

## TP Docker : Les types de réseaux

Docker propose 4 modes de réseau principaux :

1. bridge (par défaut)
2. host
3. none
4. réseau personnalisé (user-defined bridge)

⇒ Chaque mode contrôle comment les conteneurs communiquent entre eux et avec l'extérieur.

### 1) Mode Bridge (par défaut)

C'est le réseau **standard** que Docker utilise quand on ne précise rien :

```
PS C:\Users\cyrin> docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
de57a609c9d5: Pull complete
53d743880af4: Pull complete
192e2451f875: Pull complete
77fa2eb06317: Pull complete
108ab8292820: Pull complete
b5feb73171bf: Pull complete
Digest: sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42
Status: Downloaded newer image for nginx:latest
```

→ Docker met automatiquement le conteneur dans le réseau **bridge**.

Vérification :

```
PS C:\Users\cyrin> docker network inspect bridge
[{"Name": "bridge", "Id": "8041594f6b444ef15953b2e7030f1996c35bcd1015f8be9f3a8793c08329fc41", "Created": "2025-12-05T08:32:30.249877203Z", "Scope": "local", "Driver": "bridge", "EnableIPv4": true, "EnableIPv6": false, "IPAM": {"Driver": "default", "Options": null, "Config": [{"Subnet": "172.17.0.0/16", "Gateway": "172.17.0.1"}]}, "Internal": false, "Attachable": false, "Ingress": false, "ConfigFrom": {"Network": ""}}
```

- Chaque conteneur reçoit une **IP privée** (ex : 172.17.0.2)
- Les conteneurs du bridge peuvent communiquer entre eux
- Pour accéder depuis l'extérieur, il faut faire un **port mapping** :

```
PS C:\Users\cyrin> docker run -d -p 8080:80 nginx
9a56dcdfbed0b60ca2fa551a6fe6e20fc3b6a211f72c185272dc7d52b3cd90f4
```

⇒ C'est le réseau le plus utilisé pour des applications normales.

## 2) Mode HOST

Ce mode fonctionne uniquement sur Linux. Sous Windows et Mac (Docker Desktop), ce mode ne fonctionne pas correctement, car Docker tourne dans une VM.

```
PS C:\Users\cyrin> docker run --network host nginx
```

- Le conteneur utilise directement le réseau de la machine
- Pas besoin de -p, car il n'y a pas d'isolation réseau
- Peu utilisé car réduit la sécurité

## 3) Mode NONE

Isolément réseau total.

```
PS C:\Users\cyrin> docker run --network none alpine sleep 1000
```

→ Le conteneur n'a aucun réseau, aucune interface, pas d'accès Internet, pas de communication.

Ce mode est utilisé pour :

- tests de sécurité
- environnements totalement isolés
- conteneurs très contrôlés

## 4) Réseau personnalisé (User-defined bridge)

C'est un réseau créé par l'utilisateur.

Avantages :

- Les conteneurs se voient par leur nom
- Plus sécurisé que le bridge par défaut
- Très utilisé en Docker Compose

- Création du réseau :

```
PS C:\Users\cyrin> docker network create monreseau
83f5fc752206b7f3850100153031bd626b6b8afdf9aa6ca7f70950fca62dbc
```

- Lancer deux conteneurs dans le même réseau :

```
PS C:\Users\cyrin> docker run -d --name site --network monreseau nginx
011502f738341e8ffd85dbbecf28542d1cba5377897beb305b55c3b61ca01069
PS C:\Users\cyrin> docker run -d --name api --network monreseau alpine sleep 1000
2fb91e451e4f60836b000e2b49654b674faaee393fb2a9f8fecca9cc4d9a3a3
```

- Tester la communication entre conteneurs :

```
PS C:\Users\cyrin> docker exec -it web sh
```

- Supprimer un conteneur

```
PS C:\Users\cyrin> docker rm -f web
```

- Supprimer un réseau

```
PS C:\Users\cyrin> docker network rm monreseau
```