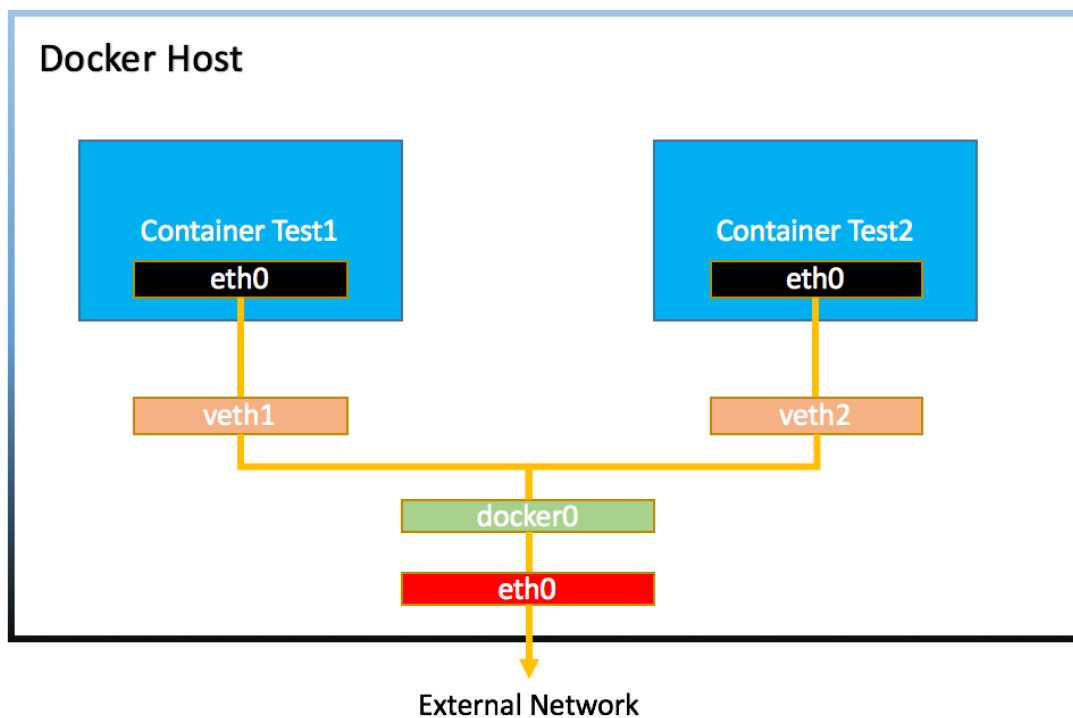




Docker network:

Basically in a docker container the every container has the unique ip address and it not be accessible outside the local system so to connect container outer side of word the one concept use that's the **docekr0 or docker bridge**.



What is docker0:

docker0 is a virtual bridge interface created by Docker. It randomly chooses an address and subnet from a private defined range. All the Docker containers are connected to this bridge and use the NAT rules created by docker to communicate with the outside world.

docker0 is a bridge. Every container using that bridge will have a local ip address. You can't use that to access to host from outside that machine

Using ip a we check that:

```
ubuntu $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:ed:97:ae brd ff:ff:ff:ff:ff:ff
    inet 172.30.1.2/24 brd 172.30.1.255 scope global dynamic enp1s0
        valid_lft 86313390sec preferred_lft 86313390sec
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:02:cf:54:46 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
ubuntu $
```

Using docker inspect command we check the ip address container

```
ubuntu $ docker run -d docker.io/openshift/hello-openshift
Unable to find image 'openshift/hello-openshift:latest' locally
latest: Pulling from openshift/hello-openshift
4f4fb700ef54: Pull complete
8b32988996c5: Pull complete
Digest: sha256:aaea76ff622d2f8bcb32e538e7b3cd0ef6d291953f3e7c9f5556c1ba5baf47e2e
Status: Downloaded newer image for openshift/hello-openshift:latest
cd64db44315af64952d362f0e8a9e22cdb3a964fec72f5947957ad4b76ecbfa
ubuntu $ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
cd64db44315a   openshift/hello-openshift          "/hello-openshift"      6 seconds ago Up 5 seconds   8080/tcp, 888
8/tcp         zealous_johnson
ubuntu $ docker inspect zealous_johnson
[
  {
    "Id": "cd64db44315af64952d362f0e8a9e22cdb3a964fec72f5947957ad4b76ecbfa",
    "Created": "2024-05-25T18:03:55.376520191Z",
    "Path": "/hello-openshift",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 3477,
      "ExitCode": 0,
```

Access the container, on host machine We not access it outside.

```
Editor  Tab 1  + 55 min
    "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": "dba427be9e92cb8379906bdfec25a76485ab52d487c47735e9bf1611de5ff7b7",
        "EndpointID": "5216a6275416eb2a2596fdefbe3aa3dfc103f433567df7cde91cab4e52c22b07",
        "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
      }
    }
  }
}
]
ubuntu $ curl 172.17.0.2:8080
Hello OpenShift!
ubuntu $
```

When the container on same host system they communicate each other

- In it just deploy 2 container and curl 1 container from 2 container

```
ubuntu $ docker run -dit --name=test httpd bash
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
09f376ebb190: Pull complete
dab55b4abfc3: Pull complete
4f4fb70ef54: Pull complete
1a6d0283f224: Pull complete
1abf9110528c: Pull complete
7bacb8f85f3a: Pull complete
Digest: sha256:43c7661a3243c04b0955c81ac994ea13a1d8a1e53c15023a7b3cd5e8bb25de3c
Status: Downloaded newer image for httpd:latest
9f2dfa57a978189c7339798aed139f77bf931398704373fd3eb53bdd5223b13f
ubuntu $ docker run -dit --name=test2 nginx bash
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
09f376ebb190: Already exists
a11fc495bafd: Pull complete
933cc8470577: Pull complete
999643392fb7: Pull complete
971bb7f4fb12: Pull complete
45337c09cd57: Pull complete
de3b062c0af7: Pull complete
Digest: sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72af059c
Status: Downloaded newer image for nginx:latest
6f3be1a2b844d5c07217d232c2fce87900a35591cccc475a20da6d4dbc827bcb
```

```
ubuntu $ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    e784f4560448   3 weeks ago    188MB
httpd         latest    356125da0595   7 weeks ago    147MB
ubuntu $ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
6f3be1a2b844   nginx     "/docker-entrypoint...." 44 seconds ago Up 43 seconds  80/tcp         test2
9f2dfa57a978   httpd     "bash"                 About a minute ago Up About a minute  80/tcp         test
ubuntu $
```

Through inspect command we get the ip address just curl in another container.

```
ubuntu $ docker inspect test1
[
  {
    "Id": "d2d2462ad4536ba98d40b0d9ace33a1039d82120da062126cbaa8a5c77fd3102",
    "Created": "2024-05-25T18:25:17.652982009Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 3208,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-05-25T18:25:17.950113178Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:e784f4560448b14a66f55c26e1b4dad2c2877cc73d001b7cd0b18e24a700a070",
    "ResolvConfPath": "/var/lib/docker/containers/d2d2462ad4536ba98d40b0d9ace33a1039d82120da062126cbaa8a5c77fd3102/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/d2d2462ad4536ba98d40b0d9ace33a1039d82120da062126cbaa8a5c77fd3102/hostname",
```

```
ubuntu $ docker exec -it test1 bash
root@678cd6e3ea9b:/usr/local/apache2# cd
root@678cd6e3ea9b:~# curl 172.17.0.3
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@678cd6e3ea9b:~#
```

To access the container outside the host machine, we required a port forwarding.

➤ Using command `ip a` check ip addresses

```
Editor  Tab 1  + 59 min
ubuntu $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:02:f5:97 brd ff:ff:ff:ff:ff:ff
    inet 172.30.1.2/24 brd 172.30.1.255 scope global dynamic enp1s0
        valid_lft 86311681sec preferred_lft 86311681sec
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:43:4a:b3:3e brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
ubuntu $
```

➤ Using `ss -tunlp | grep <PORT_NO>` check port is attach or not.

```
ubuntu $ ss -tunlp | grep 18080
ubuntu $ ss -tunlp | grep 22
tcp    LISTEN  0      511             0.0.0.0:40205   0.0.0.0:*      users:(("node",pid=1221,fd=18))
tcp    LISTEN  0      128             0.0.0.0:22     0.0.0.0:*      users:(("sshd",pid=659,fd=3))
tcp    LISTEN  0      128             [::]:22       [::]:*         users:(("sshd",pid=659,fd=4))
ubuntu $
```


Forward the port using -p.

➤ Check the result enter a machine ip with port.

```
ubuntu $ docker run -d --name=c1 -p 18080:80 docker.io/httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
09f376ebb190: Pull complete
dab55b4abfc3: Pull complete
4f4fb70ef54: Pull complete
1a6d0283f224: Pull complete
1abf9110528c: Pull complete
7bacb8f85f3a: Pull complete
Digest: sha256:43c7661a3243c04b0955c81ac994ea13a1d8a1e53c15023a7b3cd5e8bb25de3c
Status: Downloaded newer image for httpd:latest
7cce233a2ff8074a861ff0ec1506c10acec9a7d8b6e4b390a5c2ed0f6976af41
ubuntu $ curl 172.30.1.2:18080
<html><body><h1>It works!</h1></body></html>
ubuntu $ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
7cce233a2ff8   httpd     "httpd-foreground"      59 seconds ago Up 58 seconds 0.0.0.0:18080->80/tcp, :::18080->80/tcp
ubuntu $
```

Custom Ports

1808	Access
------	--------



It works!

Task: deploy super-mario game container and access it by port forwarding.

➤ Run super mario game container.

```
ubuntu $ docker run -d -p 28080:8080 pengbai/docker
-supermario
Unable to find image 'pengbai/docker-supermario:latest' locally
latest: Pulling from pengbai/docker-supermario
092586df9206: Pull complete
ef599477fae0: Pull complete
4530c6472b5d: Pull complete
d34d61487075: Pull complete
272f46008219: Pull complete
5cb62ceb26af: Pull complete
663538d2581c: Pull complete
d226fa7f84b7: Pull complete
9efb6b4ae4f4: Pull complete
77d53852f858: Pull complete
32af9fc740ac: Pull complete
36d34cc6e970: Pull complete
f20dfa6a3a87: Pull complete
Digest: sha256:8d909e68fbc38de1dc6bef982e0ca8c364f3c8021525c9487cec4374a53f15f5
Status: Downloaded newer image for pengbai/docker-supermario:latest
f4bfde0e87bc7a2b0d66da76f7e38f65dc826f296c1ad5a9ca5f0610b7e4bfda
ubuntu $ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
f4bfde0e87bc	pengbai/docker-supermario	"catalina.sh run"	8 seconds ago	Up 7 seconds	0.0.0.0:28080->8080/tcp, :::28080->8080/tcp	
7cce233a2ff8	httpd	"httpd-foreground"	13 minutes ago	Up 13 minutes	0.0.0.0:18080->80/tcp, :::18080->80/tcp	

```
ubuntu $
```

```
ubuntu $ docker run -d -p 28080:8080 pengbai/docker
-supermario
Unable to find image 'pengbai/docker-supermario:latest' locally
latest: Pulling from pengbai/docker-supermario
092586df9206: Pull complete
ef599477fae0: Pull complete
4530c6472b5d: Pull complete
d34d61487075: Pull complete
272f46008219: Pull complete
5cb62ceb26af: Pull complete
663538d2581c: Pull complete
d226fa7f84b7: Pull complete
9efb6b4ae4f4: Pull complete
77d53852f858: Pull complete
32af9fc740ac: Pull complete
36d34cc6e970: Pull complete
f20dfa6a3a87: Pull complete
Digest: sha256:8d909e68fbc38de1dc6bef982e0ca8c364f3c8021525c9487cec4374a53f15f5
Status: Downloaded newer image for pengbai/docker-supermario:latest
f4bfde0e87bc7a2b0d66da76f7e38f65dc826f296c1ad5a9ca5f0610b7e4bfda
ubuntu $ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
f4bfde0e87bc	pengbai/docker-supermario	"catalina.sh run"	8 seconds ago	Up 7 seconds	0.0.0.0:28080->8080/tcp, :::28080->8080/tcp	
7cce233a2ff8	httpd	"httpd-foreground"	13 minutes ago	Up 13 minutes	0.0.0.0:18080->80/tcp, :::18080->80/tcp	

```
ubuntu $
```

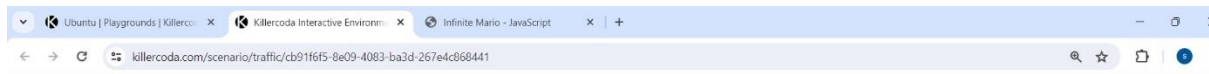
New Terminal Window

New Editor Window

Traffic / Ports

Font Size: 1 2 3 4 5 6 7 8

Forward traffic to the port 28080



Traffic Port Accessor

Access HTTP services which run in your environment

The services need to run on all interfaces (like 0.0.0.0) and not just localhost

Host 1

Common Ports

80 8080

Custom Ports

28080

Access

Check the result

