

Le démarrage les différents conteneurs d'une application

```
Transaction de test réussie.
Exécution de la transaction
  Préparation      :                               1/1
  Installation     : epel-release-9-4.el9.noarch  1/1
  Exécution du scriptlet: epel-release-9-4.el9.noarch 1/1
Many EPEL packages require the CodeReady Builder (CRB) repository.
It is recommended that you run /usr/bin/crb enable to enable the CRB repository.

  Vérification de   : epel-release-9-4.el9.noarch  1/1

Installé:
  epel-release-9-4.el9.noarch

Terminé !
```

Installation de #yum install -y epel-release

```
Installer 1 Paquet

Taille totale des téléchargements : 1.7 M
Taille des paquets installés : 8.7 M
Téléchargement des paquets :
python3-pip-21.2.3-6.el9.noarch.rpm          1.1 MB/s | 1.7 MB    00:01
-----
Total                                         939 kB/s | 1.7 MB    00:01
Test de la transaction
La vérification de la transaction a réussi.
Lancement de la transaction de test
Transaction de test réussie.
Exécution de la transaction
  Préparation      :                               1/1
  Installation     : python3-pip-21.2.3-6.el9.noarch 1/1
  Exécution du scriptlet: python3-pip-21.2.3-6.el9.noarch 1/1
  Vérification de   : python3-pip-21.2.3-6.el9.noarch 1/1

Installé:
  python3-pip-21.2.3-6.el9.noarch

Terminé !
```

```
#yum install -y python3-pip
```

```
[root@SRV-DOCKER-9 user]# docker-compose --version
docker-compose version 1.29.2, build unknown
```

```
$docker-compose --version
```

```
[root@SRV-DOCKER-9 user]# mkdir monappliweb
[root@SRV-DOCKER-9 user]# cd monappliweb/
[root@SRV-DOCKER-9 monappliweb]#
```

```
$mkdir monappliweb
```

```
$cd monappliweb
```

```
Running setup.py install for docopt ... done
Running setup.py install for dockerpty ... done
Successfully installed attrs-23.1.0 bcrypt-4.0.1 certifi-2023.5.7 cffi-1.15.1 charset-normalizer-3.1.0 click-8.1.7 cryptography-41.0.0 dockerpty-0.4.1 dockerpty-ptyprocess-0.5.0 idna-3.4 ipaddress-1.0.5 itsdangerous-2.1.2 Jinja2-3.1.2 MarkupSafe-2.1.3 packaging-23.1 paramiko-3.1.0 pycparser-2.21 pynacl-1.5.0 pyrsistent-0.19.3 python-dotenv-0.21.1 python-jinja2-3.1.2 python-paramiko-3.1.0 python-pycparser-2.21 python-pynacl-1.5.0 python-pyrsistent-0.19.3 python-pysocks-1.7.1 python-requests-2.31.0 python-urllib3-2.0.4 pysocks-1.7.1 pytz-2023.3 requests-2.31.0 urllib3-2.0.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the venv library, which is recommended for pip to avoid interference with system packages. See https://pip.pypa.io/warnings/venv
```

```
GNU nano 5.6.1 docker-compose.yml
version: "3"
services:
  phpmyadmin:
    image: netroby/alpine-phpmyadmin
    ports:
      - "8080:80"
    environment:
      - MYSQL_ROOT_PASSWORD=joliverie
    networks:
      - NetPhpMyAdmin
  db:
    image: mariadb
    ports:
      - "3306:3306"
    environment:
      - MYSQL_ROOT_PASSWORD=joliverie
    networks:
      - NetPhpMyAdmin
networks:
  NetPhpMyAdmin:
```

Création et configuration du fichier docker-compose.yml dans le dossier monappliweb

```
[root@SRV-DOCKER-9 monappliweb]# docker-compose ps

```

Name	Command	State
monappliweb_db_1	docker-entrypoint.sh mariadb	Up
monappliweb_phpmyadmin_1	php -S 0.0.0.0:8080 -t /www/	Up

```
tcp
8080/tcp
```

Quelle commande entrez-vous pour retrouver le nouveau nom des conteneurs de votre applications ? La commande qu'on entre pour retrouver le nouveau nom des conteneurs est docker-compose ps

```
[root@SRV-DOCKER-9 monappliweb]# docker-compose config
networks:
  NetPhpMyAdmin: {}
services:
  db:
    environment:
      MYSQL_ROOT_PASSWORD: joliverie
    image: mariadb
    networks:
      NetPhpMyAdmin: null
    ports:
      - published: 3306
        target: 3306
  phpmyadmin:
    environment:
      MYSQL_ROOT_PASSWORD: joliverie
    image: netroby/alpine-phpmyadmin
    networks:
      NetPhpMyAdmin: null
    ports:
      - published: 8080
        target: 80
version: '3'
```

Quelle commande entrez-vous pour retrouver les services de votre applications ? Docker-compose config

Stack details ↻





Stack

Information

! This stack was created outside of Portainer. Control over this stack is limited.

Stack details

monappliweb



Containers

Q Search...

▶ Start

□ Stop

⊘ Kill

↻ Restart

⏸ Pause

▶ Resume

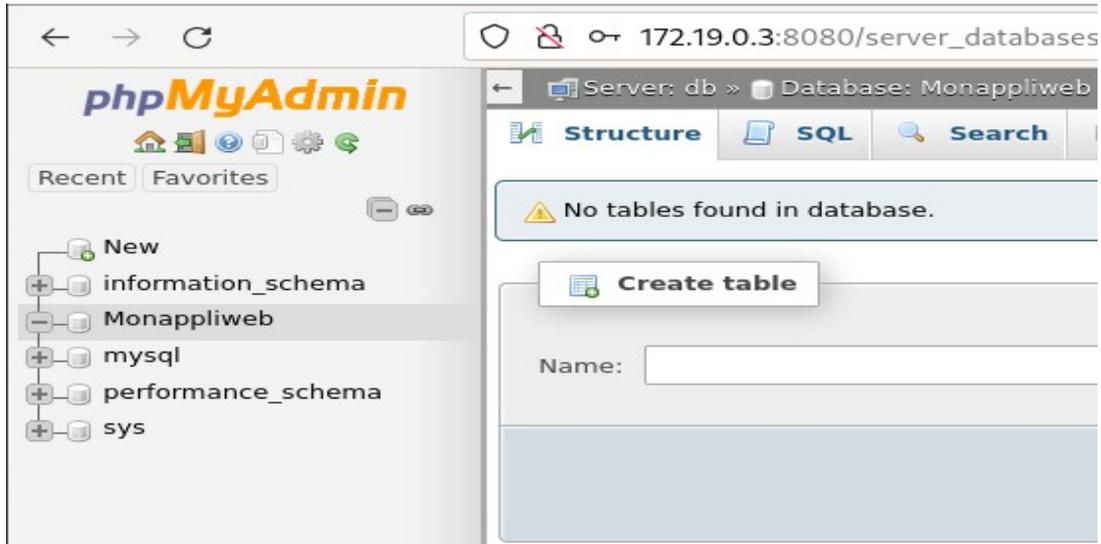
🗑 Remove

<input type="checkbox"/> Name ↓↑	State ↓↑ Filter ▼	Quick Actions	Stack ↓↑	Image ↓↑
<input type="checkbox"/> monappliweb_db_1	running	📄 🕒 ⏸ ▶ 🗑	monappliweb	mariadb
<input type="checkbox"/> monappliweb_phpmyadmin_1	running	📄 🕒 ⏸ ▶ 🗑	monappliweb	netroby/alpine-phpmyadmin

Que représente la colonne stack dans la vue conteneurs de Portainer ? La colonne stack dans la vue conteneurs de Portainer représente le dossier correspondant à l'emplacement des conteneurs

Quick Actions	Stack ↓↑	Image ↓↑	Created ↓↑	IP Address ↓↑	GPUs	Published Ports
📄 🕒 ⏸ ▶ 🗑	monappliweb	mariadb	2023-05-11 14:33:16	172.19.0.2	none	🔗 3306:3306
📄 🕒 ⏸ ▶ 🗑	monappliweb	netroby/alpine-phpmyadmin	2023-05-11 14:33:16	172.19.0.3	none	🔗 8080:80

Quelles sont les adresses IP internes des 2 conteneurs ?
 172.19.0.2 pour mariadb
 et 172.19.0.3 pour phpmyadmin



En utilisant l'adresse IP interne, créer une nouvelle table dans votre base de données avec phpmyadmin.

```
[root@SRV-DOCKER-9 monappliweb]# docker-compose down
Stopping monappliweb_phpmyadmin_1 ...
Stopping monappliweb_db_1          ... done
```

Arrêt de l'application avec la commande docker-compose down

Quelle différence faites-vous entre docker-compose up et docker-compose start ? La différence est que docker-compose up créer, recréer et s'attache à des conteneurs alors que docker-compose start démarre les conteneurs existants.

Quelle différence faites-vous entre docker-compose down et docker-compose stop ? La différence est que docker-compose down arrête et élimine les conteneurs alors que docker-compose stop arrête seulement les conteneurs et peuvent être redémarrés avec docker-compose start.