

# Labo 5 - Le réseau, partie 1

Pour voir comment se comporte le réseau entre les conteneurs, la machine hôte et le réseau local, nous allons créer un conteneur avec le serveur web nginx

```
sudo docker run --rm nginx
```

```
[nancre@DOCKER ~]$ sudo docker run --rm nginx
[sudo] password for nancre:
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
bb79b6b2107f: Pull complete
5a9f1c0027a7: Pull complete
b5c20b2b484f: Pull complete
166a2418f7e8: Pull complete
1966ea362d23: Pull complete
Digest: sha256:aeade65e99e5d5e7ce162833636f692354c227ff438556e5f3ed0335b7cc2f1b
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
```

On voit que le conteneur est créé et en cours d'exécution. Le port 80 est ouvert mais uniquement sur le conteneur. Il va falloir faire un mappage de port pour accéder au serveur nginx du conteneur à travers le navigateur web

Pour ce faire il faut utiliser la commande précédente et rajouter l'option `-p` avec le port de l'hôte 80 et celui du conteneur 80

```
sudo docker run --rm -p 80:80 nginx
```

```
[nancre@DOCKER ~]$ sudo docker run --rm -p 80:80 nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
```

Si nous ouvrons un navigateur avec l'adresse ip du conteneur, nous arrivons sur nginx



Attention, il n'est pas possible d'ouvrir le même port hôte pour un autre conteneur

Il est également possible de laisser l'hôte choisir son port

```
sudo docker run --rm 80 nginx
```

Puis sur un autre terminal de la machine hôte :

```
sudo netstat -nate
```

```
[nancre@DOCKER ~]$ netstat -nate
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User        Inode
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      0           25832
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      0           26532
tcp        0      0 192.168.1.220:35550     62.138.222.4:80        TIME_WAIT  0           0
tcp        0      64 192.168.1.220:22        192.168.1.202:60620    ESTABLISHED 0           298810
tcp        0      0 192.168.1.220:22        192.168.1.202:49784    ESTABLISHED 0           168257
tcp6       0      0 :::22                   :::*                     LISTEN      0           25834
tcp6       0      0 :::1:25                  :::*                     LISTEN      0           26533
tcp6       0      0 :::32768                 :::*                     LISTEN      0           300365
[nancre@DOCKER ~]$
```

On voit que le port 32768 est apparu. Nginx sera donc accessible sur votre navigateur via <http://votreadresseip:32768>

Nous pouvons également trouver le port aléatoire créé par l'hôte avec la commande suivante depuis la machine hôte

```
sudo docker inspect lenomdevotreconteneur
```

```

"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "889e7ef9d52115da93161613041390463cb764e4e7ae3cd49b64a39bf91cb3c2",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "80/tcp": [
      {
        "HostIp": "0.0.0.0",
        "HostPort": "32768"
      }
    ]
  },
  "SandboxKey": "/var/run/docker/netns/889e7ef9d521",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "4eb28e8850add2e65cd517a6c0e11d8089f40ee34b9ef4720d2443555ca66481",
  "Gateway": "172.17.0.1",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "172.17.0.2",
  "IPPrefixLen": 16,
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:02",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "d9334d3a443ebd9a800252f615b340f75db02f15fba51fa2da77d8aecce61958",
      "EndpointID": "4eb28e8850add2e65cd517a6c0e11d8089f40ee34b9ef4720d2443555ca66481",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:02",
      "DriverOpts": null
    }
  }
}
]
[nancre@DOCKER ~]$ █

```

Nous pouvons voir également quels sont les ports exposés par une image d'un conteneur :

```
sudo docker inspect nginx
```

```
[nancre@DOCKER ~]$ sudo docker inspect nginx:latest
[
  {
    "Id": "sha256:c39a868aad02a383c7e490e0fc4a5b0217f667f2de764bc2755e315a5adf64a1",
    "RepoTags": [
      "nginx:latest"
    ],
    "RepoDigests": [
      "nginx@sha256:aeade65e99e5d5e7ce162833636f692354c227ff438556e5f3ed0335b7cc2f1b"
    ],
    "Parent": "",
    "Comment": "",
    "Created": "2020-11-05T18:21:10.923474978Z",
    "Container": "16434ccbe2eab08b0e7e63149c390b8a346c8cf92899ba6063c1b2653628f3c1",
    "ContainerConfig": {
      "Hostname": "16434ccbe2ea",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "ExposedPorts": {
        "80/tcp": {}
      }
    },
  }
]
```

---

Revision #1

Created 2026-04-24 13:16:18 UTC by Nico là

Updated 2026-04-24 13:16:30 UTC by Nico là