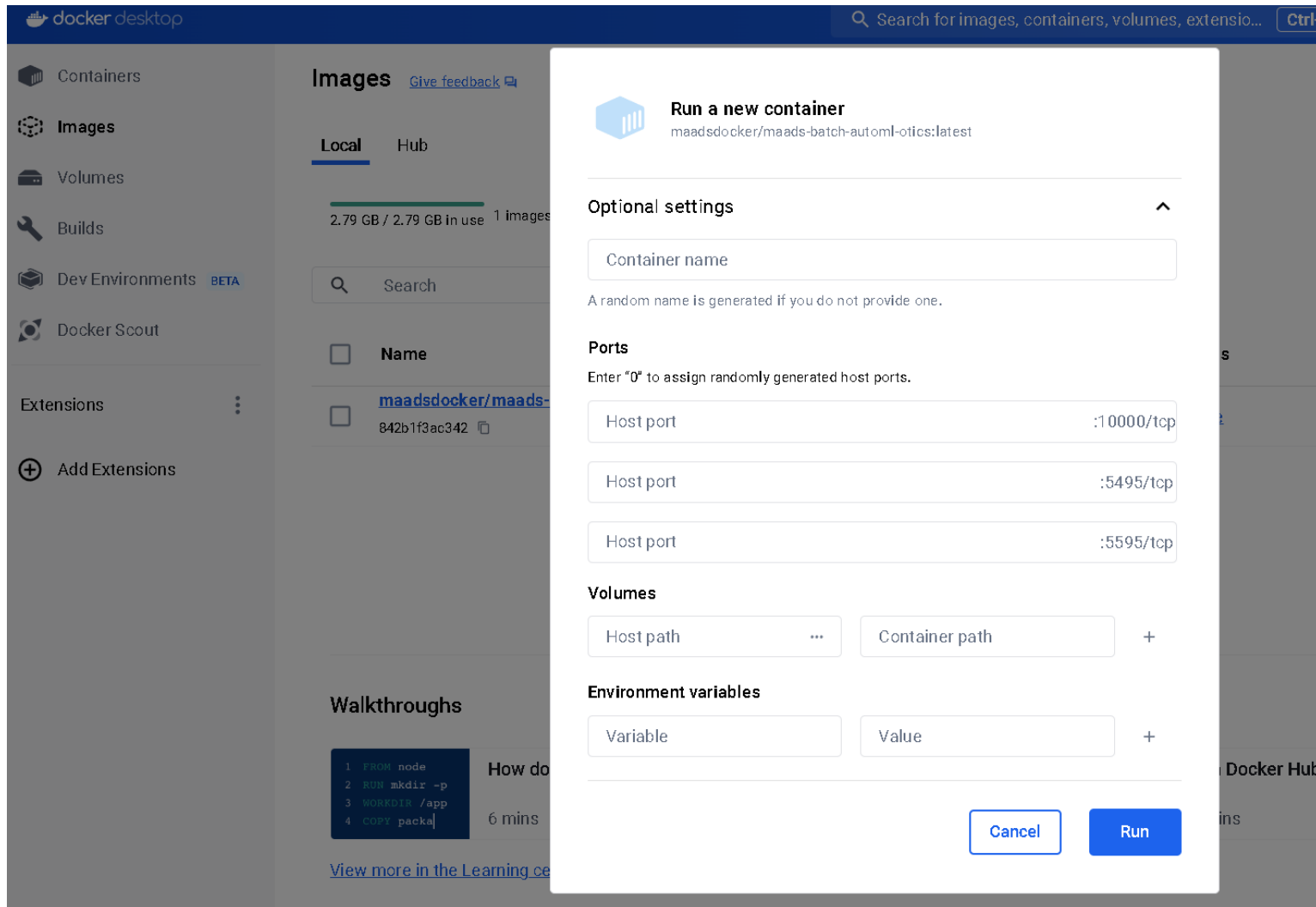
- Download the latest Docker Desktop for your computer



1. Login to your Docker hub account
2. Pull the docker container
3. Click
4. Enable SOCKS proxy support
5. Click RUN

# Docker Desktop Setup

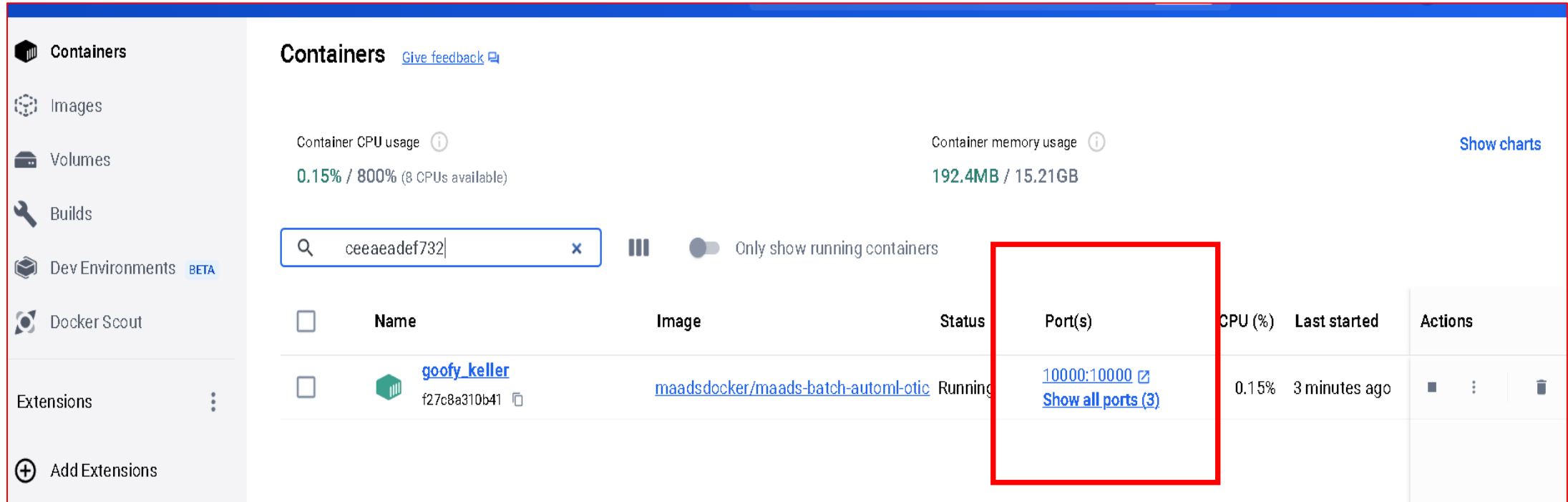
- Enter the Volumes and Environment Variables



- Enter ALL the Volume paths for **HOST PATH** and **CONTAINER PATH** in STEP 5
- ENTER the Environment Variables:
  - **TRAININGPORT=5595**
  - **PREDICTIONPORT=5495**
  - **ABORTPORT=10000**
  - **COMPANYNAME=OTICS**
  - **MAXRUNTIME=20**
  - **MAINHOST=127.0.0.1**
- ENTER the Host Port:
  - 10000
  - 5495
  - 5595
- Click RUN!

# Docker Desktop Setup

- Enter the Volumes and Environment Variables



The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images, Volumes, Builds, Dev Environments (marked BETA), Docker Scout, Extensions, and Add Extensions. The main area is titled 'Containers' and includes a search bar with the text 'ceeaeade732'. Below the search bar, there are two summary cards: 'Container CPU usage' showing '0.15% / 800% (8 CPUs available)' and 'Container memory usage' showing '192.4MB / 15.21GB'. A toggle switch for 'Only show running containers' is also present. A table lists the containers. The first container, 'goofy\_keller', is highlighted with a red box around its 'Port(s)' column. The 'Port(s)' column for this container shows '10000:10000' with a link to 'Show all ports (3)'. The table has columns for Name, Image, Status, Port(s), CPU (%), Last started, and Actions.

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
goofy_keller f27c8a310b41	maadsdocker/maads-batch-automl-otic	Running	10000:10000 <a href="#">Show all ports (3)</a>	0.15%	3 minutes ago	

If successful you WILL see the PORTS section filled.

# Docker Setup Common Issues and Resolution

1. **PROBLEM:** You may get Docker.sock permission denied ERROR

1. Try: `sudo chmod 666 /var/run/docker.sock`

2. **REMOVE DOCKER AND RE-install:** `sudo apt remove docker.io` and `sudo apt autoremove`

3. **PROBLEM: Docker Daemon not running**

1. Try the following until Docker Daemon is running and you can run hello-world:

1. `sudo dockerd &`

2. Manually Start Docker:

a. `sudo service docker start`

b. `sudo systemctl start docker`

4. **PROBLEM:** If you get a registry issue then type: `echo "nameserver 8.8.8.8">/etc/resolv.conf`

5. **PROBLEM:** If you get docker: open /var/lib/docker/tmp/GetImageBlob549217035: no such file or directory

a. Type: `systemctl restart docker`

6. **PROBLEM:** if you have mount overlay2 issues try: `rm -rf ~/Library/Containers/com.docker.docker`

7. **PROBLEM:** If you have "too many symbolink" error:

a. `stop docker`

b. `sudo rm -rf /var/lib/docker`

8. **PROBLEM:** IF you get "Deadline exceeded" or "failed to solve with frontend dockerfile.v0:

a. `export DOCKER_BUILDKIT=0`

b. `export COMPOSE_DOCKER_CLI_BUILD=0`

9. **PROBLEM:** If you get IP Tables issue then try: (failed to start daemon: Error initializing network controller: error obtaining controller instance: failed to create NAT chain DOCKER: iptables failed: iptables -t nat -N DOCKER: iptables/1.8.7 Failed to initialize nft: Protocol not supported)

a. `sudo update-alternatives --set iptables /usr/sbin/iptables-legacy`

b. `sudo update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy`



# Support

- Email: [support@otics.ca](mailto:support@otics.ca)