

## Installation of AgensGraph in Docker for Windows and MacOS

Author: Zhengtong Yan

Some students who use Windows and MacOS systems said they have some problems when installing and using the AgensGraph, such as the libxml problem in Windows and other problems in MacOS. AgensGraph provide the docker installation method which can solve your problems (I have tested in my Windows 10 computer), so this guide will teach you how to install AgensGraph in Docker to solve those problem. Docker is an open-source platform that uses containers. Developers use it to create, deploy, and run different applications. Unlike running a virtual machine, you don't need to create a virtual operating system. You can run applications using the system kernel.

### References:

- **Install Docker Desktop on Windows**  
<https://docs.docker.com/desktop/windows/install/>
- **Install Docker Desktop on Mac**  
<https://docs.docker.com/desktop/mac/install/>
- **YouTube video: Installing Docker on Mac OS**  
[https://www.youtube.com/watch?v=y6QJV\\_RSUEM](https://www.youtube.com/watch?v=y6QJV_RSUEM)
- **Docker Postgres Backup/Restore Guide (with examples)**  
<https://simplebackups.com/blog/docker-postgres-backup-restore-guide-with-examples/#postgres-restore-database-command-on-docker>
- **How to Restore Database Dumps for Postgres in Docker Container**  
<https://simkimsia.com/how-to-restore-database-dumps-for-postgres-in-docker-container/>
- <https://github.com/docker/for-win/issues/12576>

### Overall Steps:

- **Step 1. Install Docker in Windows or MacOS**
- **Step 2. Run AgensGraph in the Docker and Import the Dataset**
- **Step 3. Connect to AgensGraph and Conduct the Queries**

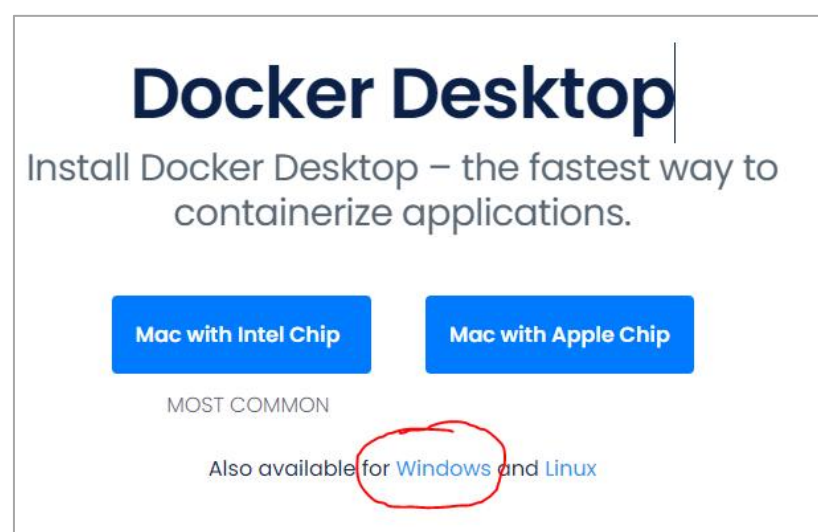
Next, I will use Windows 10 as an example to how you how to install docker. For MacOS users, you can also follow step 2 and step 3 after installing Docker in your computer. The installation of Docker in MacOS is much simple than in Windows, you can refer to the documentation and the following video

- ✧ **Documentation:** <https://docs.docker.com/desktop/mac/install/>
- ✧ **Video:** [https://www.youtube.com/watch?v=y6QJV\\_RSUEM](https://www.youtube.com/watch?v=y6QJV_RSUEM)

### Step 1: Install the Docker

#### (1) Download the Docker Desktop Installer

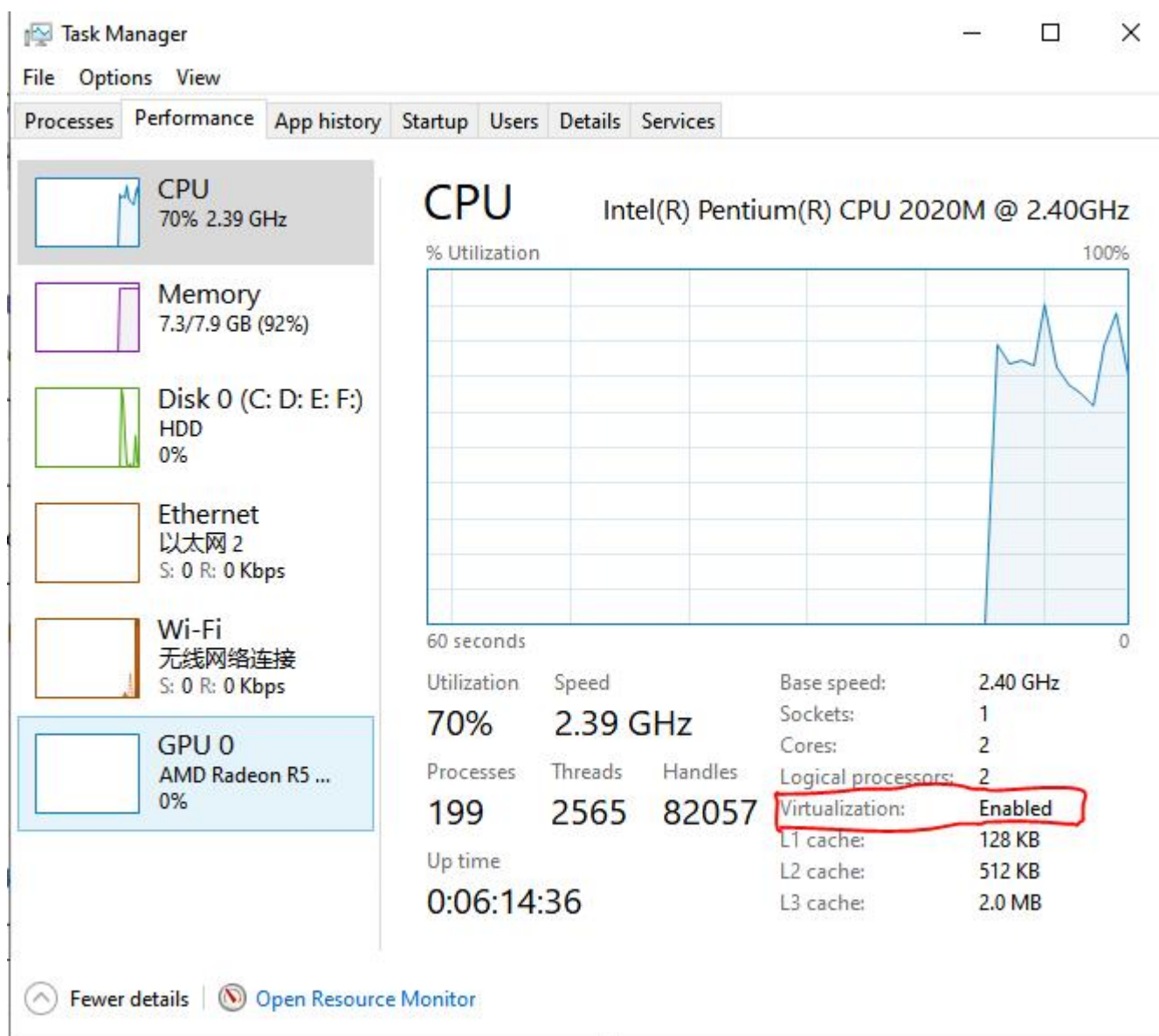
Go to <https://www.docker.com/products/docker-desktop/> and click “Windows” to download the installer. If you use MacOS, please click “Mac with Intel Chip” or “Mac with Apple Chip” to download the installer based on the chip of your computer.



#### (2) Enable Virtualization Windows 10 in BIOS

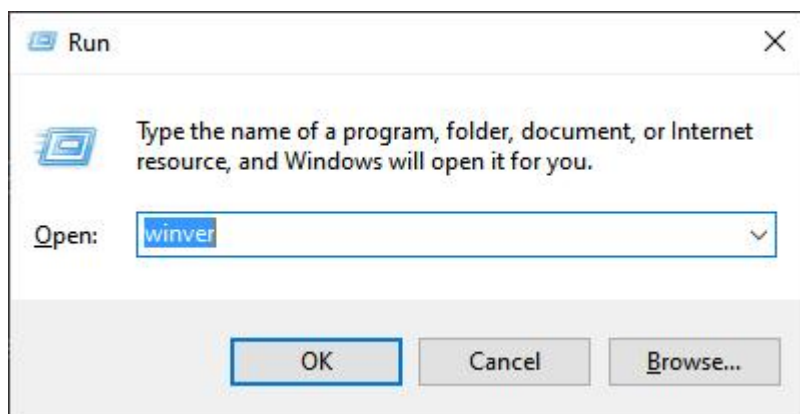
- ✓ Power off your computer.
- ✓ Then press the specific hotkey to enter BIOS. The hotkeys may vary due to different brands. It usually is Esc, F2 or Del, etc.
- ✓ Then navigate to the Advanced tab, press Enter to continue.
- ✓ Select Virtualization and enable it.
- ✓ After that, save the changes and reboot your computer.

Finally, check in the task manager to ensure the virtualization is enabled as follows:



### (3) Check your windows version

Press Win+R and then type [winver](#) to check the information of your windows version.



You can see that my Windows 10 version is Windows 10 Pro with version 2004(OS Build 19041.1288).



#### (4) Enable Hyper-V or WSL 2 backend

Your Windows machine must meet the following requirements in order to successfully install Docker, you can find the information from:

<https://docs.docker.com/desktop/windows/install/>

### System requirements

Your Windows machine must meet the following requirements to successfully install Docker Desktop.

WSL 2 backend

Hyper-V backend and Windows containers

#### Hyper-V backend and Windows containers

- Windows 11 64-bit: Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Pro 2004 (build 19041) or higher, or Enterprise or Education 1909 (build 18363) or higher.

For Windows 10 and Windows 11 Home, see [System requirements for WSL 2 backend](#).

- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
  - 64 bit processor with [Second Level Address Translation \(SLAT\)](#)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see [Virtualization](#).

If your system meets the requirements (see the red rectangle), you can use the Hyper-V backend. But if you use Windows 10 Home, you need to use the WSL 2 backend. **I recommend you to use the Hyper-V backend because it is more easy to configure and use than WSL 2 backend.**

To enable Hyper-V, open Control Panel in Windows and click “Programs”.

Adjust your computer's settings

View by: Category

System and Security

Review your computer's status

Save backup copies of your files with File History

Backup and Restore (Windows 7)

User Accounts

Change account type

Network and Internet

View network status and tasks

Appearance and Personalization

Hardware and Sound

View devices and printers

Add a device

Adjust commonly used mobility settings

Clock and Region

Change date, time, or number formats

Programs

Uninstall a program

Ease of Access

Let Windows suggest settings

Optimize visual display

Then, click [Turn Windows features on or off](#):

Programs and Features

Uninstall a program

Turn Windows features on or off

View installed updates

Run programs made for previous versions of Windows

How to install a program

Default Programs

Change default settings for media or devices

Java

Next, select [Hyper-V and the two sub-folders](#), click OK. You need wait several minutes and then restart the computer.

Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

☐

Active Directory Lightweight Directory Services

☐

Containers

☐

Data Center Bridging

☐

Device Lockdown

☐

Guarded Host

☒

Hyper-V

- ☒

Hyper-V Management Tools
- ☒

Hyper-V Platform

☒

Internet Explorer 11

☐

Internet Information Services

☐

Internet Information Services Hostable Web Core

☐

Legacy Components

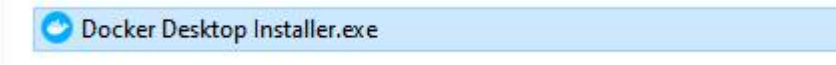
OK

Cancel

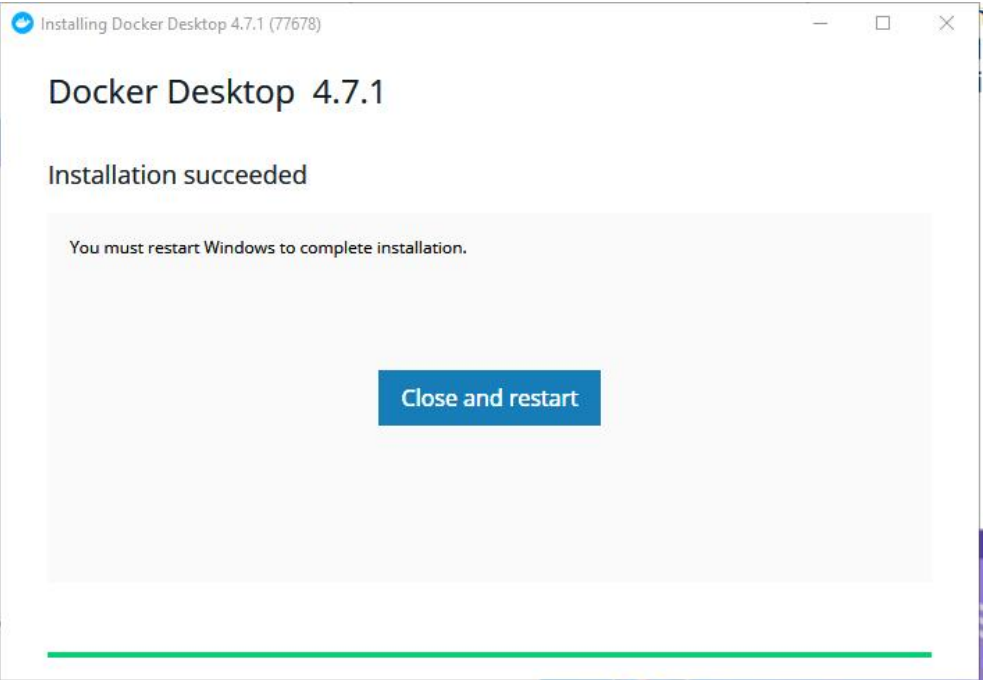
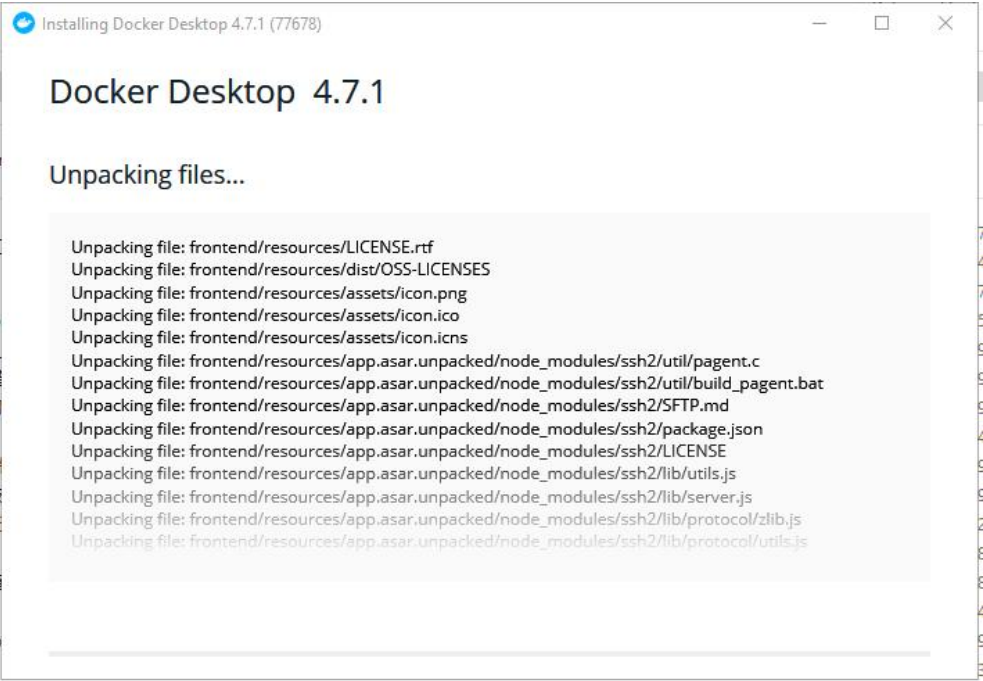
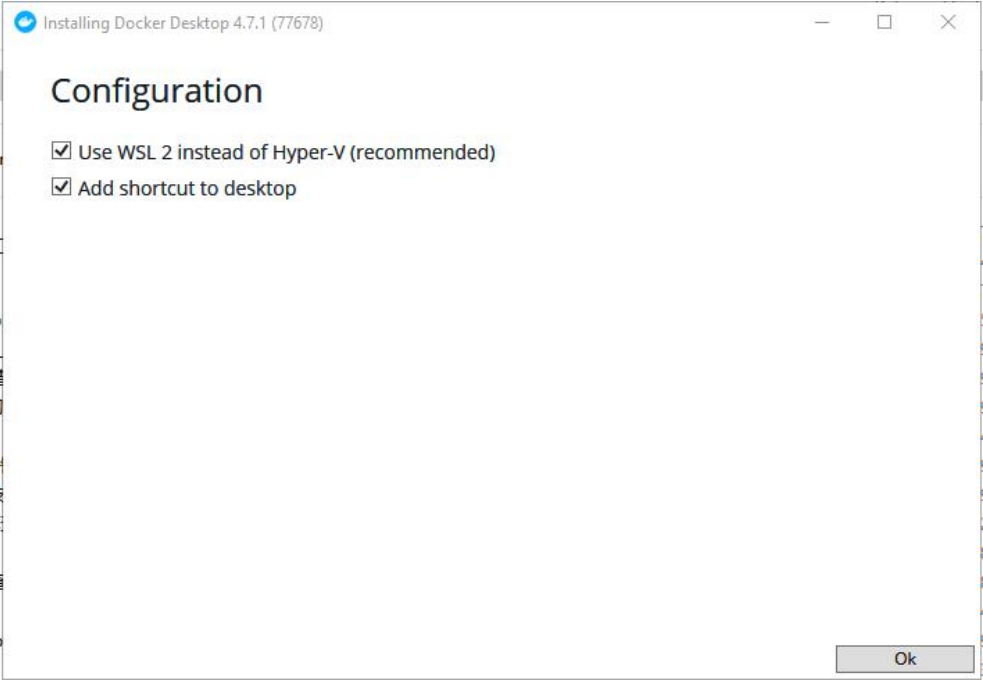


(5) Install Docker

Double click the installer “Docker Desktop Installer.exe”:



Do Not select “Use WSL 2 instead of Hyper-V”, here I made a mistake!



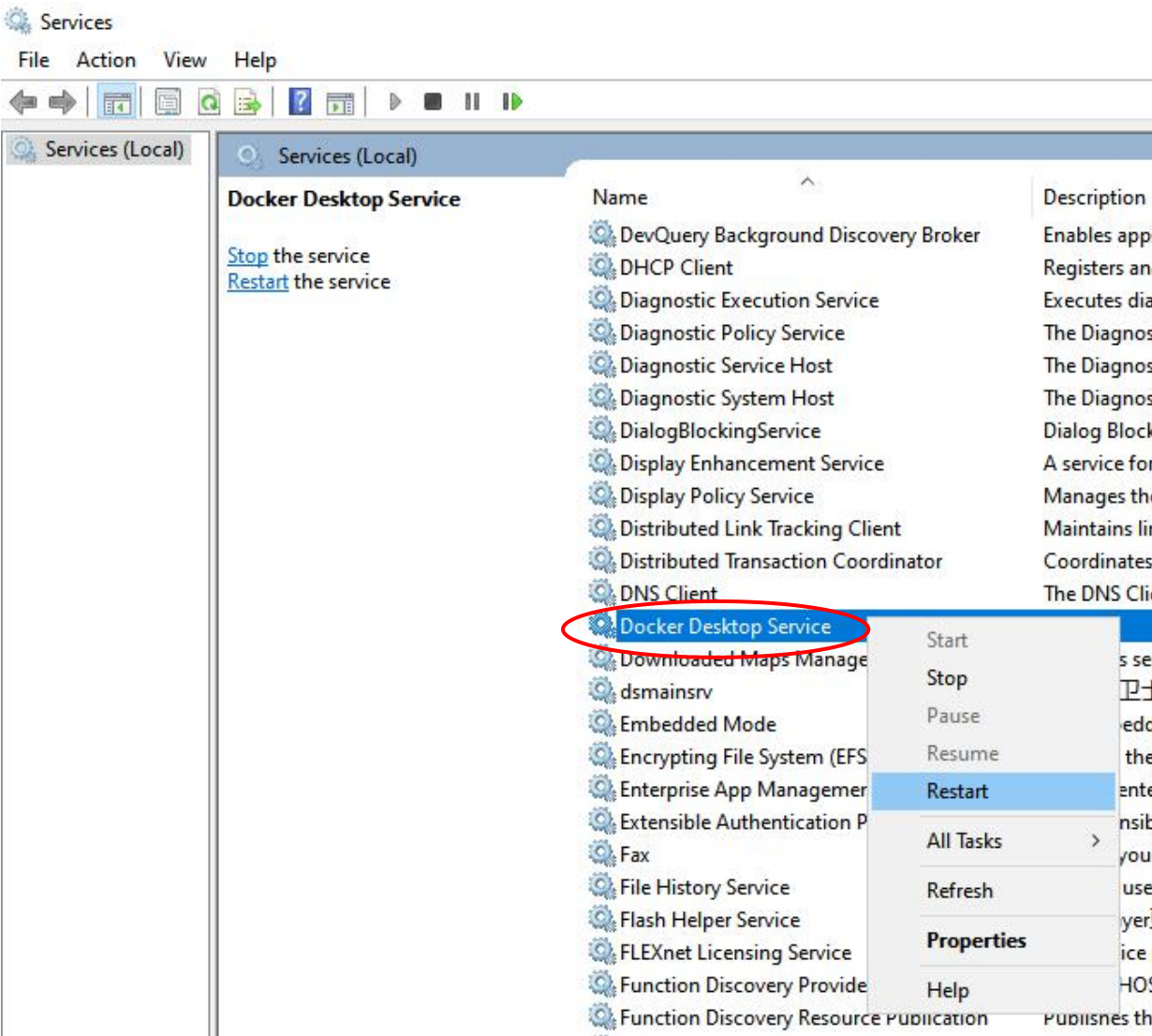
After the installation is finished, open the [settings.json](#) file in C:\Users\Administrator\AppData\Roaming\Docker.

er > System (C:) > Users > Administrator > AppData > Roaming > Docker				
Name	Date modified	Type	Size	
cni	5/5/2022 12:29	File folder		
extensions	5/5/2022 12:06	File folder		
.npsid	5/5/2022 12:16	NPSID File	1 KB	
.trackid	5/5/2022 11:48	TRACKID File	1 KB	
http_proxy.json	5/5/2022 17:33	JSON File	1 KB	
last-start-linux-daemon.json	5/5/2022 13:14	JSON File	1 KB	
locked-directories	5/5/2022 12:41	File	1 KB	
settings.json	5/5/2022 13:15	JSON File	3 KB	

Change the “wslEngineEnabled” into false.

```
"useWindowsContainers": false,  
"wslEngineEnabled": false,  
"licenseTermsVersion": 2,
```

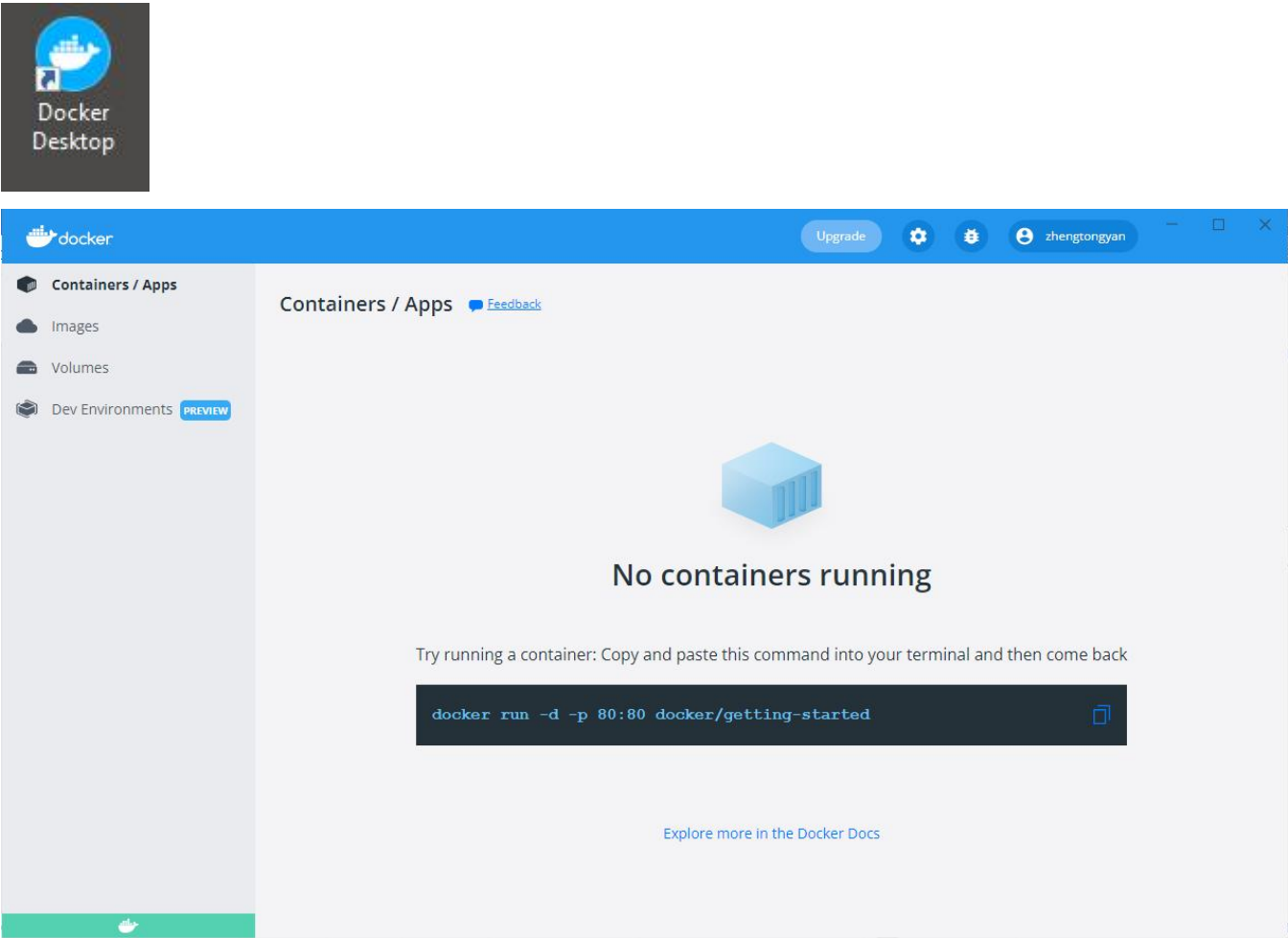
Restart the Docker Desktop Service.



Step 2: Run AgensGraph in the Docker and Import the Dataset

(1) Open and Run Docker

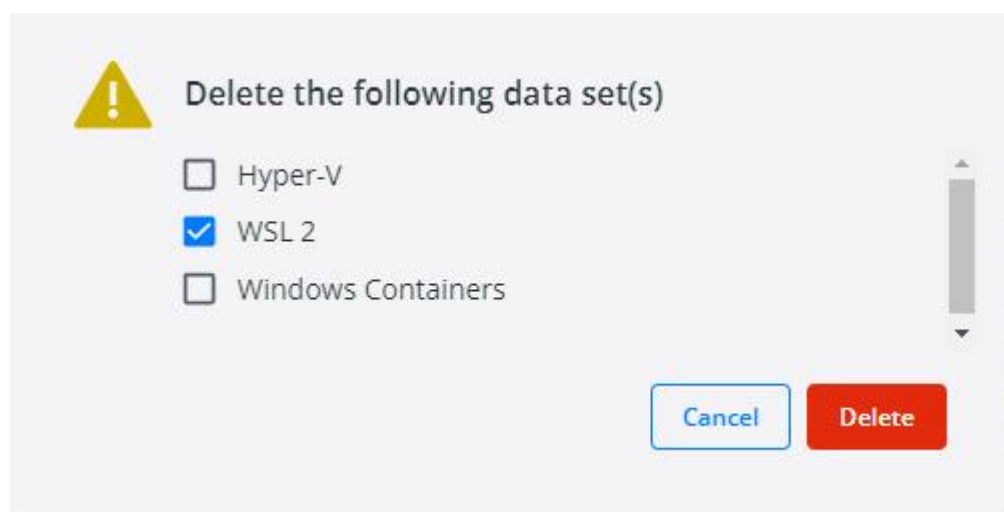
Double click the shortcut in the desktop to open Docker.



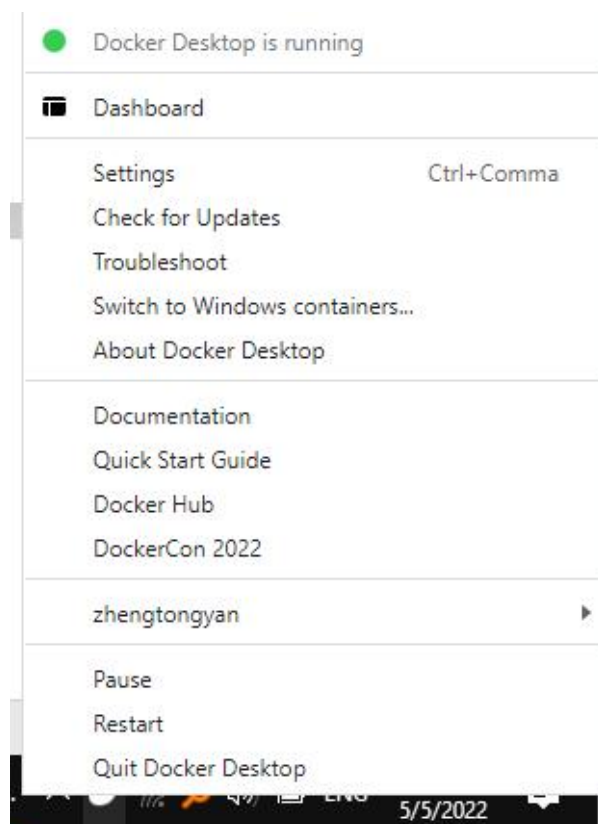
If you your docker shows “Docker desktop stopping”, right click the icon in the windows toolbar as follows:



Click Troubleshoot --> Clean / Purge data --> click on WSL 2 --> Delete as follow to delete the WSL 2 data Sets. Once removed, the docker daemon restarted and it worked fine.



Right click on the icon of docker, you will see “Docker Desktop is running”. This means the Docker has been running successfully.



Open a Powershell and run `docker --version` command to check the information:

```
Windows PowerShell
PS C:\Users\Administrator> docker --version
Docker version 20.10.14, build a224086
PS C:\Users\Administrator> docker version
Client:
 Cloud integration: v1.0.23
 Version:          20.10.14
 API version:      1.41
 Go version:       go1.16.15
 Git commit:       a224086
 Built:            Thu Mar 24 01:53:11 2022
 OS/Arch:          windows/amd64
 Context:          default
 Experimental:     true

Server: Docker Desktop 4.7.1 (77678)
Engine:
 Version:          20.10.14
 API version:      1.41 (minimum version 1.12)
 Go version:       go1.16.15
 Git commit:       87a90dc
 Built:            Thu Mar 24 01:46:14 2022
 OS/Arch:          linux/amd64
 Experimental:     false
containerd:
 Version:          1.5.11
 GitCommit:        3df54a852345ae127d1fa3092b95168e4a88e2f8
runc:
 Version:          1.0.3
 GitCommit:        v1.0.3-0-gf46b6ba
docker-init:
 Version:          0.19.0
 GitCommit:        de40ad0
```

Run `docker run hello-world` command to check the docker is already been running successfully.



```
Windows PowerShell
PS C:\Users\Administrator> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:10d7d58d5ebd2a652f4d93fdd86da8f265f5318c6a73cc5b6a9798ff6d2b2e67
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/
```

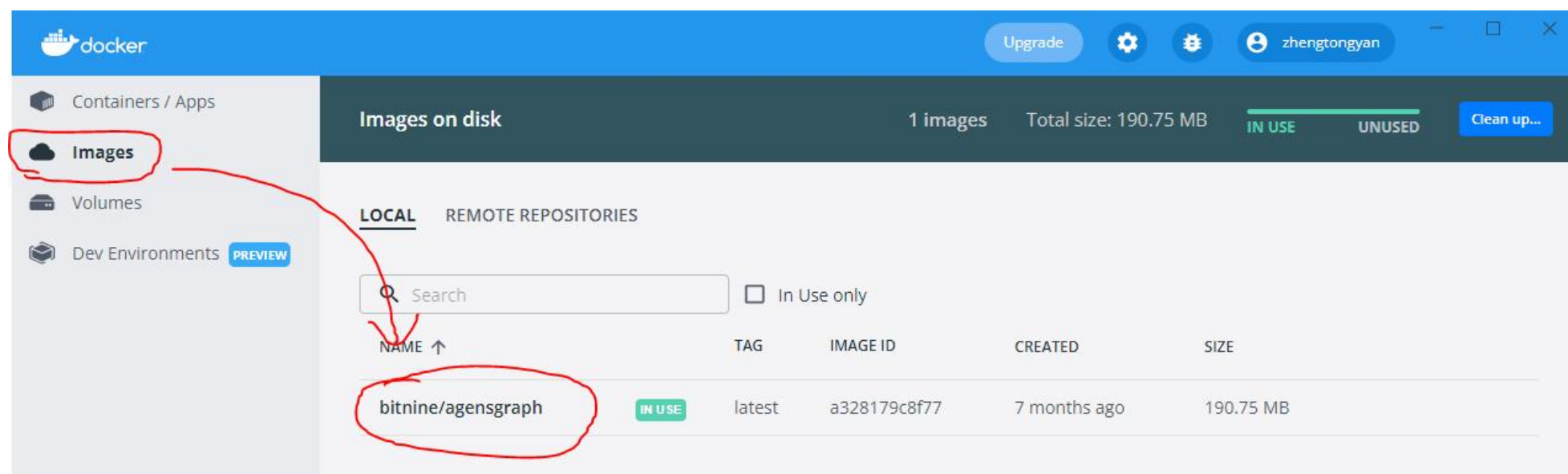
## (2) Install AgensGraph

Open Powershell, run the following command to install AgensGraph image:

```
docker run --name agensgraph -e POSTGRES_PASSWORD=agensgraph -d bitnine/agensgraph:latest
```

```
Select Windows PowerShell
PS C:\Users\Administrator> docker run --name agensgraph -e POSTGRES_PASSWORD=agensgraph -d bitnine/agensgraph:latest
Unable to find image 'bitnine/agensgraph:latest' locally
latest: Pulling from bitnine/agensgraph
a0d0a0d46f8b: Pull complete
25a6d429225d: Pull complete
516a6d54c0d6: Pull complete
67195bd16522: Pull complete
cf34c2a0daf1: Pull complete
87c0f56f0247: Pull complete
a457e511aaac: Pull complete
0e0ada9ff970: Pull complete
Digest: sha256:aa0812aaae0c46158f2dd40345718cb26cc7e2336d8b1beb1ef640447c757cc0
Status: Downloaded newer image for bitnine/agensgraph:latest
0e46809590a582c02a0389c2353b563988187263476c6fff6c3723976c59b3e7
```

Finally, you can see a image in the Docker Desktop:



## (3) Connect to the AgensGraph container

Find the container-id and container-name in which the AgensGraph is running:

```
docker ps -a
```

```
PS C:\Users\Administrator> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0e46809590a5	bitnine/agensgraph:latest	"docker-entrypoint.s... "	2 hours ago	Up 2 hours	5432/tcp	agensgraph

Run the below command to enter into the container (with the container-id from the previous step, i.e. 0e46809590a5):

```
docker exec -it <container-id> bash
```

```
PS C:\Users\Administrator> docker exec -it 0e46809590a5 bash
bash-5.1#
```

Run the below command to connect to AgensGraph:

```
agens -U postgres -d postgres
```

```
bash-5.1# agens -U postgres -d postgres
agens (11.11)
Type "help" for help.

postgres=#
```

Create a database called "unibench\_m3d" and a graph called "unibench\_graph", and set the graph\_path. The commands are as follows:

```
CREATE DATABASE unibench_m3d;
```

```
CREATE GRAPH unibench_graph;
```

```
SET graph_path = unibench_graph;
```

```
postgres=# CREATE DATABASE unibench_m3d;
CREATE DATABASE
postgres=# CREATE GRAPH unibench_graph;
CREATE GRAPH
postgres=# SET graph_path = unibench_graph;
SET
postgres=#
```

#### (4) Restore the dataset

Docker containers have their own volumes. They have their unique limits, like the disk volumes in your host system. **pg\_restore** command will happen within the container's volume. If the files (**m3d.dmp**) aren't in the docker container, you will need to transfer the file form the host system to the Docker container.

**Open a new Powershell**, and run the following command to find the volumes available in the Docker container (with the container-id from the previous step, i.e. 0e46809590a5)

```
docker inspect -f '{{.Mounts}}' <container-id> | python -m json.tool
```

```
PS C:\Users\Administrator> docker inspect -f '{{.Mounts}}' 0e46809590a5 | python -m json.tool
[
  {
    "Type": "volume",
    "Name": "0aa9c844bd7850955baf2fa032167672567bd9c3101d7019bc84542f4c3700c9",
    "Source": "/var/lib/docker/volumes/0aa9c844bd7850955baf2fa032167672567bd9c3101d7019bc84542f4c3700c9/_data",
    "Destination": "/var/lib/postgresql/data",
    "Driver": "local",
    "Mode": "",
    "RW": true,
    "Propagation": ""
  }
]
```

In this case, we have **/var/lib/postgresql/data** as the volume paths.

Pick a volume and copy the m3d.dmp file by running the following command:

```
docker cp <path of the m3d.dmp> <container-name>:<path-to-volume>
```

In my case, I pick the volume /backups. Which gives us the following.

- ✓ path of the m3d.dmp file in my computer: **e:\Database\_System\Benchmarks\UniBench\Dataset\Unibench\_OLAP\m3d.dmp**
- ✓ container-name: **agensgraph**
- ✓ path-to-volume: **/var/lib/postgresql/data**

```
docker cp e:\Database_System\Benchmarks\UniBench\Dataset\Unibench_OLAP\m3d.dmp agensgraph:/var/lib/postgresql/data
```

```
PS C:\Users\Administrator> docker cp e:\Database_System\Benchmarks\UniBench\Dataset\Unibench_OLAP\m3d.dmp agensgraph:/var/lib/postgresql/data
PS C:\Users\Administrator>
```



It will take some time to transfer the file from the host system into the volume of the container.

Run the following command to restore the dataset (container-name is agensgraph in my docker)

`docker exec <container-name> pg_restore -U postgres -d unibench_m3d -O -w -v /var/lib/postgresql/data/m3d.dmp`

```
PS C:\Users\Administrator> docker exec agensgraph pg_restore -U postgres -d unibench_m3d -O -w -v /var/lib/postgresql/data/m3d.dmp
pg_restore: connecting to database for restore
pg_restore: creating SCHEMA "public"
pg_restore: [archiver (db)] Error while PROCESSING TOC:
pg_restore: [archiver (db)] Error from TOC entry 5; 2615 2200 SCHEMA public egallinucci
pg_restore: [archiver (db)] could not execute query: ERROR: schema "public" already exists
    Command was: CREATE SCHEMA public;

pg_restore: creating COMMENT "SCHEMA public"
pg_restore: creating SCHEMA "unibench_graph"
pg_restore: creating EXTENSION "plpgsql"
pg_restore: creating COMMENT "EXTENSION plpgsql"
pg_restore: creating EXTENSION "hstore"
pg_restore: creating COMMENT "EXTENSION hstore"
pg_restore: creating EXTENSION "pg_dropcache"
pg_restore: [archiver (db)] Error from TOC entry 2; 3079 744615 EXTENSION pg_dropcache
pg_restore: [archiver (db)] could not execute query: ERROR: could not open extension control file "/usr/local/share/postgresql/extension/pg_dropcache.control"
    file or directory
    Command was: CREATE EXTENSION IF NOT EXISTS pg_dropcache WITH SCHEMA public;

pg_restore: creating COMMENT "EXTENSION pg_dropcache"
pg_restore: [archiver (db)] Error from TOC entry 2587; 0 0 COMMENT EXTENSION pg_dropcache
pg_restore: [archiver (db)] could not execute query: ERROR: extension "pg_dropcache" does not exist
    Command was: COMMENT ON EXTENSION pg_dropcache IS 'clears buffer cache';

pg_restore: creating FUNCTION "public.add_gold_column()"

pg_restore: creating INDEX "public.fr_f_idcust"
pg_restore: creating INDEX "public.fr_f_iddate"
pg_restore: creating INDEX "public.fr_f_idgroup"
pg_restore: creating INDEX "public.fr_f_idorder"
pg_restore: creating INDEX "public.mm1_dp_pk"
pg_restore: creating INDEX "public.mm1_f_idcust"
pg_restore: creating INDEX "public.mm1_f_idorder"
pg_restore: creating INDEX "public.mm1_f_orderdate"
pg_restore: creating INDEX "public.mm1_f_pk"
pg_restore: creating INDEX "public.mm2_dp_feedback"
pg_restore: creating INDEX "public.nr_bt_f"
pg_restore: creating INDEX "public.nr_dp_pk"
pg_restore: creating INDEX "public.nr_f_asin"
pg_restore: creating INDEX "public.nr_f_pk"
pg_restore: creating INDEX "public.sm3d_f_asin"
pg_restore: creating INDEX "public.sm3d_f_idcust"
pg_restore: creating INDEX "public.sm3d_f_iddate"
pg_restore: creating INDEX "public.sm3d_f_idorder"
pg_restore: creating INDEX "unibench_graph.ag_edge_end_idx"
pg_restore: creating INDEX "unibench_graph.ag_edge_id_idx"
pg_restore: creating INDEX "unibench_graph.ag_edge_start_idx"
pg_restore: creating INDEX "unibench_graph.customer_idcust_idx"
pg_restore: creating INDEX "unibench_graph.customer_idcust_idx1"
pg_restore: creating INDEX "unibench_graph.knows_end_idx"
pg_restore: creating INDEX "unibench_graph.knows_id_idx"
pg_restore: creating INDEX "unibench_graph.knows_start_idx"
WARNING: errors ignored on restore: 3
```

Now all the tables have been imported into the database called unibench\_m3d, then you can run the example queries (FR, NR and M3D queries)!