
docker-network-inspect

Inspecter un réseau

Retourne des informations sur un ou plusieurs réseaux. Par défaut, cette commande affiche le résultat dans un objet JSON. par exemple, si 2 conteneurs sont connectés à un réseau bridge :

```
sudo docker run -itd --name=container1 busybox  
f2870c98fd504370fb86e59f32cd0753b1ac9b69b7d80566ffc7192a82b3ed27  
sudo docker run -itd --name=container2 busybox  
bda12f8922785d1f160be70736f26c1e331ab8aaf8ed8d56728508f2e2fd4727
```

La commande inspect affiche les conteneurs, par id :

```
sudo docker network inspect bridge  
[  
  {  
    "Name": "bridge",  
    "Id": "b2b1a2cba717161d984383fd68218cf70bbbd17d328496885f7c921333228b0f",  
    "Scope": "local",  
    "Driver": "bridge",  
    "IPAM": {  
      "Driver": "default",  
      "Config": [  
        {  
          "Subnet": "172.17.42.1/16",  
          "Gateway": "172.17.42.1"  
        }  
      ]  
    },  
    "Containers": {  
      "bda12f8922785d1f160be70736f26c1e331ab8aaf8ed8d56728508f2e2fd4727": {  
        "Name": "container2",  
        "EndpointID": "0aebb8fcd2b282abe1365979536f21ee4ceaf3ed56177c628eae9f706e00e019",  
        "MacAddress": "02:42:ac:11:00:02",  
        "IPv4Address": "172.17.0.2/16",  
        "IPv6Address": ""  
      },  
      "f2870c98fd504370fb86e59f32cd0753b1ac9b69b7d80566ffc7192a82b3ed27": {  
        "Name": "container1",  
        "EndpointID": "a00676d9c91a96bbe5bcfb34f705387a33d7cc365bac1a29e4e9728df92d10ad",  
        "MacAddress": "02:42:ac:11:00:01",  
        "IPv4Address": "172.17.0.1/16",  
        "IPv6Address": ""  
      }  
    },  
    "Options": {  
      "com.docker.network.bridge.default_bridge": "true",  
      "com.docker.network.bridge.enable_icc": "true",  
      "com.docker.network.bridge.enable_ip_masquerade": "true",  
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",  
      "com.docker.network.bridge.name": "docker0",  
      "com.docker.network.driver.mtu": "1500"  
    }  
  }  
]
```

Retourner les informations sur le réseau utilisateur :

```
docker network create simple-network
69568e6336d8c96bbf57869030919f7c69524f71183b44d80948bd3927c87f6a
docker network inspect simple-network
[
  {
    "Name": "simple-network",
    "Id": "69568e6336d8c96bbf57869030919f7c69524f71183b44d80948bd3927c87f6a",
    "Scope": "local",
    "Driver": "bridge",
    "IPAM": {
      "Driver": "default",
      "Config": [
        {
          "Subnet": "172.22.0.0/16",
          "Gateway": "172.22.0.1/16"
        }
      ]
    },
    "Containers": {},
    "Options": {}
  }
]
```

OPTIONS

-f, --format="" Formate la sortie en utilisant le template Go