

1. Installation of Docker on Windows:

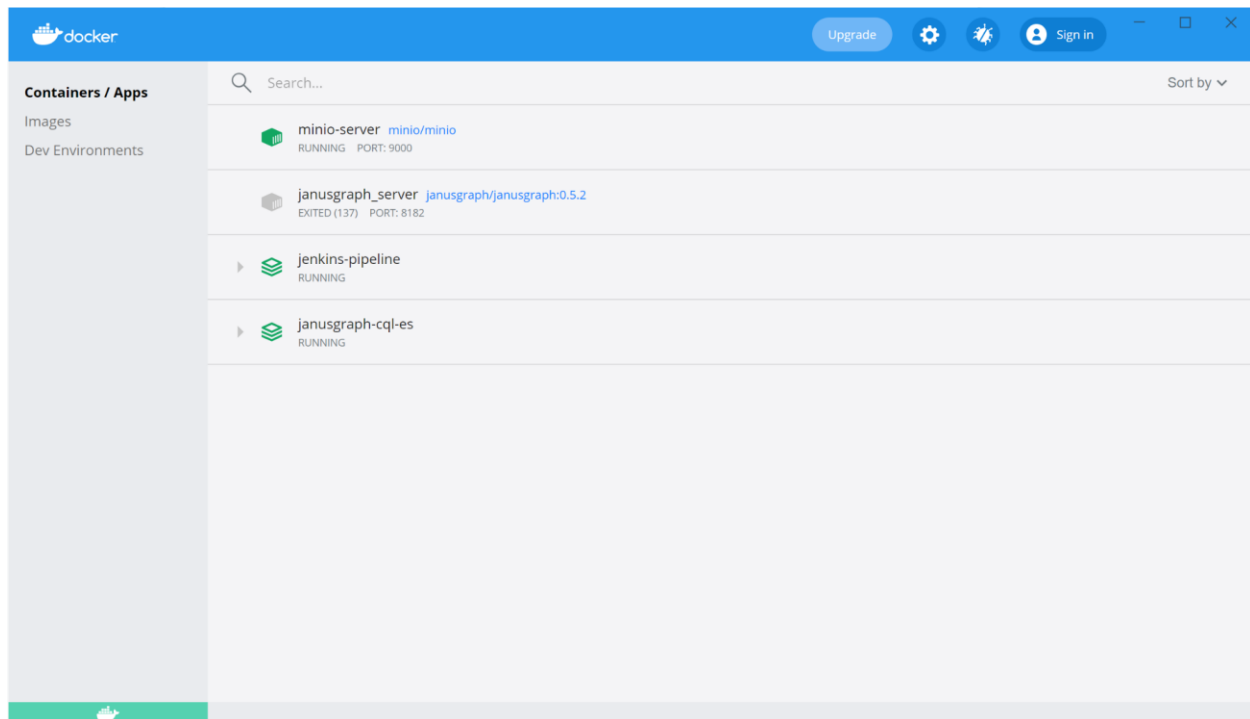
Docker download is here: [Install Docker Desktop on Windows | Docker Documentation](#)

You **need** WSL2 on Windows (**Windows Subsystem for Linux 2**)

This is possible on Windows 10:

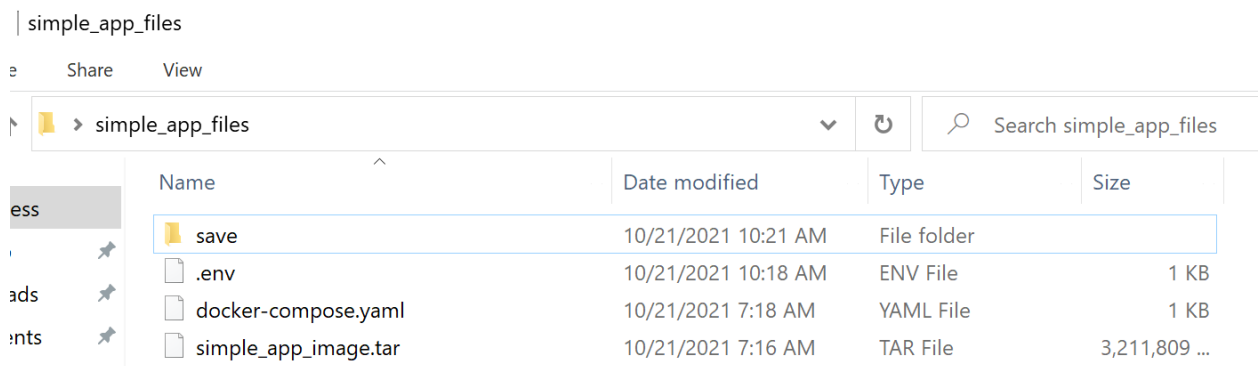
[How to install WSL2 \(Windows Subsystem for Linux 2\) on Windows 10 • Pureinfotech](#)

Once both are installed you can use “docker” in cmd and view running containers in Docker for Windows GUI



2. Running the example docker image

a. Download and unzip simple_app_files:



simple_app_image.tar is the docker image.

docker-compose.yml is needed to run the image
.env file can be changed to mount "save" folder

- b. Load docker image. In cmd run **docker load < simple_app_image.tar**

You should see this output:

```
43c67172d1d1: Loading layer [=====>] 65.57MB/65.57MB
21ec61b65b20: Loading layer [=====>] 991.2kB/991.2kB
1d0dfb259f6a: Loading layer [=====>] 15.87kB/15.87kB
f55aa0bd26b8: Loading layer [=====>] 3.072kB/3.072kB
d464fd09fff6: Loading layer [=====>] 38.07MB/38.07MB
9544d313f086: Loading layer [=====>] 2.048kB/2.048kB
46824cd17241: Loading layer [=====>] 127.8MB/127.8MB
34770a10d86a: Loading layer [=====>] 66.71MB/66.71MB
a2f73abb760e: Loading layer [=====>] 308.1MB/308.1MB
dd168fc288d5: Loading layer [=====>] 4.096kB/4.096kB
04e7796931d0: Loading layer [=====>] 2.68GB/2.68GB
cbfca6981914: Loading layer [=====>] 4.608kB/4.608kB
0a27dd33560e: Loading layer [=====>] 2.56kB/2.56kB
bfa1729a2d00: Loading layer [=====>] 2.56kB/2.56kB
47441d77360d: Loading layer [=====>] 7.68kB/7.68kB
812ce2729bfe: Loading layer [=====>] 1.407MB/1.407MB
a0459e7ad042: Loading layer [=====>] 7.68kB/7.68kB
Loaded image: simple_app:0.0
```

- c. Update .env file to change LOCAL_DIR variable:

LOCAL_DIR=<path>\simple_app_files\save

- d. Run the container with docker-compose:

docker-compose -f docker-compose.yml up -d

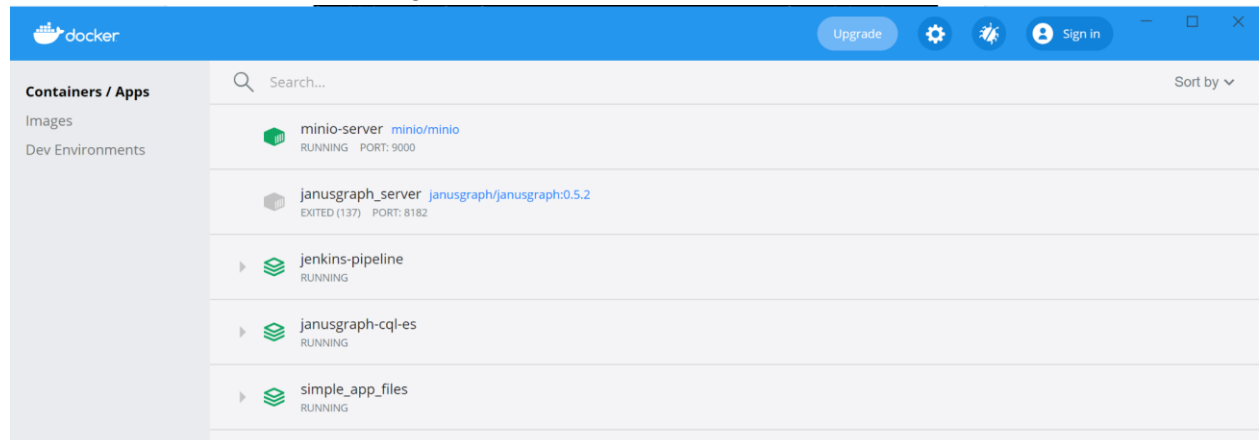
You should see this output:

Docker Compose is now in the Docker CLI, try `docker compose up`

Creating network "simple_app_files_default-network" with driver "bridge"

Creating simple_app ... done

You can see the docker running in the GUI:



To stop the container run: **docker-compose -f docker-compose.yml down**

- e. Run the app: go to http://localhost:5006/simple_app
- f. Pressing “SAVE” will save files to your **mounted** drive

3. Creating docker image:

- a. Unzip docker_example_app.zip and go to simple_app folder:

| PC > Desktop > docker_example_app > simple_app | | | | | Search simple_app | |
|--|---------------------|--------------|------|--|-------------------|--|
| Name | Date modified | Type | Size | | | |
| save | 10/21/2021 10:06 AM | File folder | | | | |
| static | 10/21/2021 1:50 PM | File folder | | | | |
| .env | 10/21/2021 10:06 AM | ENV File | 1 KB | | | |
| conda_env.yaml | 10/20/2021 11:37 PM | YAML File | 1 KB | | | |
| docker-compose.yaml | 10/21/2021 9:50 AM | YAML File | 1 KB | | | |
| Dockerfile | 10/21/2021 9:57 AM | File | 2 KB | | | |
| entrypoint | 10/21/2021 12:13 AM | Shell Script | 1 KB | | | |
| main | 10/21/2021 9:51 AM | Python File | 3 KB | | | |

- b. To build a new docker image run in cmd *from this folder*: **docker build -t simple_app2:0.0 .**

```
[+] Building 34.3s (9/20)
=> [internal] load build definition from Dockerfile
=> == transferring dockerfile: 1.15kB
=> [internal] load .dockerignore
=> == transferring context: 2B
=> [internal] load metadata for docker.io/library/ubuntu:18.04
=> [ 1/15] FROM docker.io/library/ubuntu:18.04@sha256:8fedbd5bd9fb72889c7bbca476949e18593cebed9b1fb9edf5b79dbbacc
=> == resolve docker.io/library/ubuntu:18.04@sha256:8fedbd5bd9fb72889c7bbca476949e18593cebed9b1fb9edf5b79dbbacc
=> == sha256:8fedbd5bd9fb72889c7bbca476949e18593cebed9b1fb9edf5b79dbbaccdd7d6 1.41kB / 1.41kB
=> == sha256:fc8d6af5ab38dab33aa53643c4c4b312c6d1f044c1a2229b2743b252b0689fc 529B / 529B
=> == sha256:5a214d77f5d747e6ed81632310baa0108301fweb875cf6bf9da560108f00072 1.46kB / 1.46kB
=> == sha256:284055322776031bac33723839acb0db2d063a525ba4fa1fd268a831c7553b26 26.71MB / 26.71MB
=> == extracting sha256:284055322776031bac33723839acb0db2d063a525ba4fa1fd268a831c7553b26
=> [internal] load build context
=> == transferring context: 1.41MB
=> https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
=> [ 2/15] RUN apt-get update
=> [ 3/15] RUN mkdir /workspace
=> [ 4/15] WORKDIR /workspace
=> [ 5/15] RUN apt-get update && apt-get install -y curl wget htop iotop tmux nano i
=> # Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
=> # Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
=> # Hit:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
=> # Hit:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
=> # Reading package lists...
=> # Reading package lists...
```

This will take a while to build. The build script is described in **Dockerfile**

To check that the image was built run: **docker images**

<path>|Desktop\docker_example_app\simple_app>docker images

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|-------------|-----|--------------|----------------|--------|
| simple_app2 | 0.0 | e4c5ae3f8ad9 | 43 seconds ago | 3.18GB |

- c. To save image as tar run: **docker save simple_app2:0.0 > simple_app_image2.tar**
(this is not necessary if you will be running this locally and not intend to share with others)