Apple M1 to deploy Docker and run Keras

By Sniper711. 2021/01/25

==Use Docker/Container (every time)==

A. Open Terminal APP and run well-prepared container.

% **sudo docker start -ai tensorflowkerasarm64flodutot8888ip0000**

Above Blue highlight text can be customized

B. In Container, enter "workbooks" folder, then launch the Jupyter,

cd workbooks

jupyter notebook --allow-root --ip 0.0.0.0

Find the output text:

http://127.0.0.1:8888/?token=xxxxxxxxxxxxxxxxxxxx....xxxxxxxxxx

Copy and paste it on browser.

Now you open Jupyter Notebook on browser.

==Prepare Docker/Container (1st time only)==

A. Install Docker Desktop – Apple M1 Tech Preview version

Download **Docker Desktop for Apple M1** .dmg APP here and run

https://docs.docker.com/docker-for-mac/apple-m1/

Open **Activity Monitor** APP to see if Docker Desktop runs on Intel CPU mode and Dock runs on Apple CPU mode. If yes, it's correct.

B. [Important before Step C] Set File Sharing Folder.

In Docker Desktop APP go *Preferences* settings -> *Resources* -> *File Sharing* -> + your Download folder path and restart Docker Desktop APP.

For example, path should looks like, /Users/(yourID)/Downloads

C. Pull/Download ARM64 version Tensorflow image

Open **Terminal** APP

Login *hub.Docker.com* website. On top left page search *Tenslorflow*, on the left page search *ARM64* build. You will see a long list of results.

I choose to use https://hub.docker.com/r/flodutot/tensorflow_aarch64

flodutot/tensorflow_aarch64.

By flodutot • Updated 2021/1/22.

Full Tensorflow 2.4 for aarch64.

Copy Docker Pull Command on this page:

% **docker pull flodutot/tensorflow_aarch64**

In Terminal APP, paste and run above command.

In Terminal APP, run command to check if docker image is there.

% sudo docker images

D. 1st time Run Docker container, install softwares:

In Terminal APP, run command and enter container.

% sudo docker run -p 8888:8888 -it -v $(pwd):/Users/sniper711/dockers --name tensorflowkerasarm64flodutot8888ip0000 flodutot/tensorflow_aarch64:latest

Above Blue highlight text can be customized

In Container, install and upgrade pip command,

In Container, install jupyter notebook,

In Container, install keras,

In Container, install xlrd,

In Container, install pandas,

In Container, install matplotlib,

In Container, install vim.tiny,

```
# python3 -m pip install --upgrade pip
# python3 -m pip install jupyter
# pip install keras
# pip install xlrd
# pip install pandas
# apt-get install python3-matplotlib
# apt-get install vim-tiny
```

E. Make an empty working folder.

After Jupyter launch you will see all Container folders and files, what a mess!

Suggest you to make an empty folder for Jupyter dedicatedly.

Therefore, at the default "/" path, make a folder calls "workbooks",

```
# mkdir workbooks
```

F. In Container, 1st time run Jupyter on browser:

Enter "workbooks" folder, then launch the Jupyter,

```
# cd workbooks
# jupyter notebook --allow-root --ip 0.0.0.0
```

Find the output text:

http://127.0.0.1:8888/?token=xxxxxxxxxxxxxxxxxxxx....xxxxxxxxxx

Copy and paste it on browser.

Now you open Jupyter Notebook on browser for the 1st time.

==Backup Container as a new Docker Image (on demand)==

A. Open Terminal APP and check all available Docker Images.

```
% sudo docker images
```

Output like this:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
flodutot/tensorflow_aarch64	latest	980408f34a61	3 days ago	1.56GB

B. Check all available Containers.

% sudo docker ps -a

Output like this:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
de77256dc802	flodutot/tensorflow_aarch64:latest	"/bin/bash"	17 hours ago	Exited (0) 9 minutes ago		tensorflowkerasarm64flodutot8888ip0000

C. Backup Container as a new Docker Image.

% **sudo docker commit de77256dc802 tensorflowkerasarm64flodutot8888ip0000:packit**

Above Green highlight text should fill in the Container ID that you want to backup.

Above Blue highlight text can be customized, Docker Image Repository:Tag names as you wish.

Check docker image to see it's newly added.

% sudo docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
tensorflowkerasarm64flodutot8888ip0000	packit	292b33599210	16 hours ago	2.33GB
flodutot/tensorflow_aarch64	latest	980408f34a61	3 days ago	1.56GB

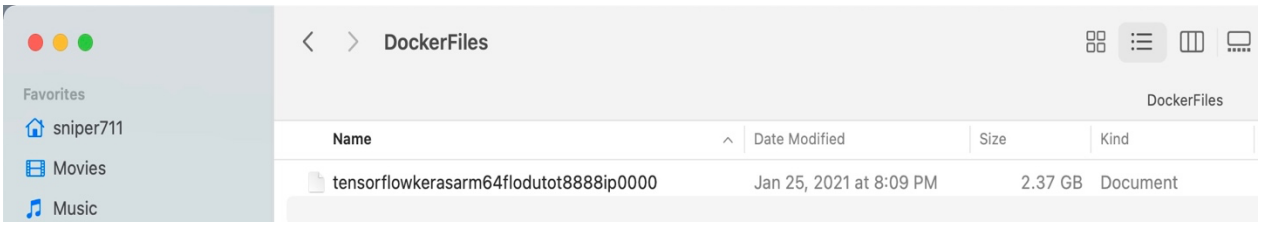
D. Export new Docker Image to hard drive.

% **sudo docker save tensorflowkerasarm64flodutot8888ip0000:packit -o /Users/sniper711/DockerFiles/tensorflowkerasarm64flodutot8888ip0000**

Above Green highlight text should fill in the Docker Image Reposity:Tag that you want to export to hard drive.

Above Blue highlight text can be customized, any hard drive folder and file name as you wish.

E. Check hard drive:



Appendix A – Useful Linux Instructions

==Check docker images==
Instruction $「sudo docker images」. You can see all docker images, all docker images' repository_name:tag.
==Check containers==
Instruction $「sudo docker ps 」, to check running containers
Instruction $「sudo docker ps -a 」, to check all containers (no matter it's running or not). You can see all containers, all containers' ID, all containers' name.

==Delete docker image==
Instruction $「**sudo docker rmi** [add docker image's repository_**name:tag**]」, to delete.
==Delete container==
Instruction $「sudo docker rm [add container's container id]」, or
Instruction $「sudo docker rm [add container's container name]」, to delete.
==Save/Snapshot a container into a docker image (to keep it forever)==
Instruction $「sudo docker commit [container_ID] [repository_name:Tag]」
==Save/Snapshot a docker image as a file to be visible by Ubuntu File Folder application, therefore you can move/backup/manage it ==
Instruction $「sudo docker save [repository_name:Tag] -o [external_ubuntu_file_path/file_name]」
==Rename Container==
Instruction $「sudo docker rename [Original container_name] [Wanted container_name]」

==After reboot, start/attach closed container, find/copy/paste URL to enter Jupyter Notebook==
Container will be deactivated, whenever quit with Ctrl+C, or PC shutdown.
You need to start/attach container.
Instruction $「sudo docker ps -a」, to check all available containers (include activated/deactivated containers).
Instruction $「sudo docker start -ai [add container's container id]」, to start/attach container.
Now you will see an unique URL link. Copy/Paste it to Firefox then enter Jupyter Notebook.

==In case of installation failure, you want to reinstall==
Add "-reinstall" after original install instruction.
For example:
sudo apt install -reinstall cuda

==In case of you failed to download while you do have internet connection, that might caused by DNS setup issue==

Instruction $「sudo gedit /etc/default/docker」

Edit document, to remove "#" sign before "DNS 8.8.8.8", then save and close document.

Reboot PC, or Instruction $「sudo service docker restart」, to reboot PC.

[Sometimes wifi disconnect because of PC enter power saving mode]

Go to Ubuntu OS, under "System" icon, find "Internet" icon.

Double click "internet" icon, wifi will be reconnected.

==Inside Jupyter Notebook, how to run instructions # (without quiting Jupyter Notebook and go to terminal==

You just need to add 1 line of code, which is start with "**!**" sign.

For example: 「! pip install」, or「! ls」, or「! cd/home/(your Ubuntu ID)/container」.

==Inside Jupyter Notebook, "Tab" key is useful (part 1)==

If you run instruction # inside Jupyter Notebook with "!" sign, and don't know all the optional parameters, here is a useful way by press "Tab" key. (without quit Jupyter Notebook)

For example, you can add 1 line of code「! Plt.」, and press "Tab" key now, it will show all the optional parameters of plt.

==Inside Jupyter Notebook, "Tab" key is useful (part 2)==

If you run instruction # inside Jupyter Notebook with "!" sign, and you need to call the dataset directory, here is a useful way by press "Tab" key. (without quit Jupyter Notebook)

For example, you can add 1 line of text 「p = patient.PatientData("/home/ and press "Tab" now, it will automatically output all the sub-directory.