

## Conteneurisation d'application avec Docker

### Docker compose

#### Objectifs :

- Créer un service product exposant une API REST avec Flask pour fournir la liste des produits.
- Créer un service website en PHP/Apache qui consomme l'API Flask et affiche les produits dans un navigateur.
- Utiliser Docker et Docker Compose pour orchestrer les deux services et permettre l'accès via le navigateur.

#### Structure :

```
PS C:\Users\cyrin\docker_compose> tree /F
Structure du dossier pour le volume OSig  W Enable Watch
Le numéro de série du volume est A6B3-8183
C:..
├── docker-compose.yml
├── product
│   ├── api.py
│   ├── Dockerfile
│   └── requirements.txt
└── website
    └── index.php
```

#### Service 1 : API REST

Créer une API REST simple avec Flask dans Docker, et la rendre accessible depuis mon navigateur via <http://localhost:5000>

- requirements.txt

```
PS C:\Users\cyrin\docker_compose\product> cat requirements.txt
Flask==2.3.3
flask-restful==0.3.9
```

- Dockerfile pour construire l'image

```
PS C:\Users\cyrin\docker_compose\product> cat Dockerfile
FROM python:3.11-slim
WORKDIR /usr/src/app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
CMD ["python", "api.py"]
PS C:\Users\cyrin\docker_compose\product>
```

- api.py : (API simple avec un endpoint /)

```

$ docker.sh  api.py 1 X
C:\Users\cyryn> docker_compose > product > api.py > ...
1 from flask import Flask
2 from flask_restful import Api, Resource
3
4 #Initialisation de l'application
5 app = Flask(__name__)
6 api = Api(app)
7
8 class Product(Resource):
9     def get(self):
10         return {
11             'products': ['Ipad pro 14', 'Iphone 13', 'Ordinateur bureautique']
12         }
13
14 #Definition d'une route
15 api.add_resource(Product, '/')
16
17 #Execution de l'applicaction
18 if __name__ == '__main__':
19     app.run(host='0.0.0.0', port=80, debug=True)]

```

- Construction de l'image Docker :

```

PS C:\Users\cyryn\docker_compose\product> docker build -t product-api .
[+] Building 28.1s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.11-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> [internal] load build context
=> [1/5] FROM docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c
=> resolve docker.io/library/python:3.11-slim@sha256:193fdd0bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c4
=> sha256:1771569cc1299abc9cc762fc4419523e721b11a3927ef968ae63ba0a4a88f2da 251B / 251B
=> sha256:b3dd773c329649f22e467ae63d1c612a039a0559dec99fffb9ada904ab5c60c55 14.36MB / 14.36MB
=> sha256:22b63e76fde1e200371ed9f3cee91161d192063bcff65c9ab6bf63819810a974 1.29MB / 1.29MB
=> sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 29.78MB / 29.78MB
=> extracting sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db
=> extracting sha256:22b63e76fde1e200371ed9f3cee91161d192063bcff65c9ab6bf63819810a974
=> extracting sha256:b3dd773c329649f22e467ae63d1c612a039a0559dec99fffb9ada904ab5c60c55
=> extracting sha256:1771569cc1299abc9cc762fc4419523e721b11a3927ef968ae63ba0a4a88f2da
=> [internal] load build context
=> transferring context: 773B
=> [2/5] WORKDIR /usr/src/app
=> [3/5] COPY requirements.txt .
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> [5/5] COPY . .
=> exporting to image
=> exporting layers
=> exporting manifest sha256:574b438d71611701173c2c6a6a1b54dccc9fe5f72fb0fa298e8b5686b73a68fa4
=> exporting config sha256:5e4416eb3d3a7de1a924e1e24e1226d7cd4771e9a825cead0c74cd94a169ae7
=> exporting attestation manifest sha256:2fa713c11bf3fa214569163bbf663a459db6ab0f85fc9acbf7b60e0575ec29fd
=> exporting manifest list sha256:18980a793dc26a15e8a07e974965b11940ce9f76147d81e629c6212612d0e9de
=> naming to docker.io/library/product-api:latest
=> unpacking to docker.io/library/product-api:latest

```

- Vérification de la création de l'image

```

PS C:\Users\cyryn\docker_compose\product> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
product-api latest 18980a793dc2 About a minute ago 215MB
redis latest 43355e+d2249 2 weeks ago 202MB
mysql latest 569c4128dfa6 6 weeks ago 1.27GB
alpine latest 4b7ce07002c6 8 weeks ago 12.8MB
PS C:\Users\cyryn\docker_compose\product>

```

- Lancement du conteneur

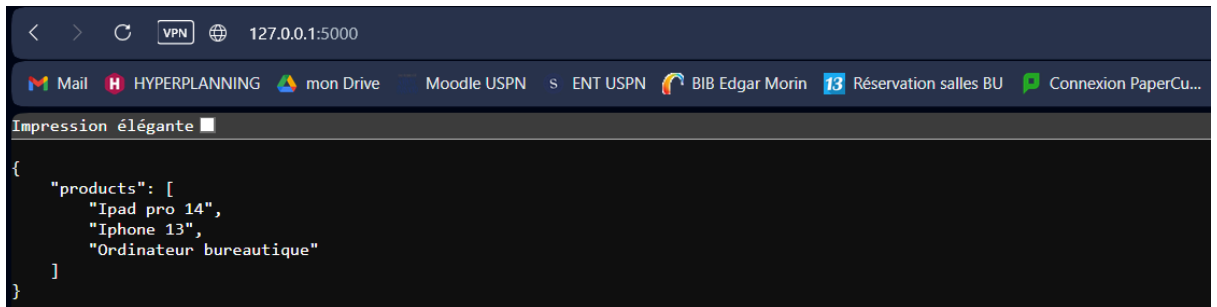
-p 5000:80 → mappe le port 5000 du PC au port 80 du conteneur.

```

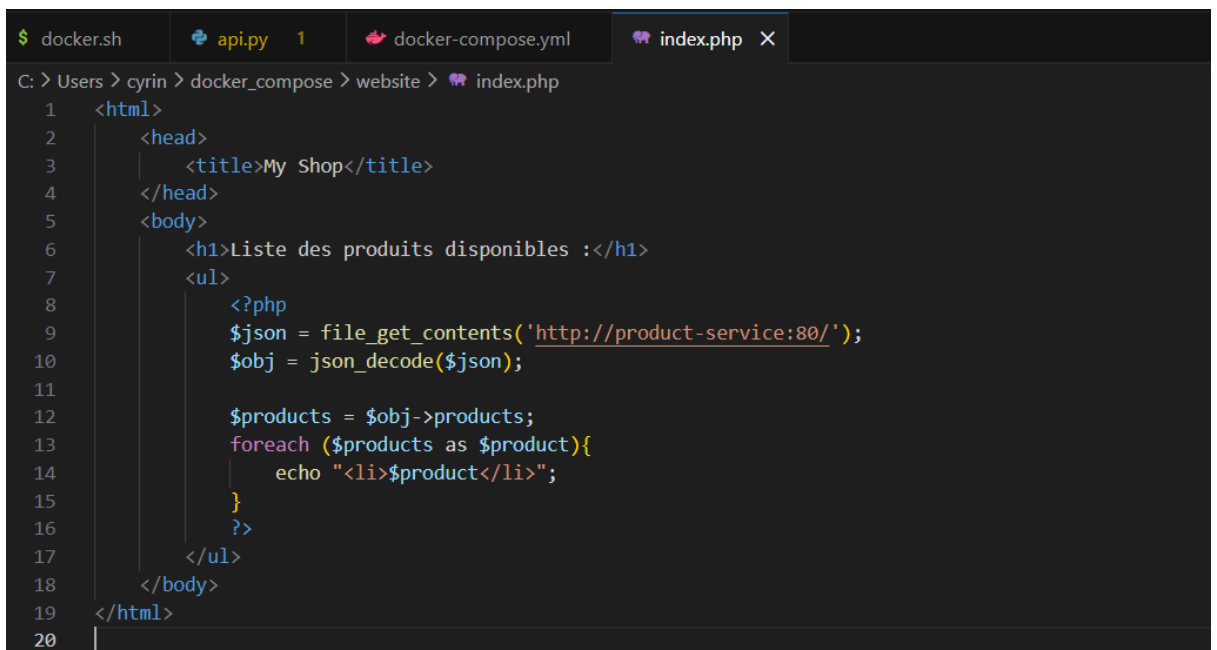
PS C:\Users\cyryn\docker_compose\product> docker run -p 5000:80 product-api
* Serving Flask app 'api'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on http://172.17.0.2:80
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 134-471-344
172.17.0.1 - - [05/Dec/2025 09:07:06] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [05/Dec/2025 09:07:07] "GET /favicon.ico HTTP/1.1" 404 -

```

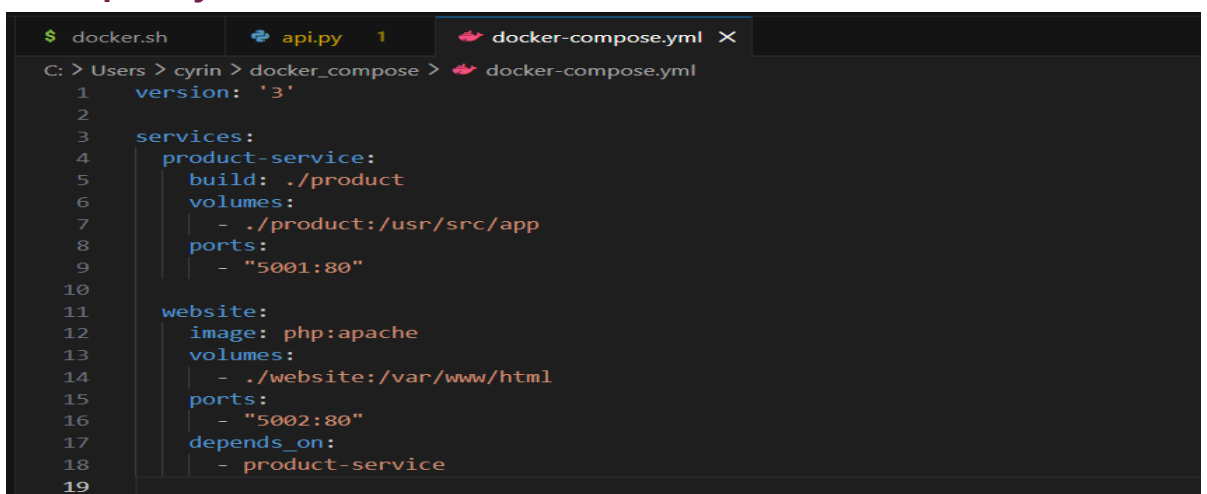
- Test dans le navigateur



## Service 2 : website (PHP/Apache)



## docker-compose.yml



- product-service :
  - Construite à partir du Dockerfile dans product/
  - Expose port 5001 sur l'hôte

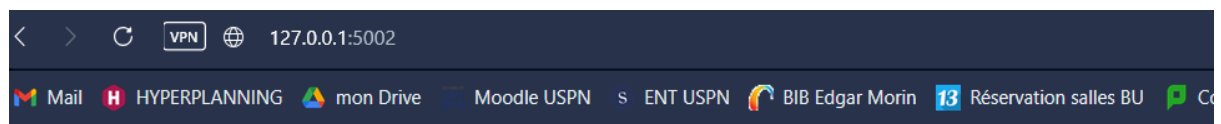
- website :
  - Image PHP/Apache officielle
  - Expose port 5002 sur l'hôte
- Dépend de product-service pour que Docker Compose démarre dans le bon ordre

## Lancer les services :

Dans le terminal, à la racine docker\_compose/

```
PS C:\Users\cyrin\docker_compose> docker compose up --build
time="2025-12-05T18:33:43+01:00" level=warning msg="C:\Users\cyrin\docker_compose\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[*] Running 12/16
✓website Pulled                                38.1s
✓eebe1c500b61 Pull complete                    0.6s
✓011574101943 Pull complete                    34.2s
✓cf8c6e80bbad Pull complete                   33.9s
✓0c4dd52a7946 Pull complete                   35.5s
✓57802a2d3c3c Pull complete                   0.7s
✓61b5a6c1c05c Pull complete                   35.7s
✓0baeb349142a Pull complete                   0.7s
✓a688ee4a13ca Pull complete                   0.4s
✓3976b59fc129 Pull complete                   0.9s
✓4f4fb700ef54 Pull complete                   0.1s
✓7d887b4d771c Pull complete                   33.8s
✓1c8df7829445 Pull complete                   34.3s
✓66e6689a6579 Pull complete                   34.4s
[*] Building 2.5s (13/13) FINISHED
=> [internal] load local base definitions                                0.1s
=> reading from stdin 5865                                             0.1s
=> [internal] load build definition from Dockerfile                    0.1s
=> transferring dockerfile: 196B                                       0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim    1.1s
=> [auth] library/python:pull token for registry-1.docker.io          0.0s
=> [internal] load .dockerignore                                        0.0s
=> transferring context: 2B                                             0.0s
=> [internal] load build context                                       0.0s
=> transferring context: 93B                                            0.0s
=> [1/5] FROM docker.io/library/python:3.11-slim@sha256:193fdd8bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c4747444d  0.1s
=> resolve docker.io/library/python:3.11-slim@sha256:193fdd8bbcb3d2ae612bd6cc3548d2f7c78d65b549fcaa8af75624c4747444d  0.1s
=> CACHED [2/5] WORKDIR /usr/src/app                                  0.0s
=> CACHED [3/5] COPY requirements.txt                                  0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt     0.0s
=> CACHED [5/5] COPY . .                                              0.0s
=> exporting to image                                                  0.3s
=> exporting layers                                                    0.0s
=> exporting manifest sha256:04a583ae05a9e7269a75acalbfa9dc77405cb413a334a42696ede01e22c3511  0.0s
=> exporting config sha256:f5c458caa32571099269e851b9e001b894775e6d30fad476dc57eb16733d6f35  0.0s
=> exporting attestation manifest sha256:854f41b1d6022d76825d5ed1105f363da16667c84a5cb6900d57190ca22bfb3b  0.1s
=> exporting manifest list sha256:acd6dfcb3a040804964ce73eac1ce2315e4919c5c4508395b405352db3500cce  0.0s
=> naming to docker.io/library/docker_compose-product-service:latest  0.0s
=> unpacking to docker.io/library/docker_compose-product-service:latest  0.0s
=> resolving provenance for metadata file                               0.1s
[*] Running 4/4
✓docker_compose-product-service Built                                0.0s
✓Network docker_compose_default Created                             0.3s
✓Container docker_compose-product-service-1 Created                 0.7s
✓Container docker_compose-website-1 Created                         0.4s
Attaching to product-service-1, website-1
product-service-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message
product-service-1 | * Serving Flask app 'api'
product-service-1 | * Debug mode: on Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directi
website-1 | [Fri Dec 05 09:34:29.682069 2025] [mpm_prefork:notice] [pid 1:tid 1] AH00163: Apache/2.4.65 (Debian) PHP/8.5.0 configured -- resuming normal operations
product-service-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
website-1 | [Fri Dec 05 09:34:29.682192 2025] [core:notice] [pid 1:tid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
product-service-1 | * Running on all addresses (0.0.0.0)
product-service-1 | * Running on http://127.0.0.1:80
product-service-1 | * Running on http://172.18.0.2:80
product-service-1 | Press CTRL-C to quit
product-service-1 | * Restarting with stat
product-service-1 | * Debugger is active!
product-service-1 | * Debugger PIN: 104-278-068
```

- Docker Compose construira l'image Flask et démarrera les deux services.
- Les services seront accessibles depuis le navigateur :
  - <http://localhost:5001> → API Flask (JSON)
  - <http://localhost:5002> → Site PHP affichant la liste des produits



## Liste des produits disponibles :

- Ipad pro 14
- Iphone 13
- Ordinateur bureautique