



# How to install testcontainers on Ubuntu wsl

# How to install testcontainers on Ubuntu wsl

## Prerequisites

Avoir installé :

- [WSL2](#)
- [Ubuntu](#)
- [Docker](#)

## Contexte

Pouvoir exécuter les tests unitaires de la couche DAO, qui utilisent le framework [Testcontainers](#) .

Ces tests, via Testcontainers, créent et utilisent des conteneurs Docker pour disposer d'une base de données PostgreSQL.

L'objectif est de permettre à Testcontainers de créer ces conteneurs dans une VM Ubuntu, préalablement créée sous WSL2.

## Configuration de Docker et Testcontainers dans la VM Ubuntu sous WSL2

### Se connecter à la VM Ubuntu WSL

Depuis un terminal Windows, ouvrir une connexion sur la VM Ubuntu WSL :

```
C:\> wsl -d Ubuntu
```

Se déplacer dans le dossier de l'utilisateur par défaut à l'aide de la commande suivante :

```
$ cd
```

### Récupérer les images Docker de base

Ces images sont nécessaires à la création des conteneurs utilisés par Testcontainers.

Récupérer l'image de base pour la *BDD PostgreSQL* :

```
$ docker pull nexus-ext.2moro.fr:8088/2moro/postgres:latest
```

Récupérer l'image utilitaire pour Testcontainers :

```
$ docker pull testcontainersofficial/ryuk:0.3.0
```

## Modifier la configuration du service docker

Afin que TestContainers puisse communiquer avec *docker*, il a besoin que ce dernier écoute sur le port 2375.

Par conséquent, il faut modifier la configuration du service *docker* comme suit, en y ajoutant le bloc *hosts* :

```
$ sudo nano /etc/docker/daemon.json
```

```
{
  "insecure-registries": [
    "nexus-ext.2moro.fr:8088"
  ],
  "hosts": [
    "tcp://127.0.0.1:2375", "unix:///var/run/docker.sock"
  ]
}
```

## Modifier le service docker

Parallèlement, il faut aussi modifier le fichier qui gère le service *docker* :

```
$ sudo nano /lib/systemd/system/docker.service
```

Puis, modifier la ligne `ExecStart` comme suit :

```
ExecStart=/usr/bin/dockerd --containerd=/run/containerd/containerd.sock
```

## Modifier la configuration de Testcontainers

Maintenant, la configuration de Testcontainers doit être modifiée afin qu'il sache comment contacter le service *docker* :

```
$ nano $HOME/.testcontainers.properties
```

### Info

Ce fichier `$HOME/.testcontainers.properties` peut ne pas exister. Avec la commande donnée ci-dessus en exemple, il sera automatiquement créé.

S'assurer que le contenu du fichier corresponde à ce qui suit :

```
docker.client.strategy =
org.testcontainers.dockerclient.EnvironmentAndSystemPropertyClientProviderStrategy
docker.host = tcp://localhost:2375
docker.cert.path = /home/<utilisateur>/.docker
ryuk.container.privileged = false
```

 **Remark**

Dans le fichier ci-dessus, la chaîne "`<utilisateur>`" est à remplacer par la valeur retournée par la commande suivante :

```
$ whoami
```

### Redémarrer le service *docker*

Afin que toutes ces modifications soient bien prises en compte, il faut demander à `systemd` de recharger sa configuration, puis redémarrer le service `docker` :

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
```