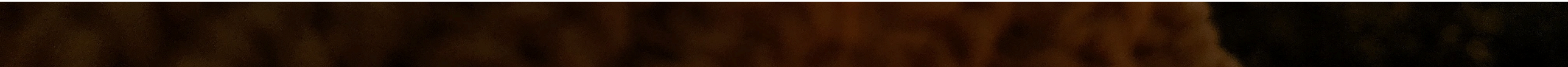
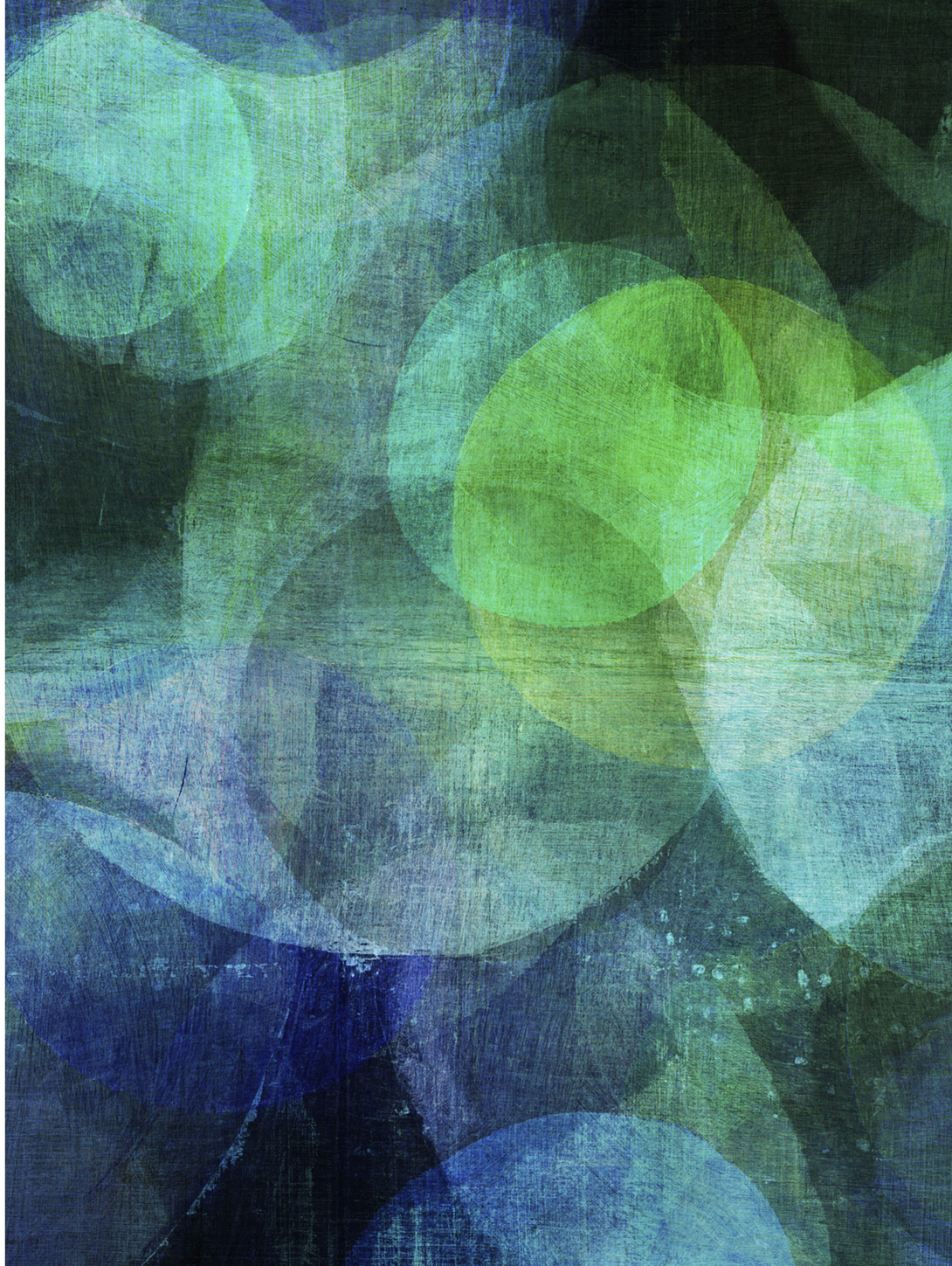
INTRODUCTION TO DOCKER

INTRODUCTION TO DOCKER

Docker Network and Volumes

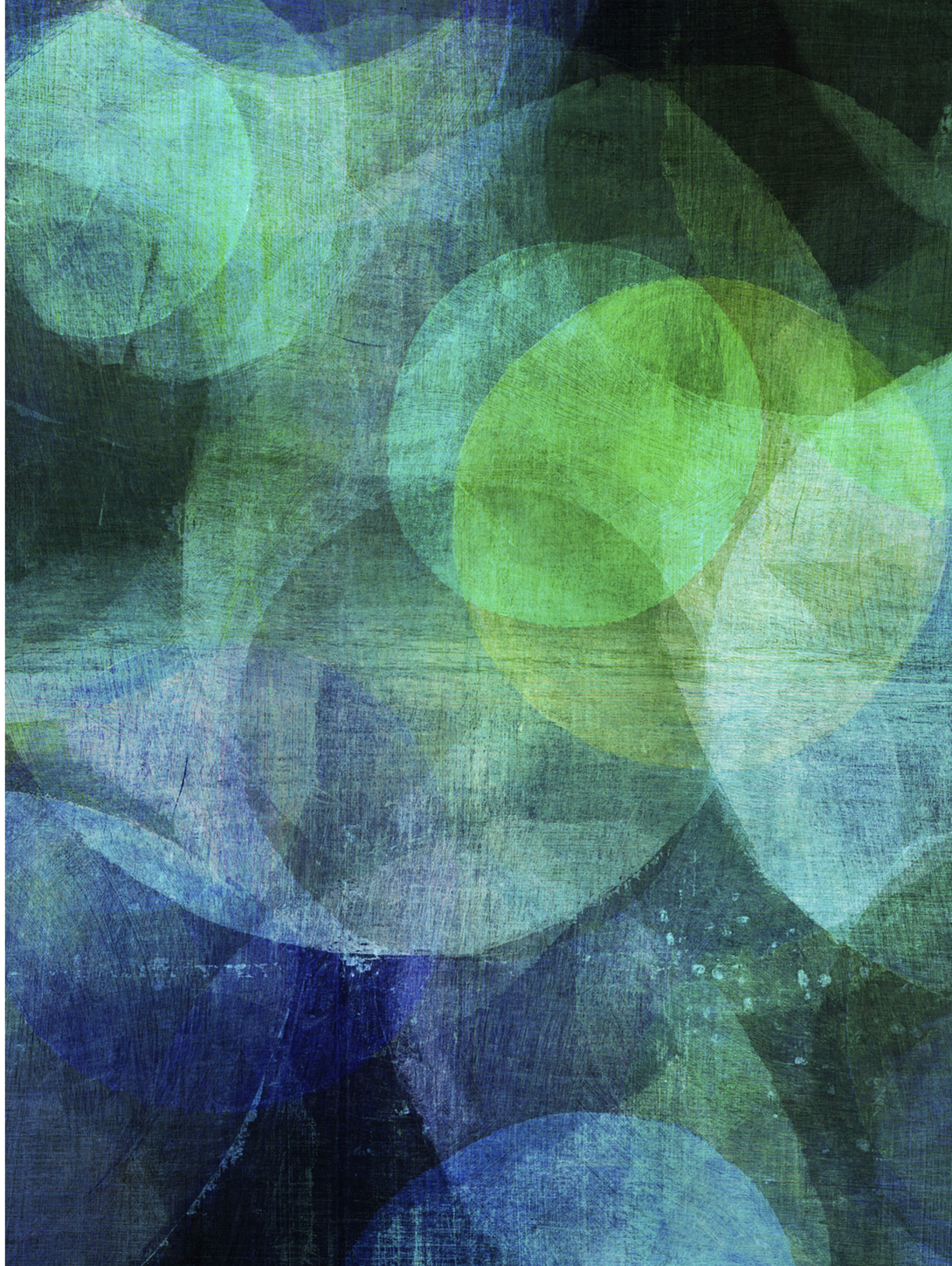


AGENDA



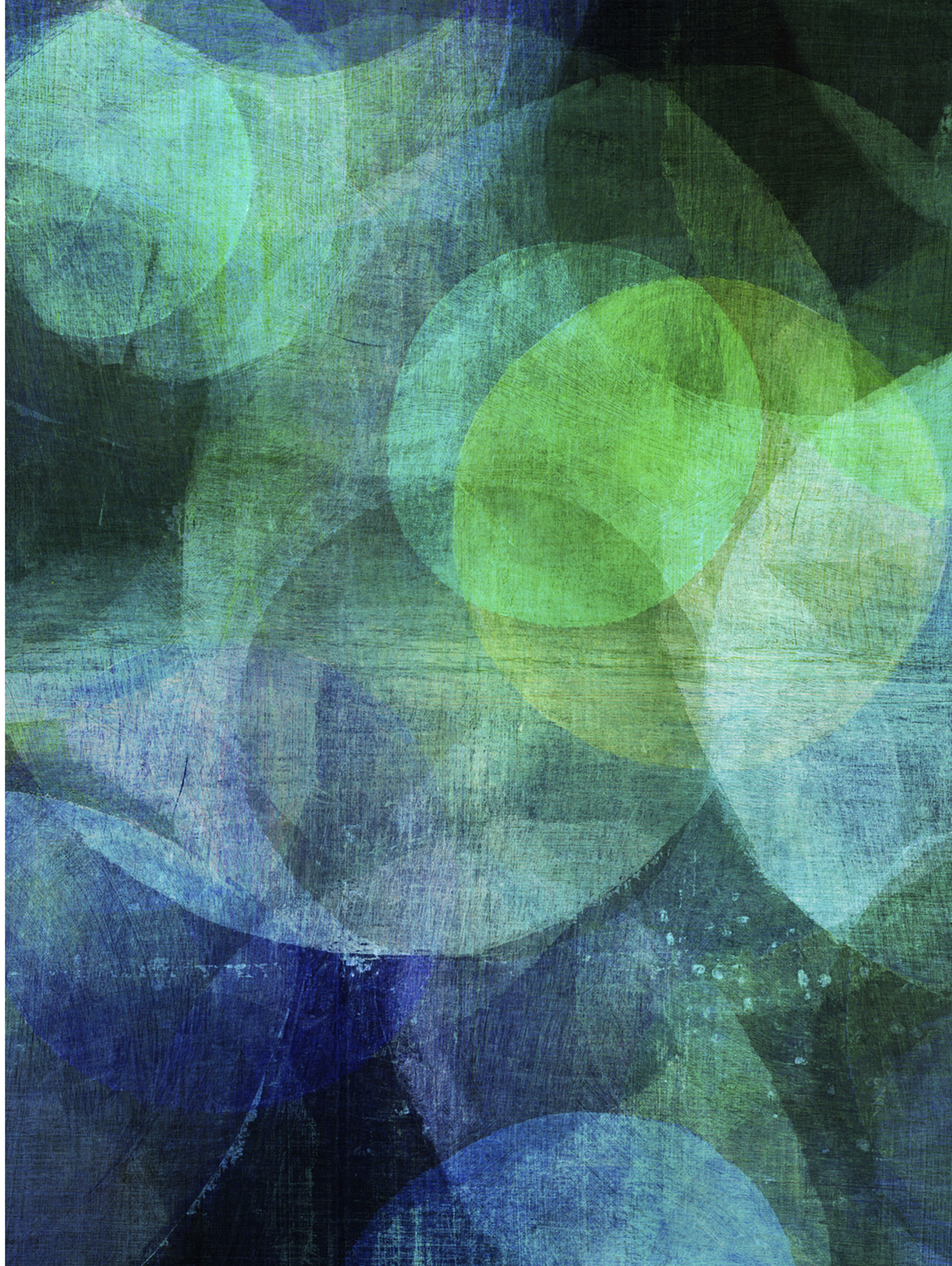
AGENDA

- Basic of Container Networking



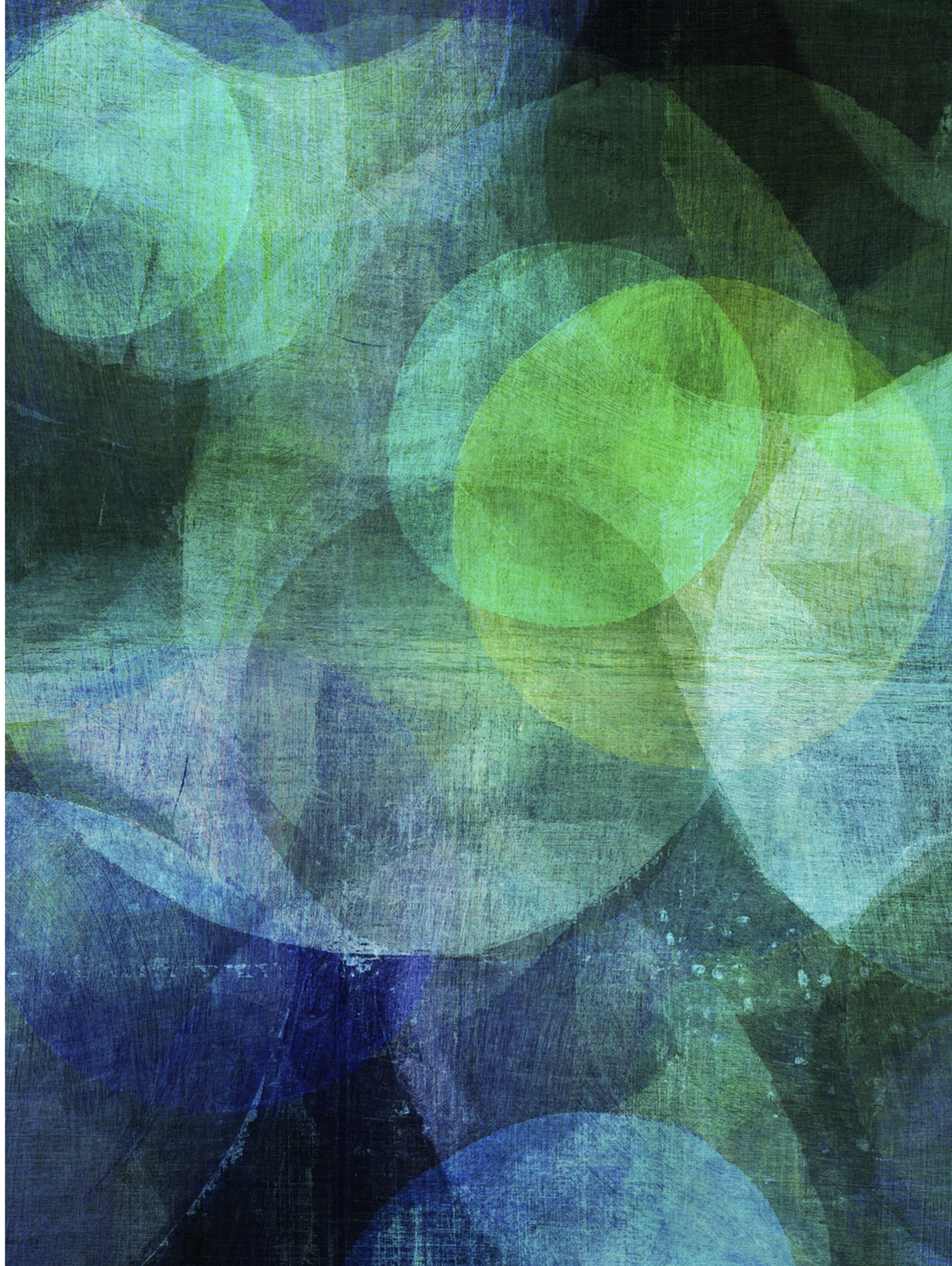
AGENDA

- Basic of Container Networking
- Docker Network



AGENDA

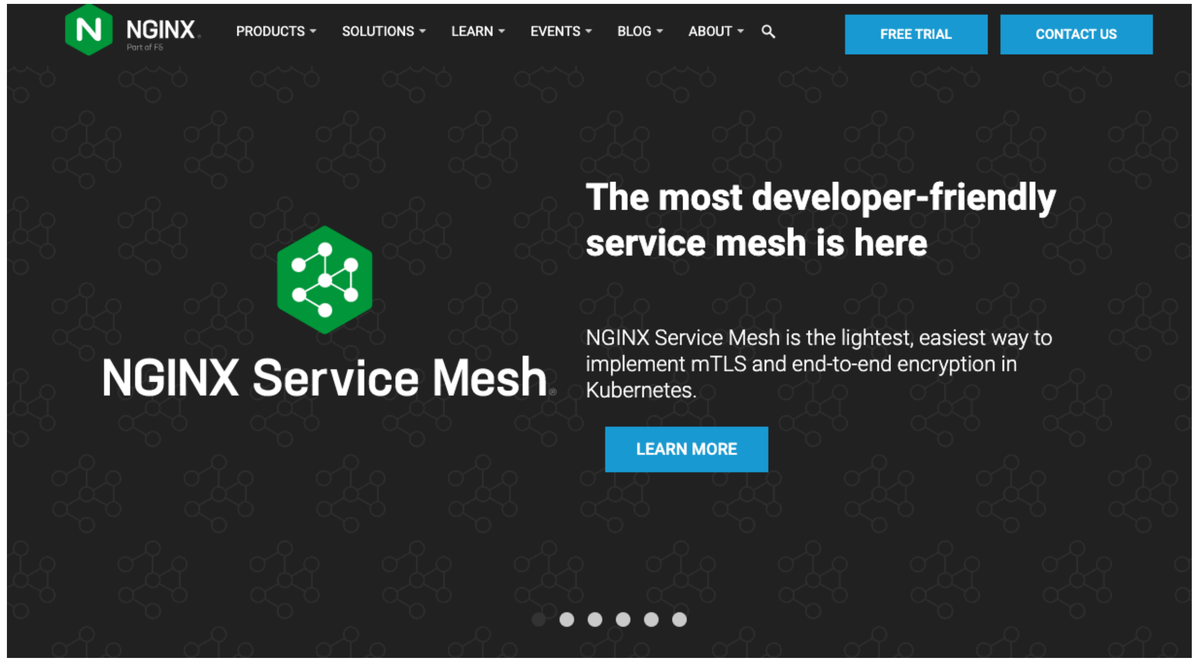
- Basic of Container Networking
- Docker Network
- Docker Storage (Volumes)



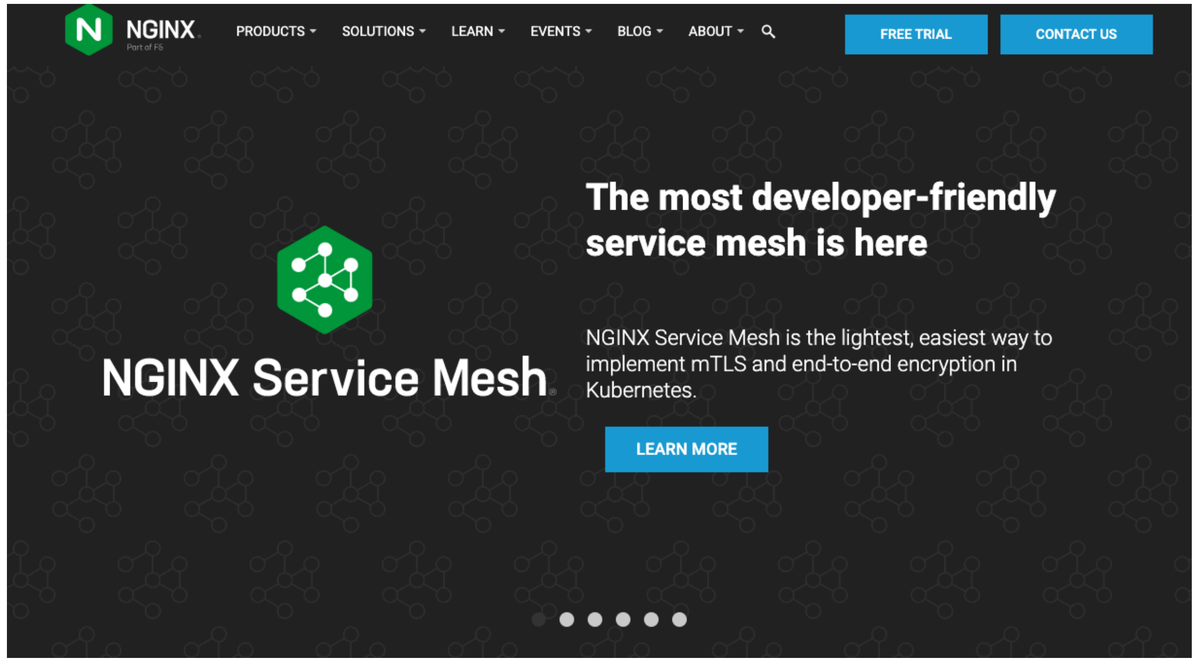
BASIC OF CONTAINER NETWORKING

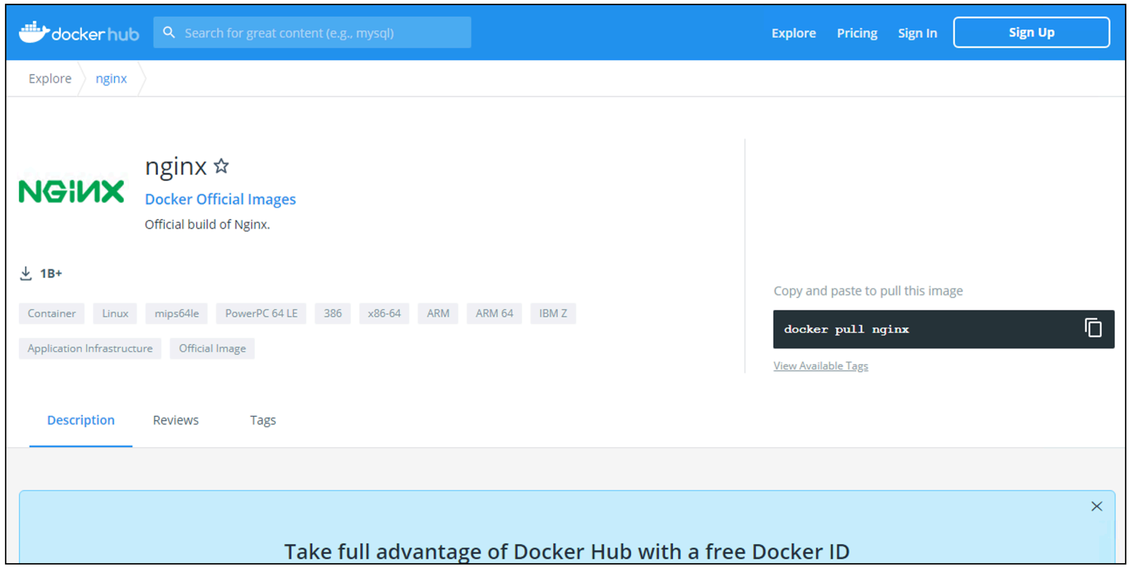
START WEB APPLICATION WITH NGINX

START WEB APPLICATION WITH NGINX

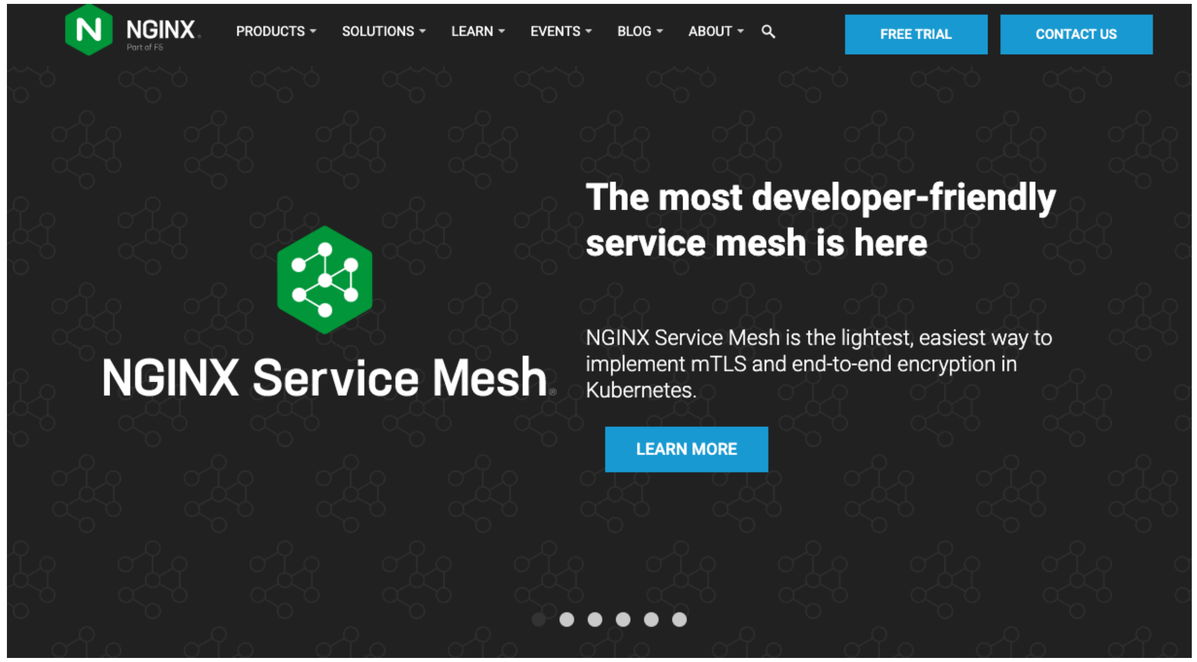


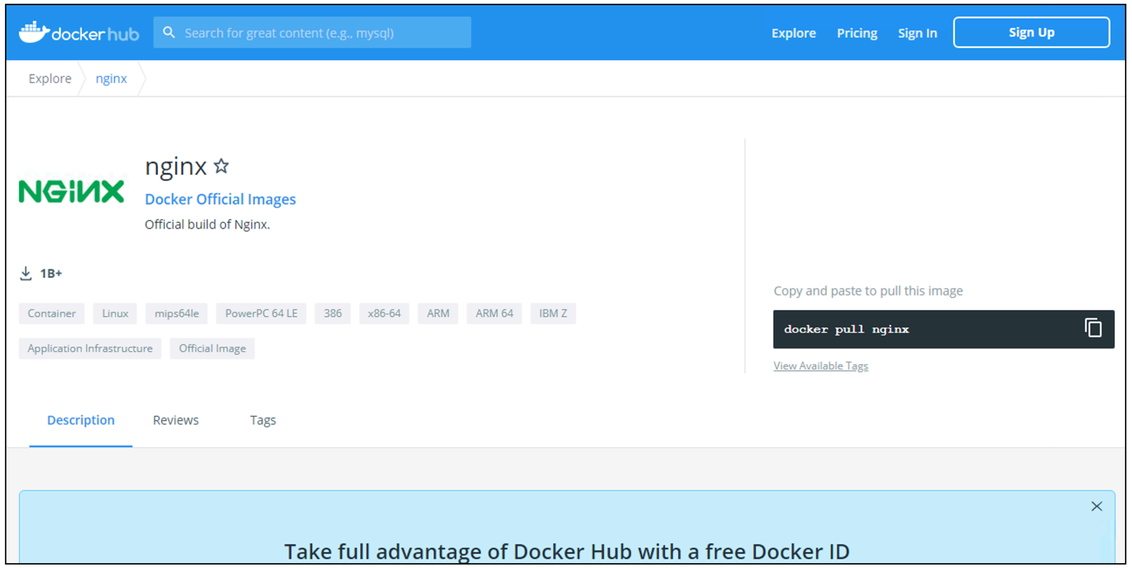
START WEB APPLICATION WITH NGINX





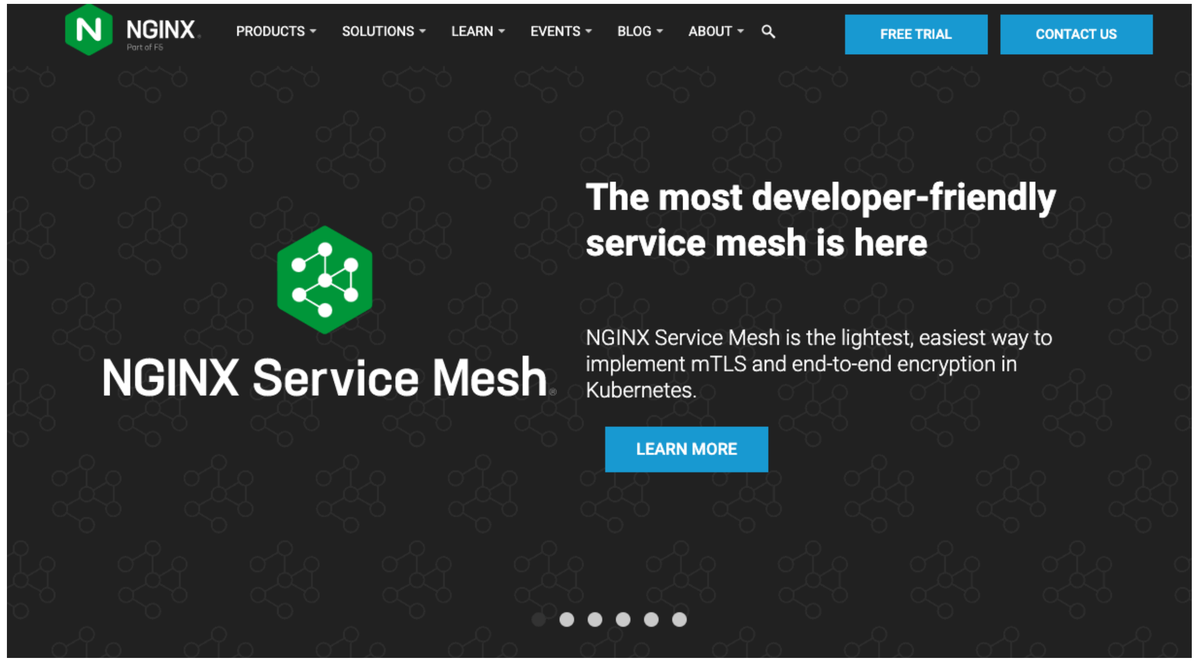
START WEB APPLICATION WITH NGINX

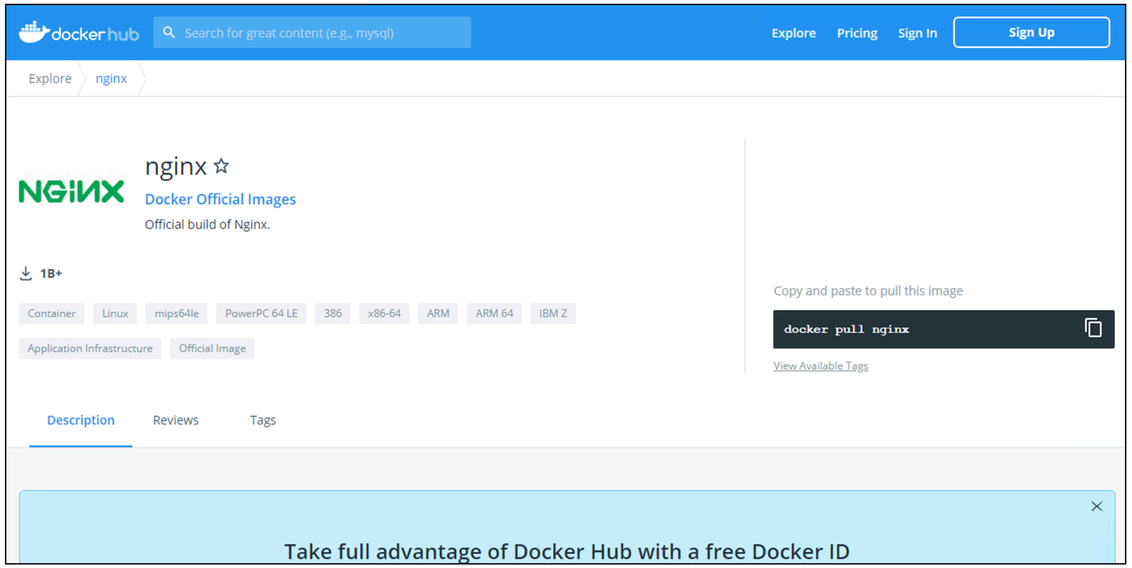





```
$ docker container run --name hello-nginx -d nginx
```

START WEB APPLICATION WITH NGINX

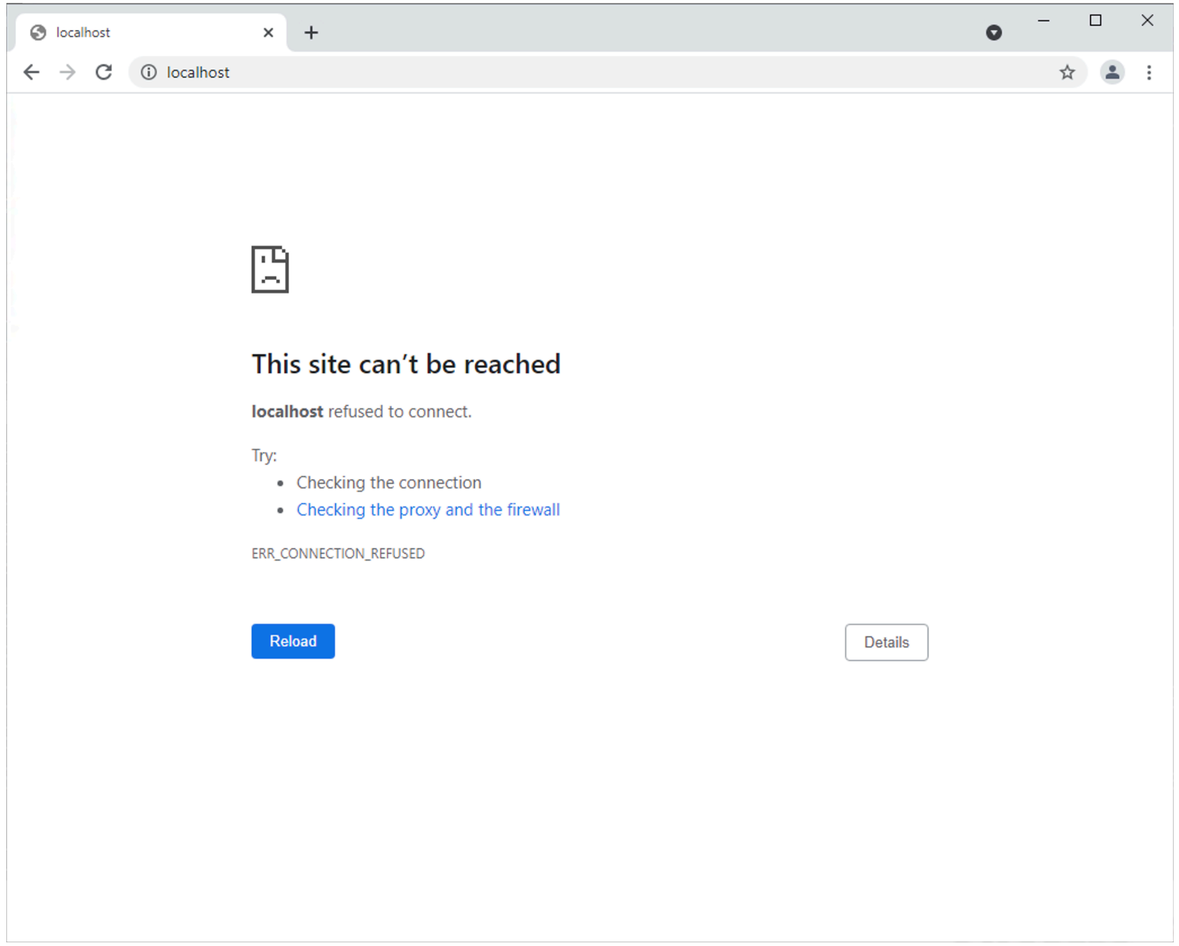




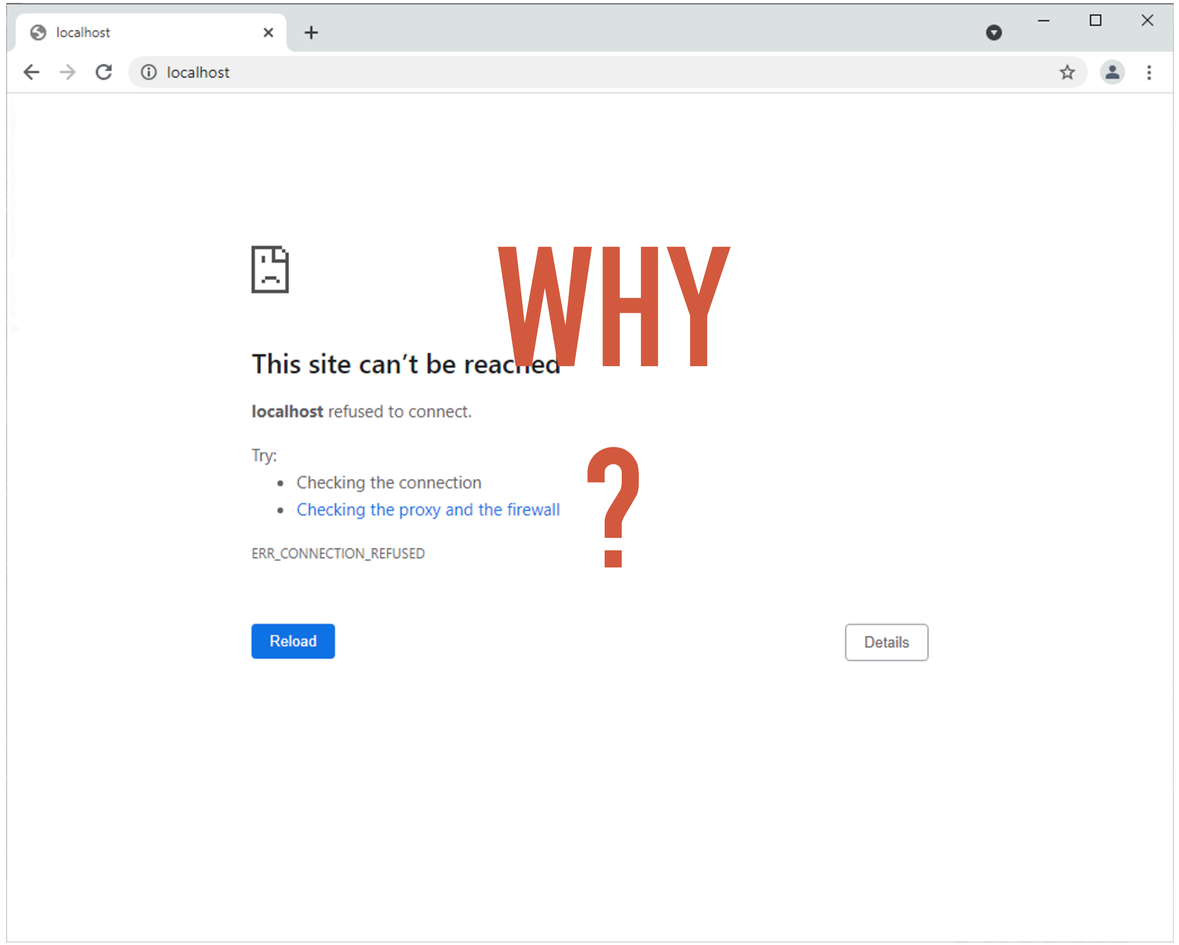

```
$ docker container run --name hello-nginx -d nginx
```

```
74dbf8abb5a5b8533c11c86e1949cfe57b5104b2fa721237abfea696c6cc6289
```

CAN'T CONNECT TO NGINX WITH PORT 80



CAN'T CONNECT TO NGINX WITH PORT 80



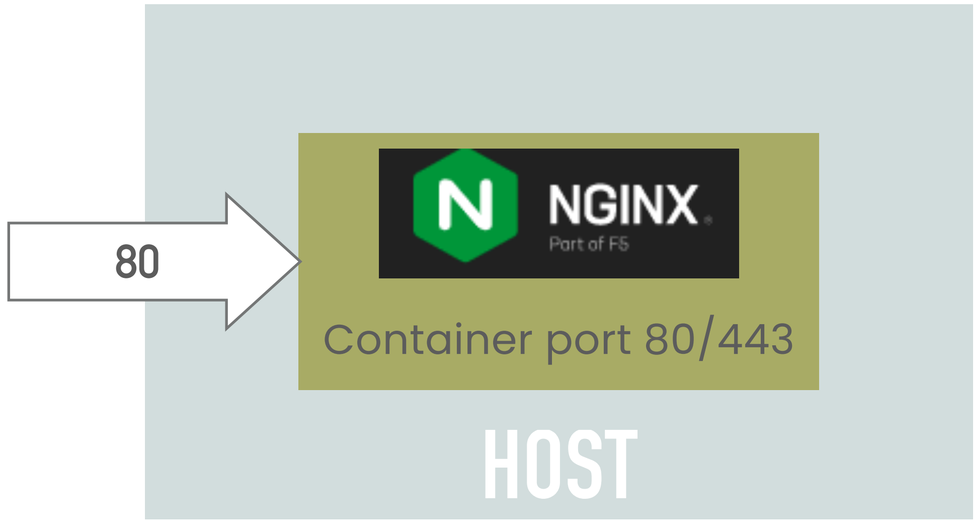
PORT OF HOST AND CONTAINER

PORT OF HOST AND CONTAINER

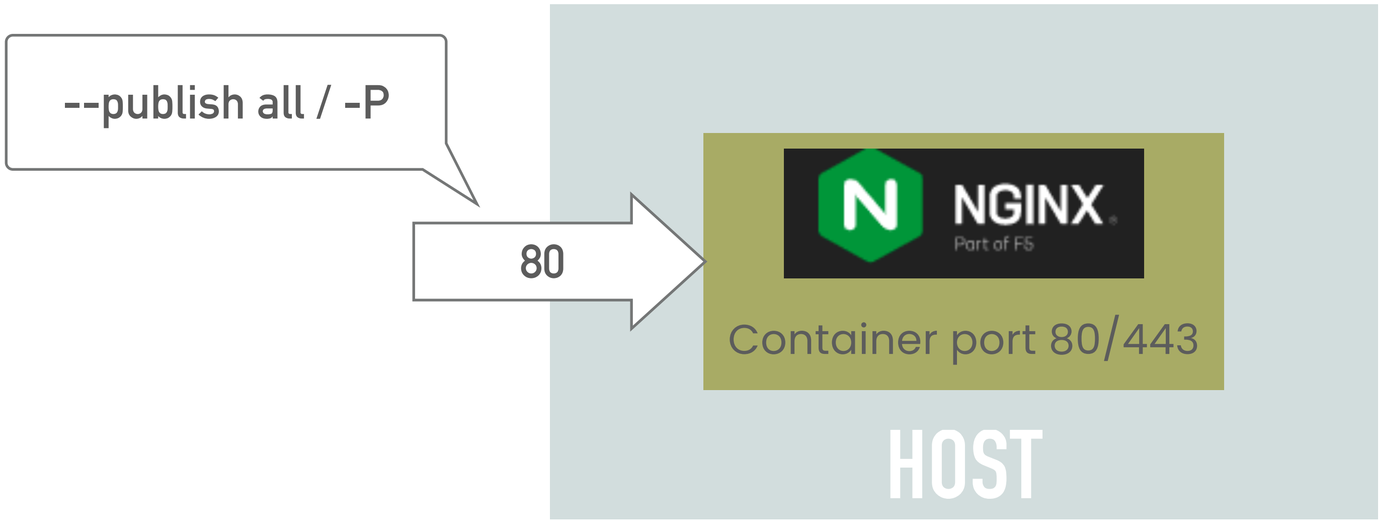
PORT OF HOST AND CONTAINER

PORT OF HOST AND CONTAINER

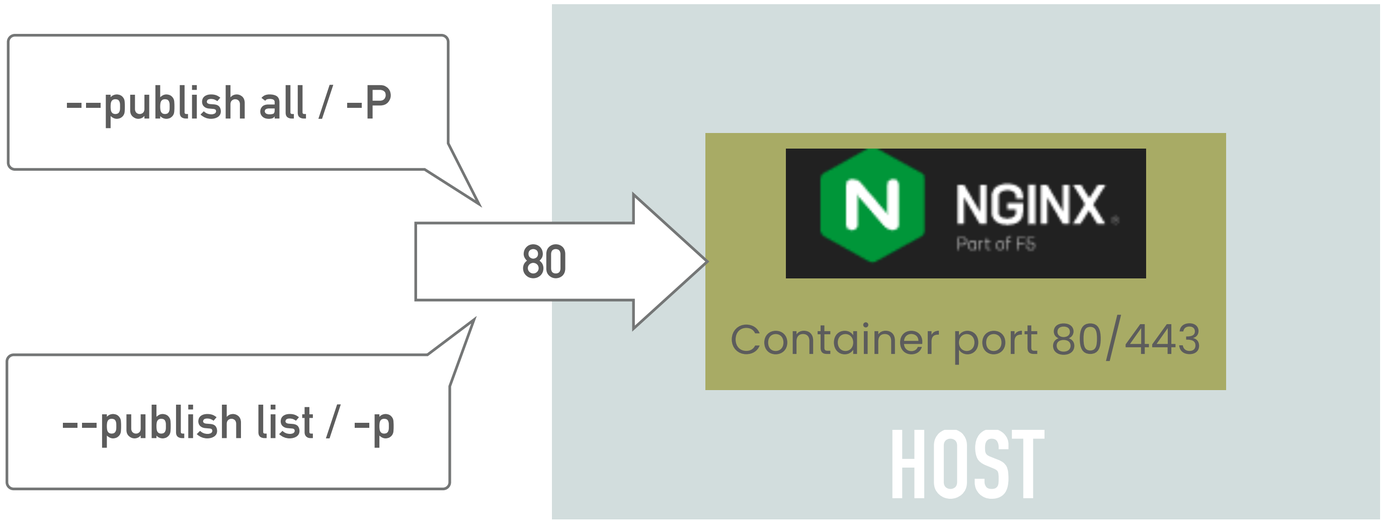
PORT OF HOST AND CONTAINER



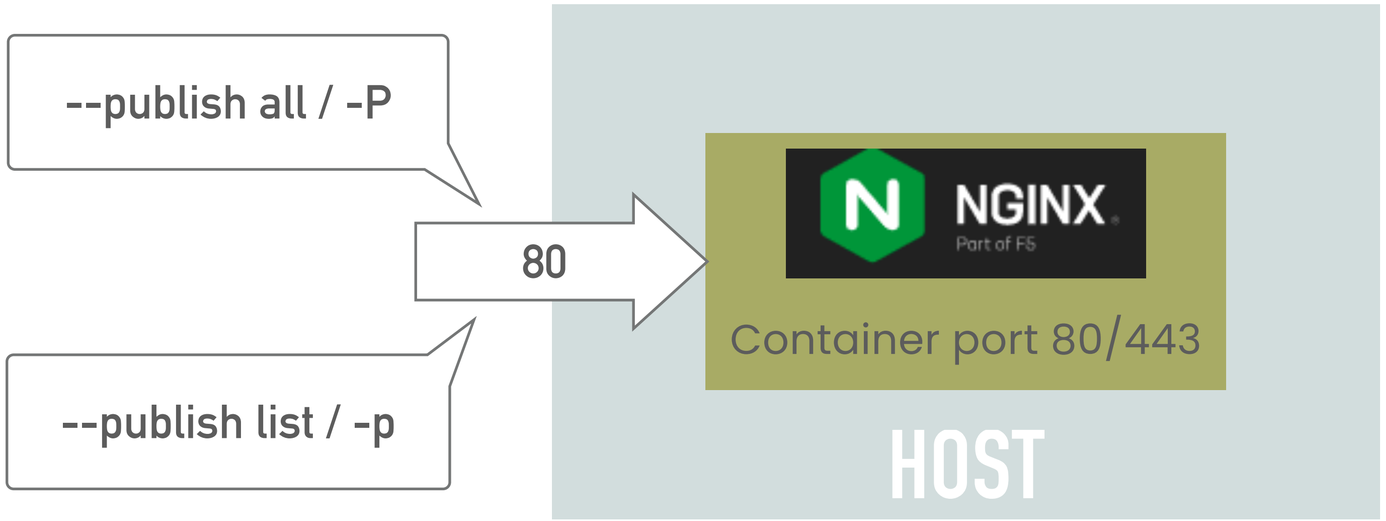
PORT OF HOST AND CONTAINER

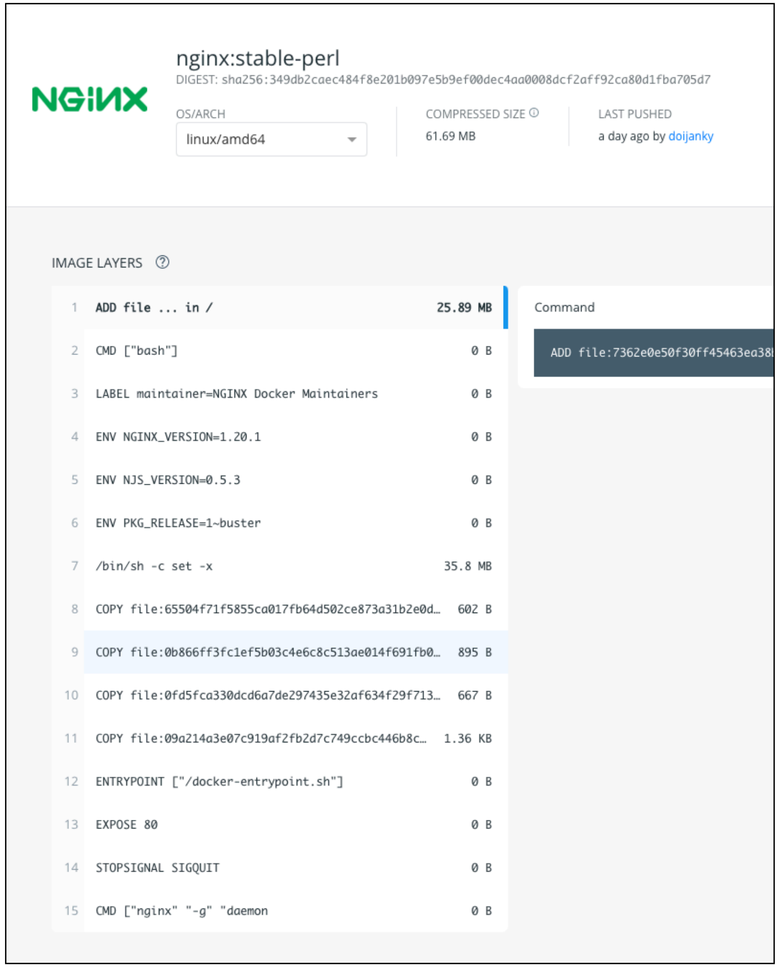


PORT OF HOST AND CONTAINER

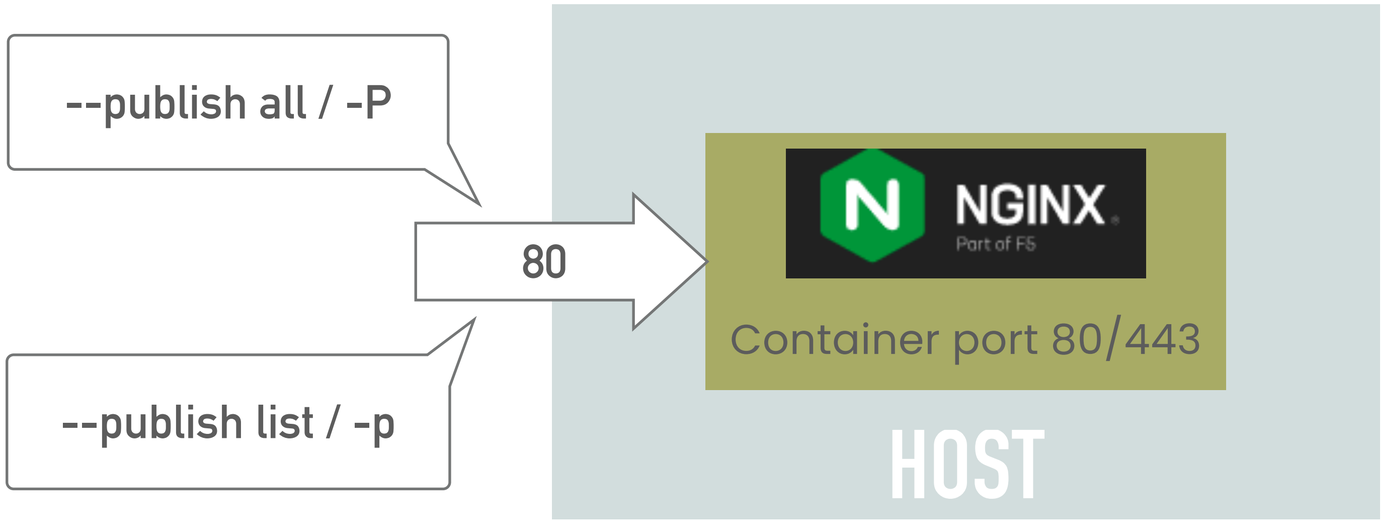


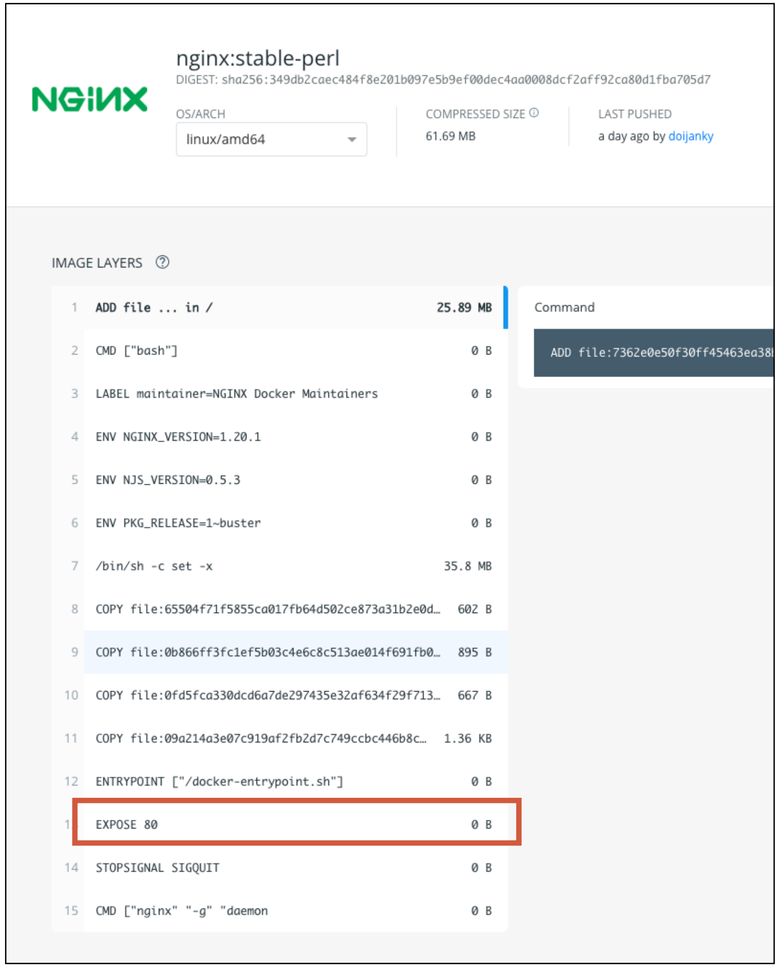
PORT OF HOST AND CONTAINER





PORT OF HOST AND CONTAINER





START WEB APPLICATION WITH NGINX

START WEB APPLICATION WITH NGINX

```
$ docker container run --name hello-nginx -d -p 80:80 nginx
```

START WEB APPLICATION WITH NGINX

```
$ docker container run --name hello-nginx -d -p 80:80 nginx
```

OR

START WEB APPLICATION WITH NGINX

```
$ docker container run --name hello-nginx -d -p 80:80 nginx
```

OR

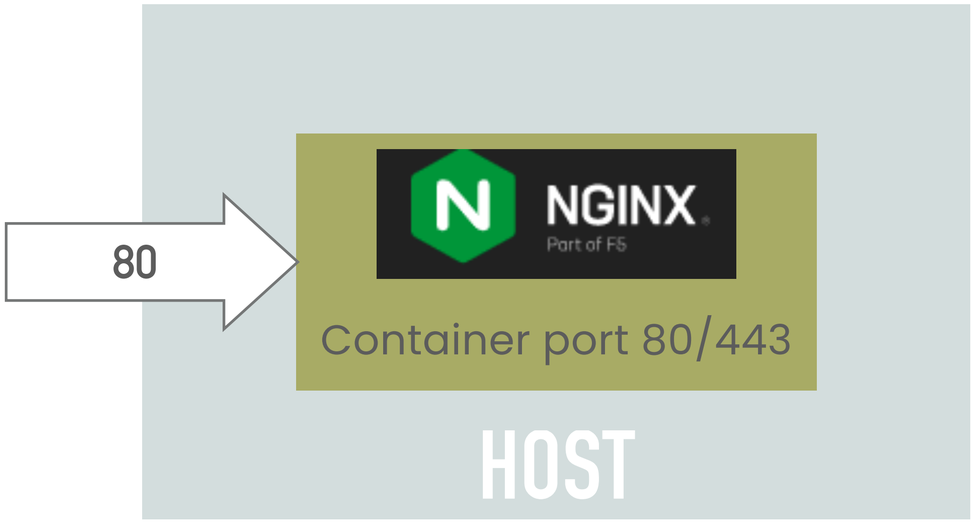
```
$ docker container run --name hello-nginx -d --publish 80:80 nginx
```

START WEB APPLICATION WITH NGINX

```
$ docker container run --name hello-nginx -d -p 80:80 nginx
```

OR

```
$ docker container run --name hello-nginx -d --publish 80:80 nginx
```

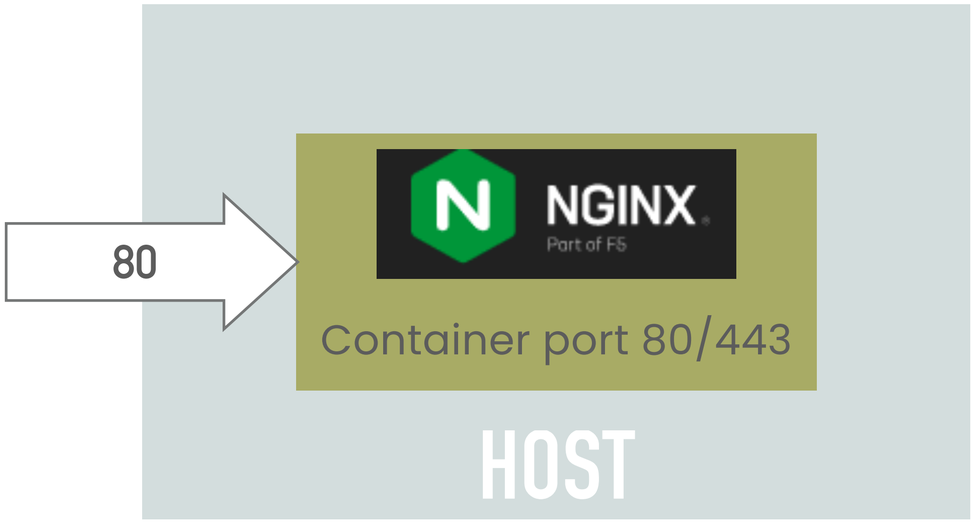


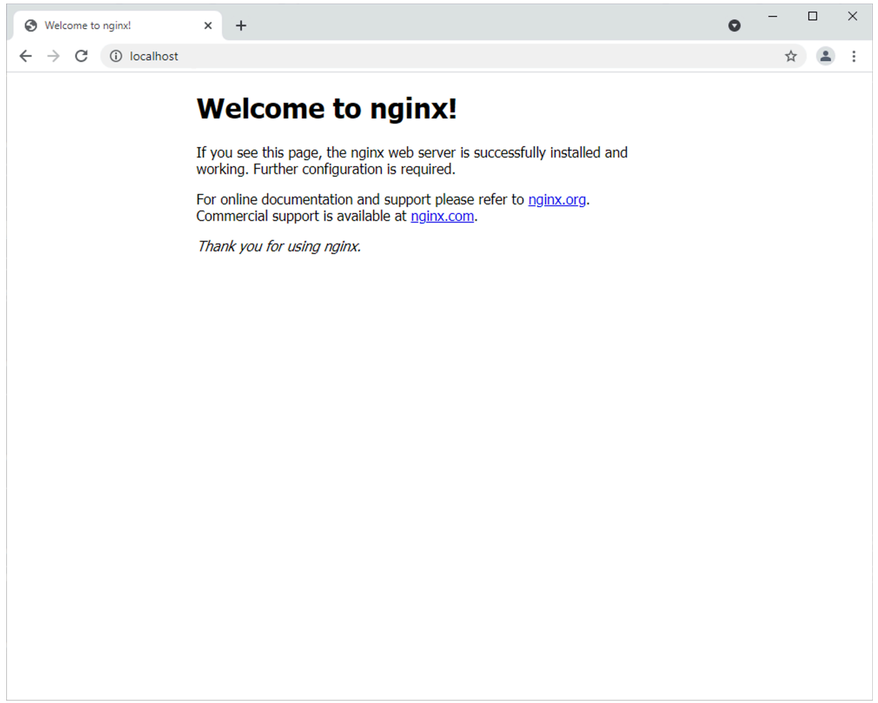
START WEB APPLICATION WITH NGINX

```
$ docker container run --name hello-nginx -d -p 80:80 nginx
```

OR

```
$ docker container run --name hello-nginx -d --publish 80:80 nginx
```



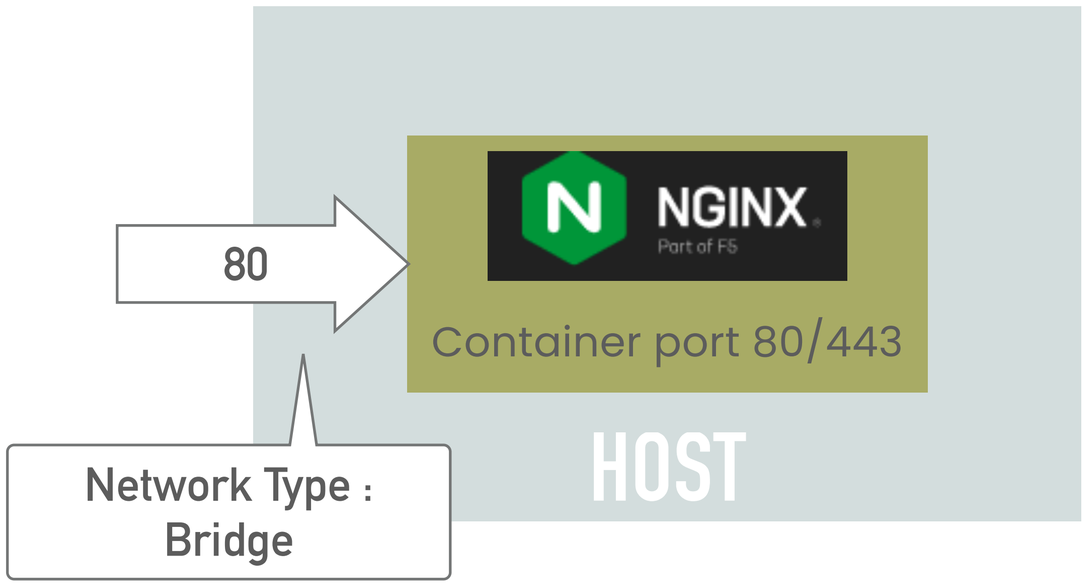


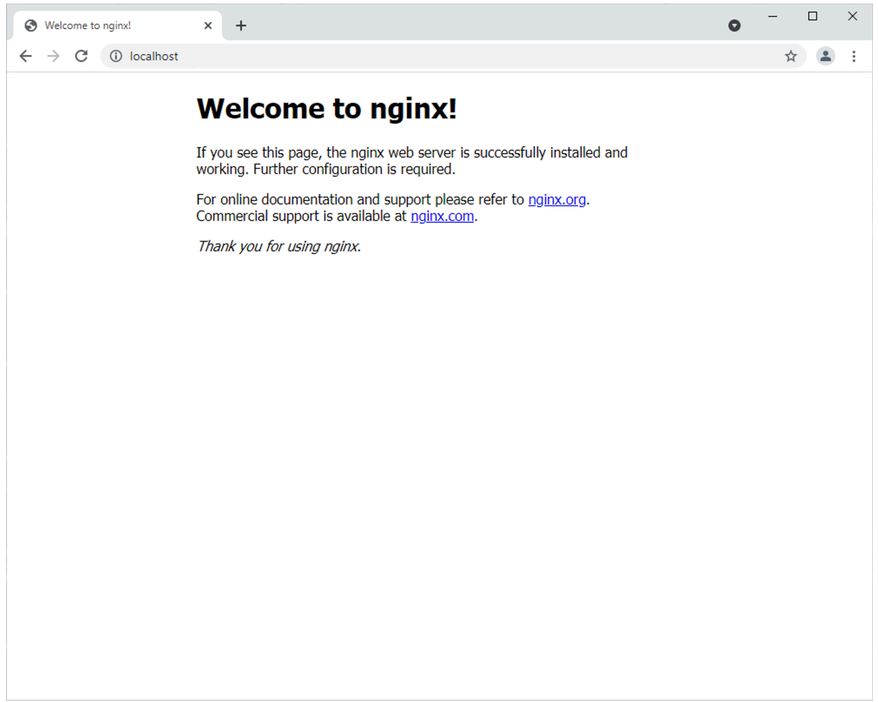
START WEB APPLICATION WITH NGINX

```
$ docker container run --name hello-nginx -d -p 80:80 nginx
```

OR

```
$ docker container run --name hello-nginx -d --publish 80:80 nginx
```



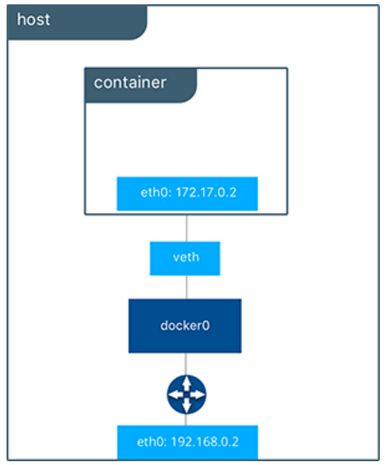


DOCKER NETWORK

DOCKER – NETWORK CLI AND NETWORK DRIVERS

$ docker network ls

```
C:\Users\karan>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
5c3bb2502623   bridge    bridge    local
6877ad312030   host      host      local
d3d51f296ce4   none      null      local
```



`bridge network`

- `bridge`: The default network driver.
- `host`: For standalone containers, remove network isolation between the container and the Docker host, and use the host's networking directly.
- `overlay`: Overlay networks connect multiple Docker daemons together and enable swarm services to communicate with each other.
- `macvlan`: Macvlan networks allow you to assign a MAC address to a container, making it appear as a physical device on your network.
- `none`: For this container, disable all networking.

DOCKER - BRIDGE NETWORK

```
$ docker network inspect bridge
```

DOCKER - BRIDGE NETWORK

$ docker network inspect bridge

```
C:\Users\karan>docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "5c3bb250262393a739a4c9f81b9291859738b94585eed3a0245ebc4d2848a010",
        "Created": "2021-05-26T01:56:57.6721089Z",
        "Scope": "local",
        "Driver": "bridge",
        "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": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "bd087effleaf79f353c1448924bed9985438382ff822638dc125b8c7b0a3ea83": {
                "Name": "hello-nginx",
                "EndpointID": "306a62f3249ab8cfad056da39ccc81d93f328bf8151640b539e803a4ad32db6c",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
```

DOCKER - BRIDGE NETWORK

$ docker network inspect bridge

```
C:\Users\karan>docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "5c3bb250262393a739a4c9f81b9291859738b94585eed3a0245ebc4d2848a010",
        "Created": "2021-05-26T01:56:57.6721089Z",
        "Scope": "local",
        "Driver": "bridge",
        "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": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "bd087effleaf79f353c1448924bed9985438382ff822638dc125b8c7b0a3ea83": {
                "Name": "hello-nginx",
                "EndpointID": "306a62f3249ab8cfad056da39ccc81d93f328bf8151640b539e803a4ad32db6c",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
```

DOCKER INTERNAL NETWORK

```
"Config": [
    {
        "Subnet": "172.17.0.0/16",
        "Gateway": "172.17.0.1"
    }
]
```

DOCKER - BRIDGE NETWORK

$ docker network inspect bridge

```
C:\Users\karan>docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "5c3bb250262393a739a4c9f81b9291859738b94585eed3a0245ebc4d2848a010",
        "Created": "2021-05-26T01:56:57.6721089Z",
        "Scope": "local",
        "Driver": "bridge",
        "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": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "bd087effleaf79f353c1448924bed99854
                "Name": "hello-nginx",
                "EndpointID": "306a62f3249ab8cfad05
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
```

DOCKER INTERNAL NETWORK

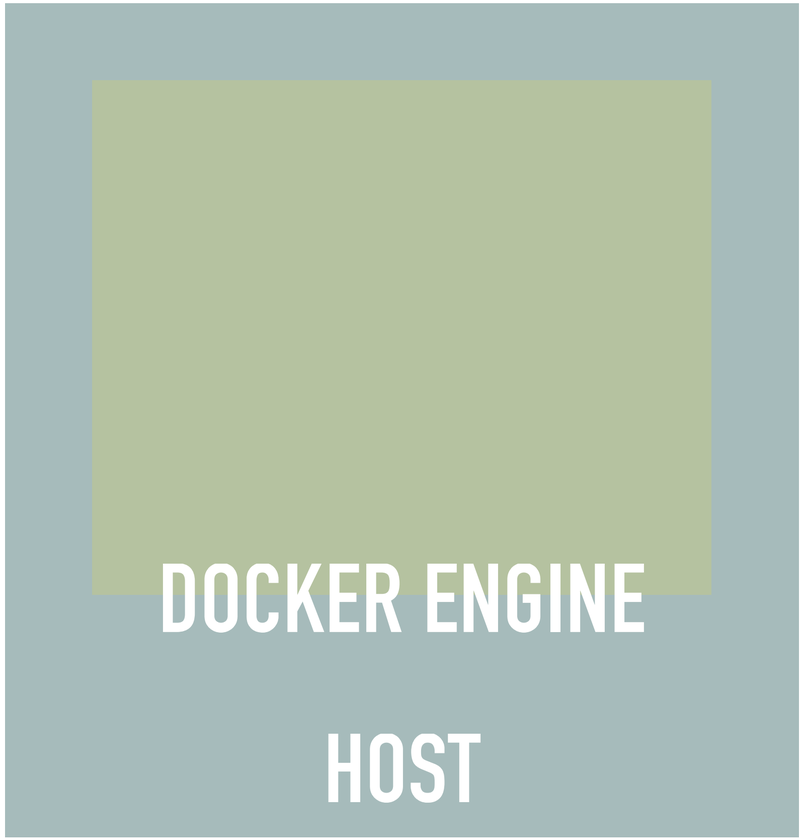
```
"Config": [
    {
        "Subnet": "172.17.0.0/16",
        "Gateway": "172.17.0.1"
    }
]
```

IP-Address of hello-nginx

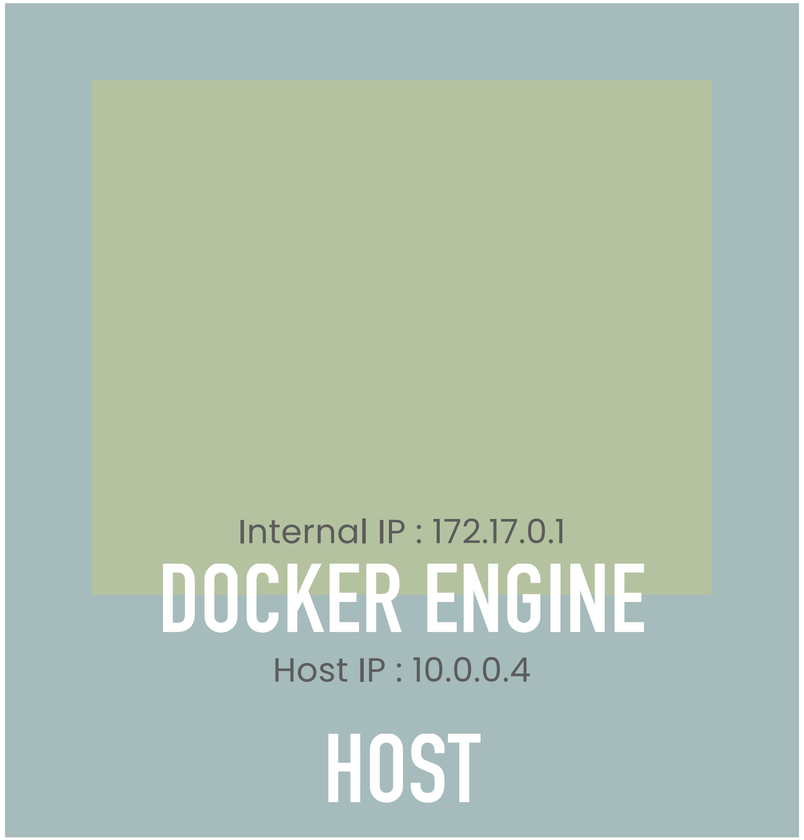
```
"Name": "hello-nginx",
"EndpointID": "306a62f3249ab8cfad056d

"MacAddress": "02:42:ac:11:00:02",
"IPv4Address": "172.17.0.2/16",
"IPv6Address": ""
```

DOCKER – BRIDGE NETWORK

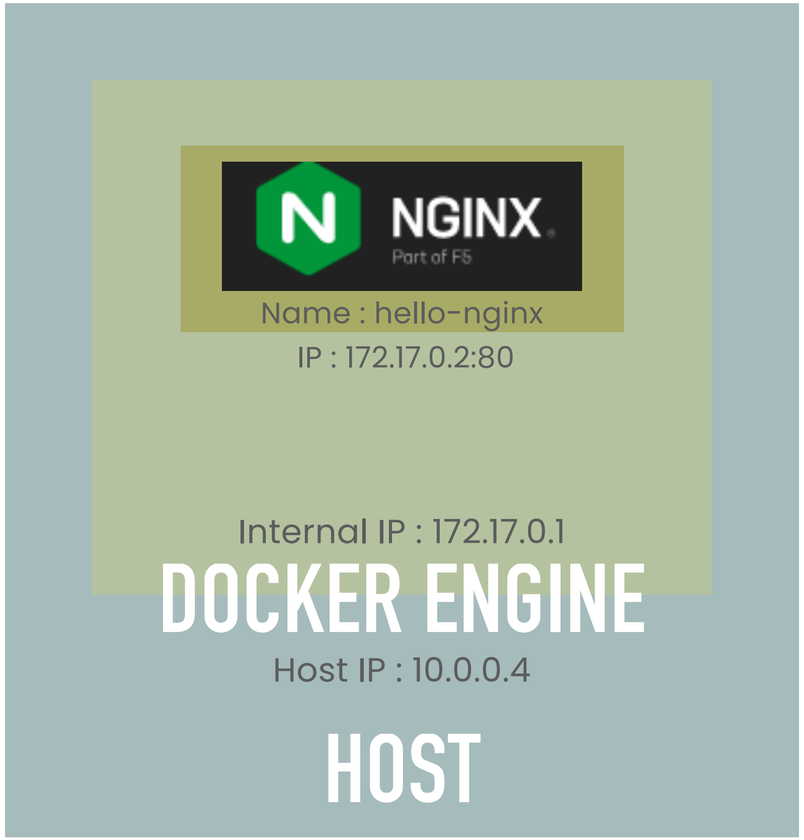
DOCKER - BRIDGE NETWORK



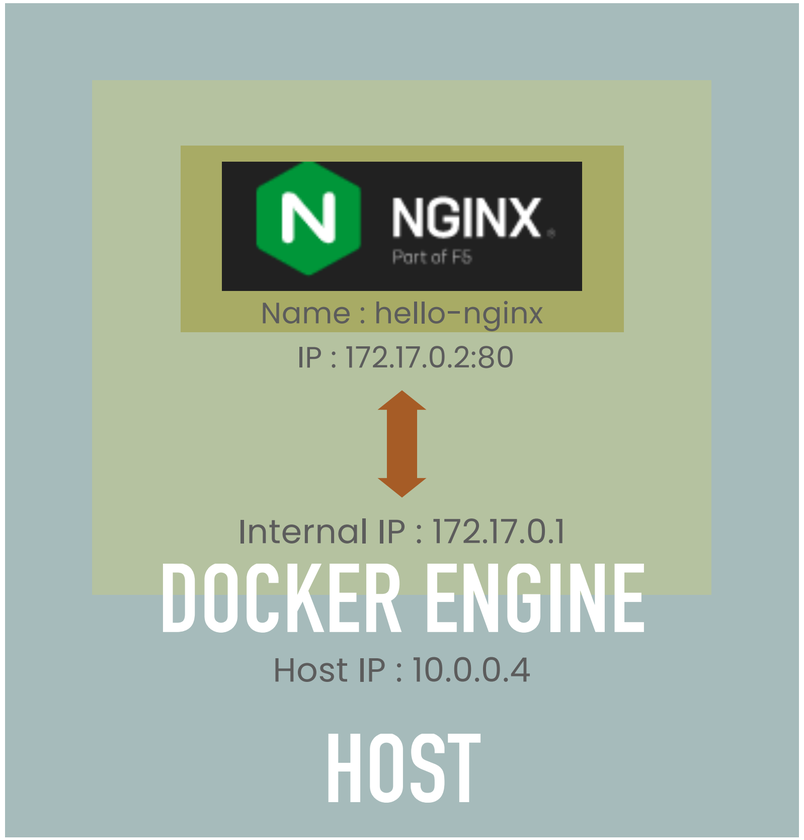
DOCKER - BRIDGE NETWORK



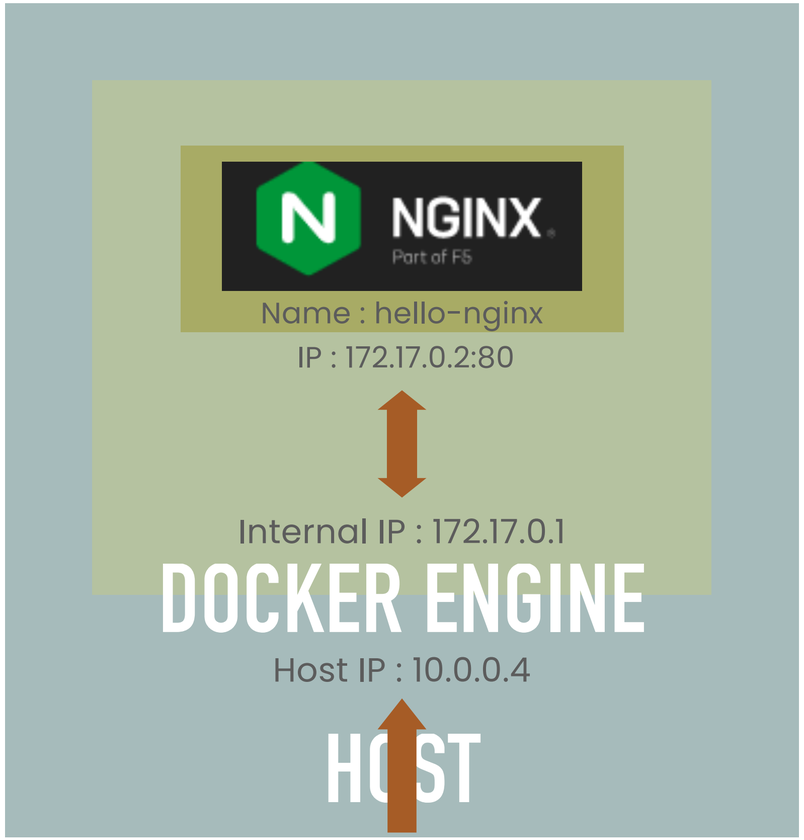
DOCKER - BRIDGE NETWORK



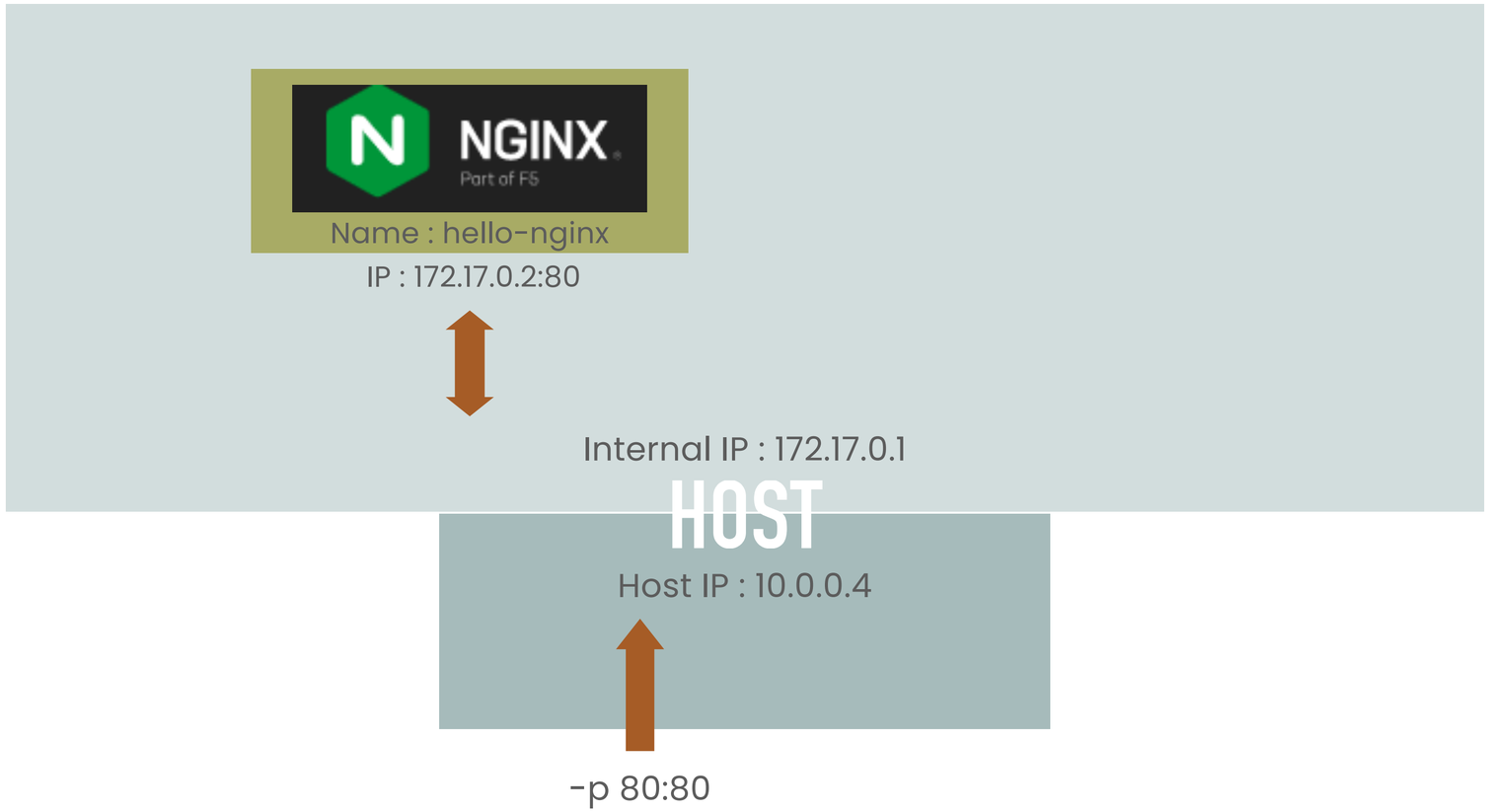
DOCKER - BRIDGE NETWORK



DOCKER - BRIDGE NETWORK

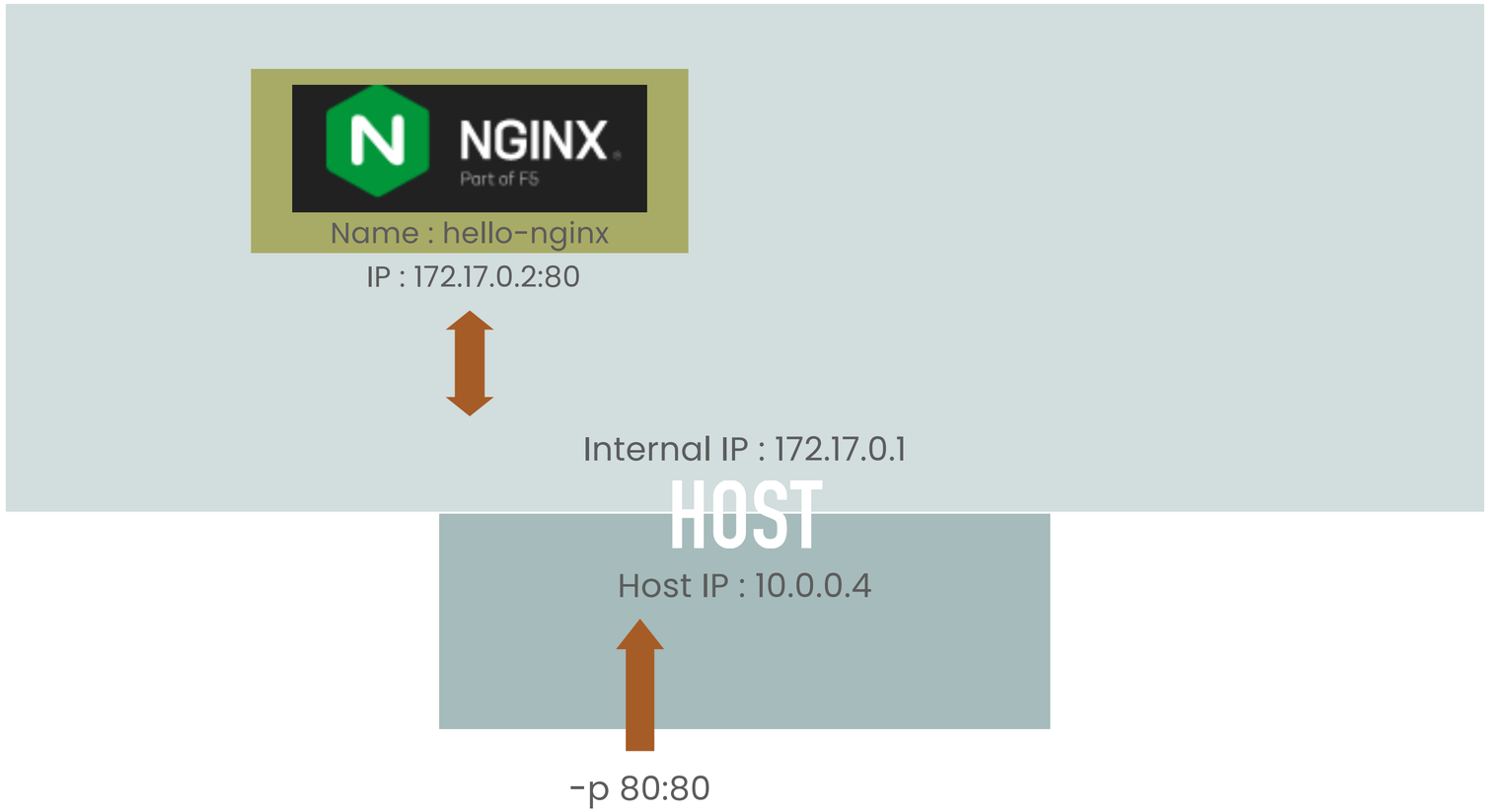


DOCKER - CREATE NEW NGINX CONTAINER



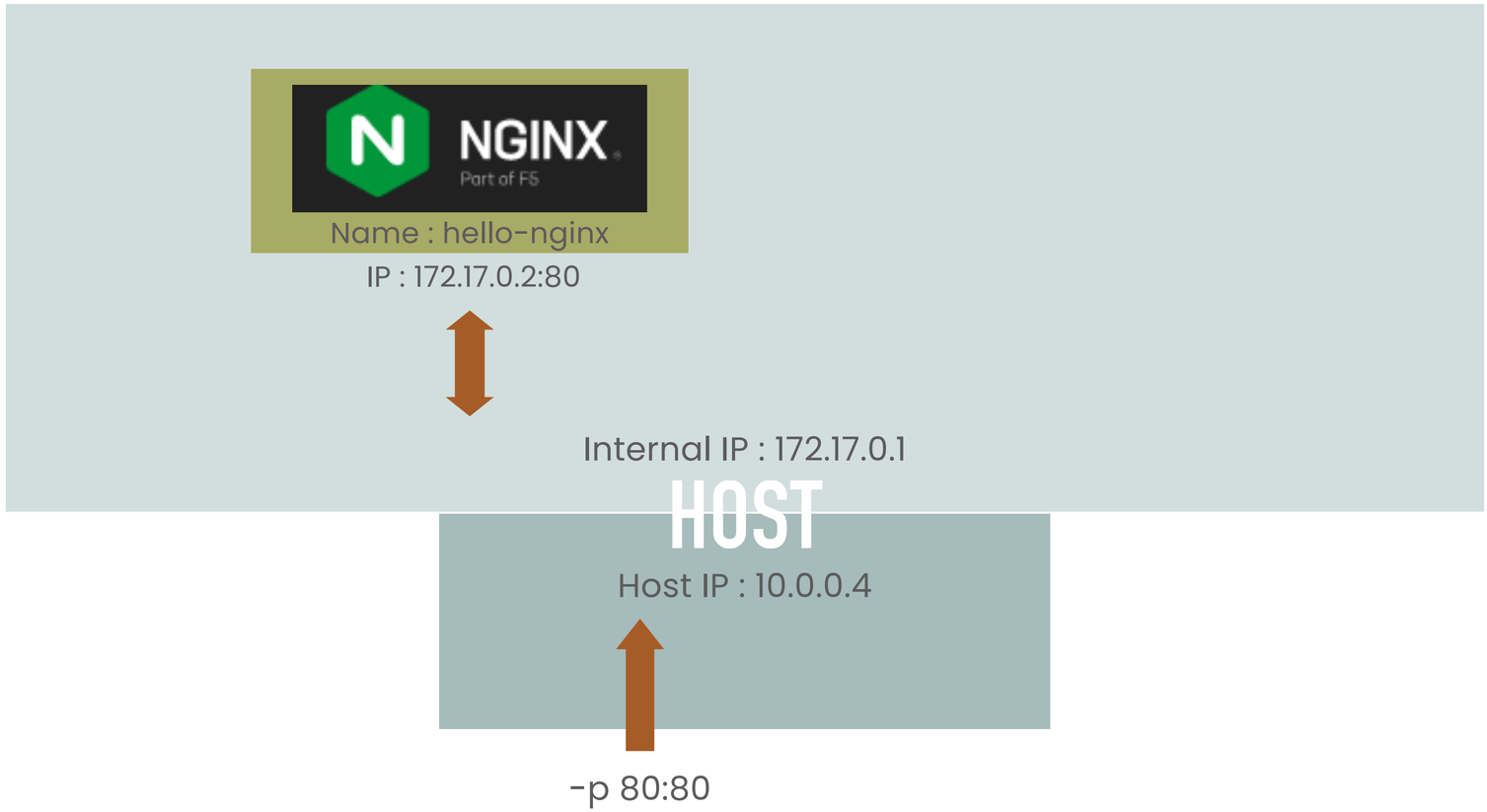
DOCKER - CREATE NEW NGINX CONTAINER

```
$ docker container run --name hello-nginx-2 -d -p 81:80 nginx
```



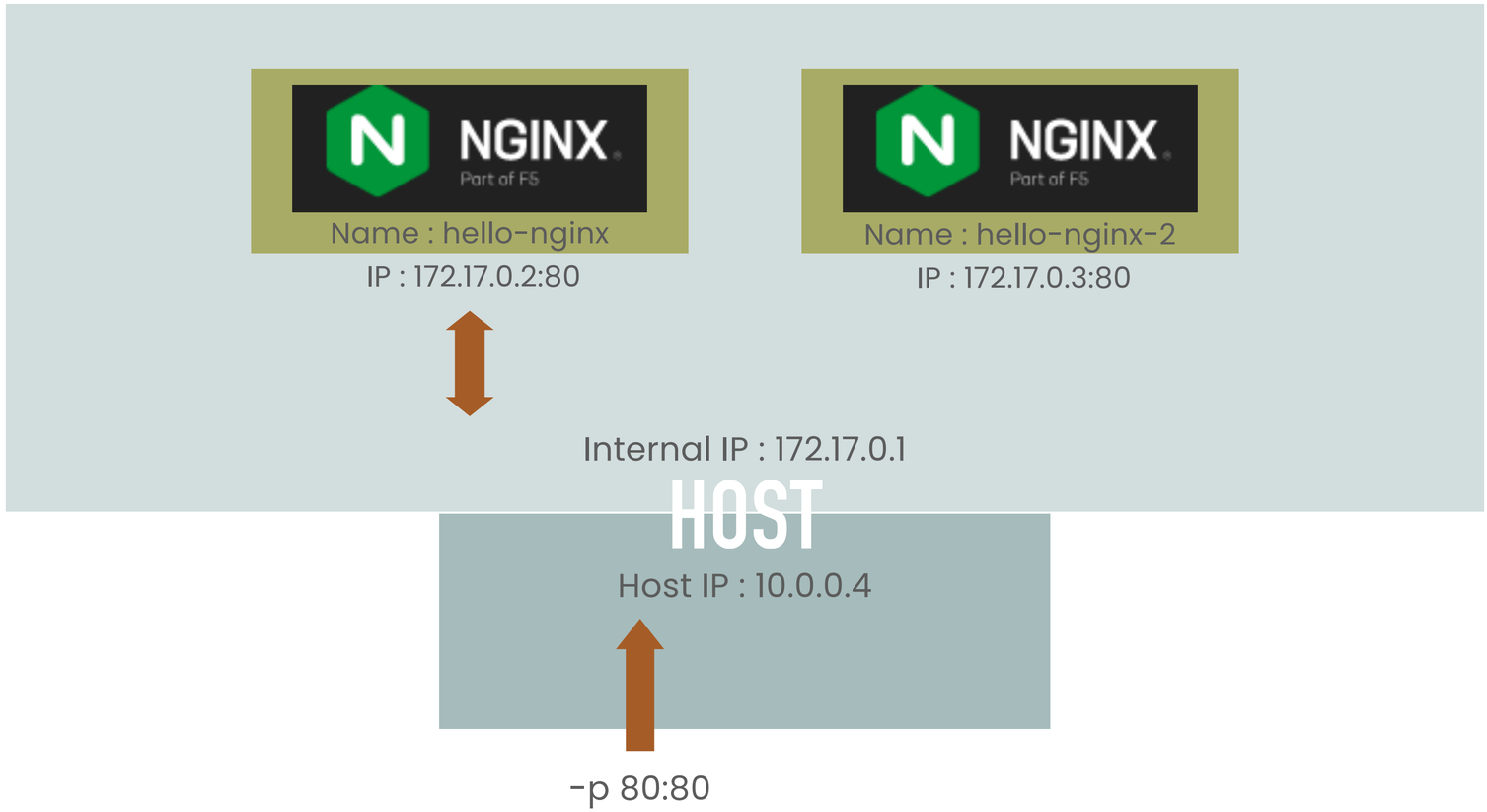
DOCKER - CREATE NEW NGINX CONTAINER

```
$ docker container run --name hello-nginx-2 -d -p 81:80 nginx
```



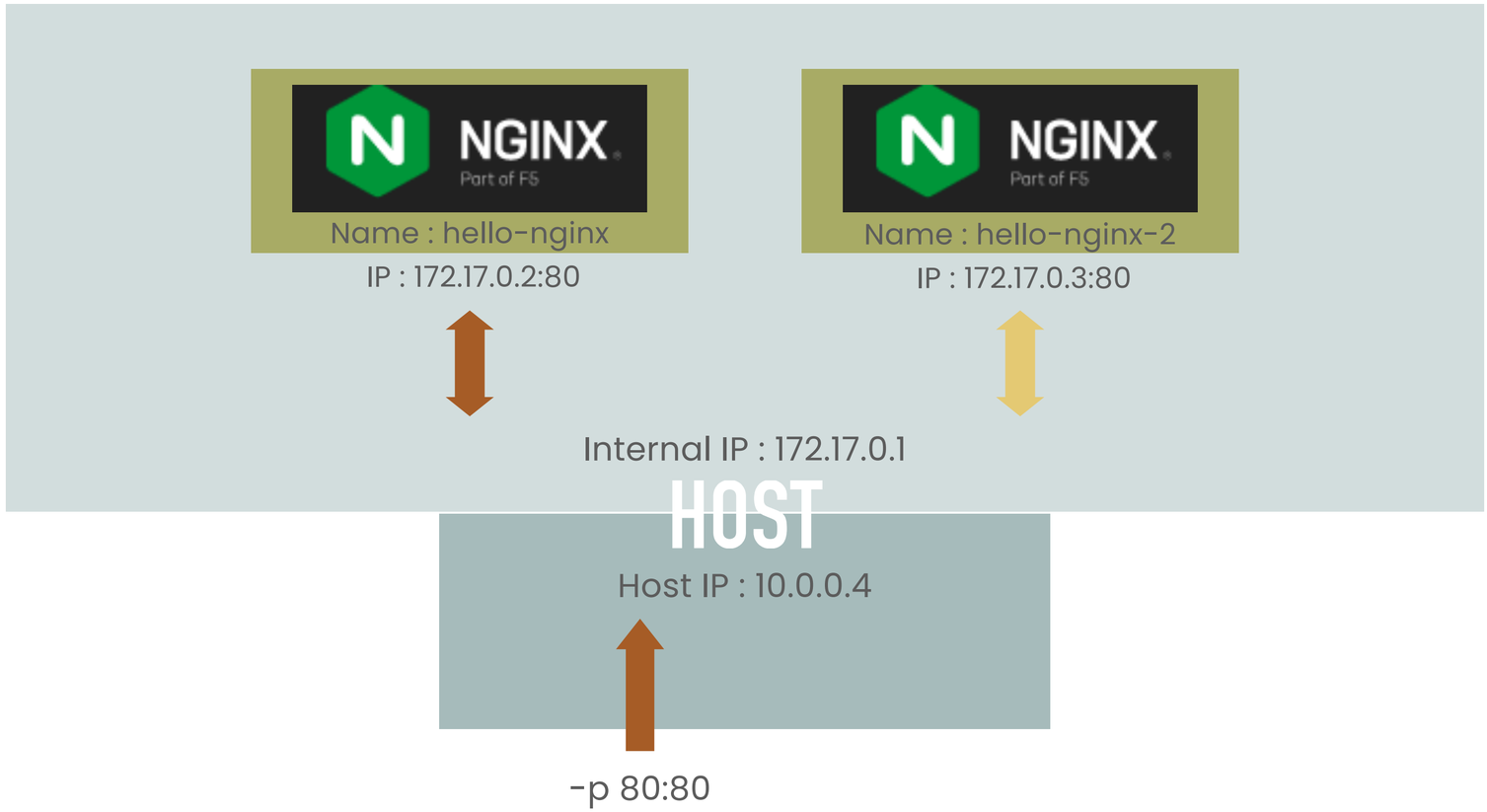
DOCKER - CREATE NEW NGINX CONTAINER

```
$ docker container run --name hello-nginx-2 -d -p 81:80 nginx
```



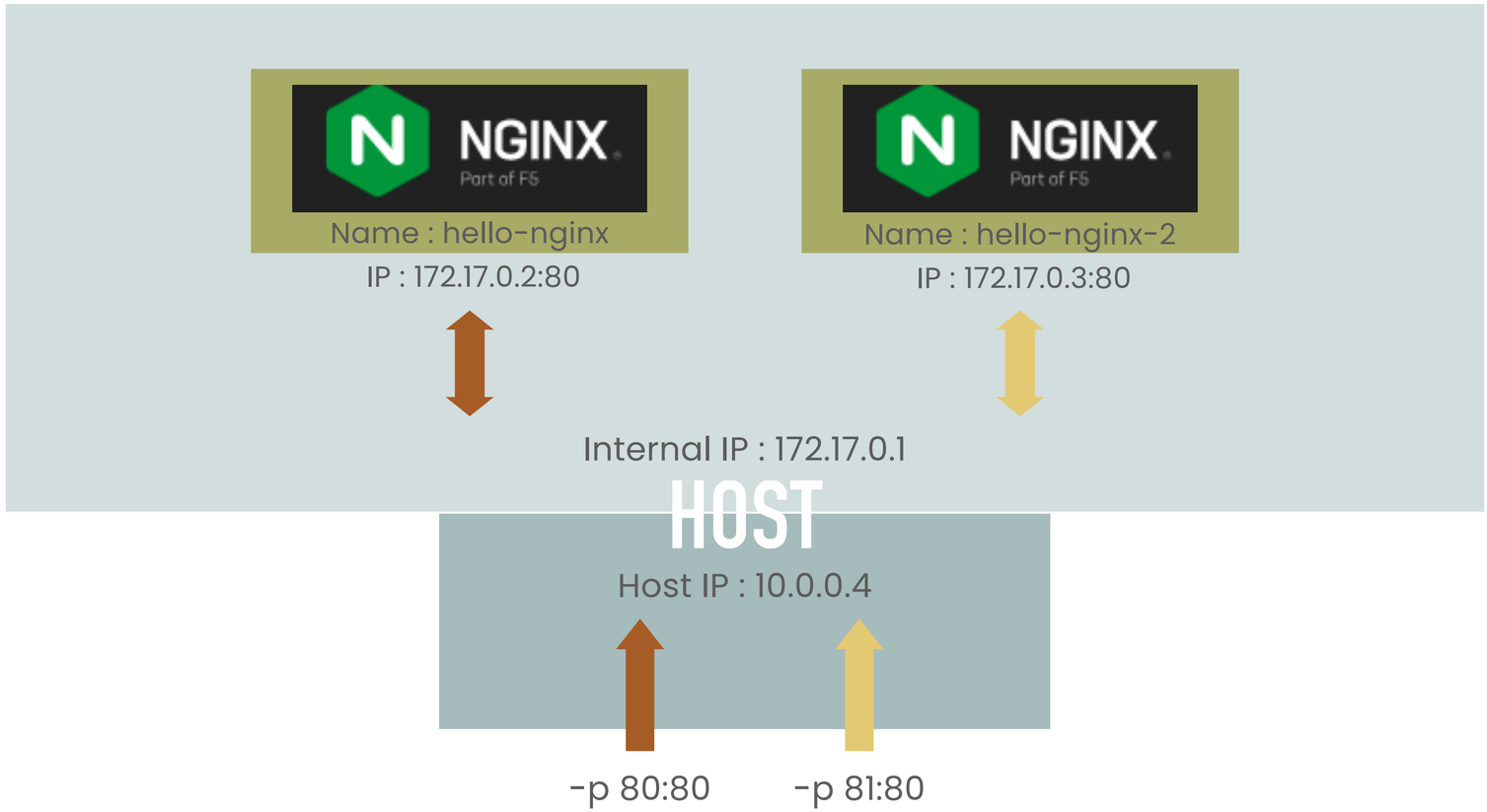
DOCKER - CREATE NEW NGINX CONTAINER

```
$ docker container run --name hello-nginx-2 -d -p 81:80 nginx
```



DOCKER - CREATE NEW NGINX CONTAINER

```
$ docker container run --name hello-nginx-2 -d -p 81:80 nginx
```



CREATE YOUR OWN BRIDGE NETWORK

```
$ docker network inspect bridge
```

CREATE YOUR OWN BRIDGE NETWORK

```
$ docker network inspect bridge
```


```
"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": ""
},
"ConfigOnly": false,
"Containers": {
    "8c6406e9cb0f8decc57e12c1875ca4cfd9d2bc7de833b992e9150f9f555072cf": {
        "Name": "hello-nginx-2",
        "EndpointID": "5f4d9d9b35bab4f6b738881c0036efa91f40ad1ce56294b5a047c450e710cf98",
        "MacAddress": "02:42:ac:11:00:03",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
    },
    "bd087eff1eaf79f353c1448924bed9985438382ff822638dc125b8c7b0a3ea83": {
        "Name": "hello-nginx",
        "EndpointID": "306a62f3249ab8cfad056da39ccc81d93f328bf8151640b539e803a4ad32db6c",
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "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",
```

CREATE YOUR OWN BRIDGE NETWORK

```
$ docker network inspect bridge
```


```
"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": ""
},
"ConfigOnly": false,
"Containers": {
    "8c6406e9cb0f8decc57e12c1875ca4cfd9d2bc7de833b992e9150f9f555072cf": {
        "Name": "hello-nginx-2",
        "EndpointID": "5f4d9d9b35bab4f6b738881c0036efa91f40ad1ce56294b5a047c450e710cf98",
        "MacAddress": "02:42:ac:11:00:03",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
    },
    "bd087eff1eaf79f353c1448924bed9985438
        "Name": "hello-nginx",
        "EndpointID": "306a62f3249ab8cfad056d
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
    }
},
"Options": {
    "com.docker.network.bridge.default_bridg
    "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",
```


IP-Address of hello-nginx


```
"Name": "hello-nginx",
"EndpointID": "306a62f3249ab8cfad056da39ccc8
"MacAddress": "02:42:ac:11:00:02",
"IPv4Address": "172.17.0.2/16",
"IPv6Address": ""
```

CREATE YOUR OWN BRIDGE NETWORK

```
$ docker network inspect bridge
```


```
"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": ""
},
"ConfigOnly": false,
"Containers": {
    "8c6406e9cb0f8decc57e12c1875ca4cfd9d2
        "Name": "hello-nginx-2",
        "EndpointID": "5f4d9d9b35bab4f6b73888
        "MacAddress": "02:42:ac:11:00:03",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
    },
    "bd087eff1eaf79f353c1448924bed9985438
        "Name": "hello-nginx",
        "EndpointID": "306a62f3249ab8cfad056d
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
    }
},
"Options": {
    "com.docker.network.bridge.default_bridg
    "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",
```

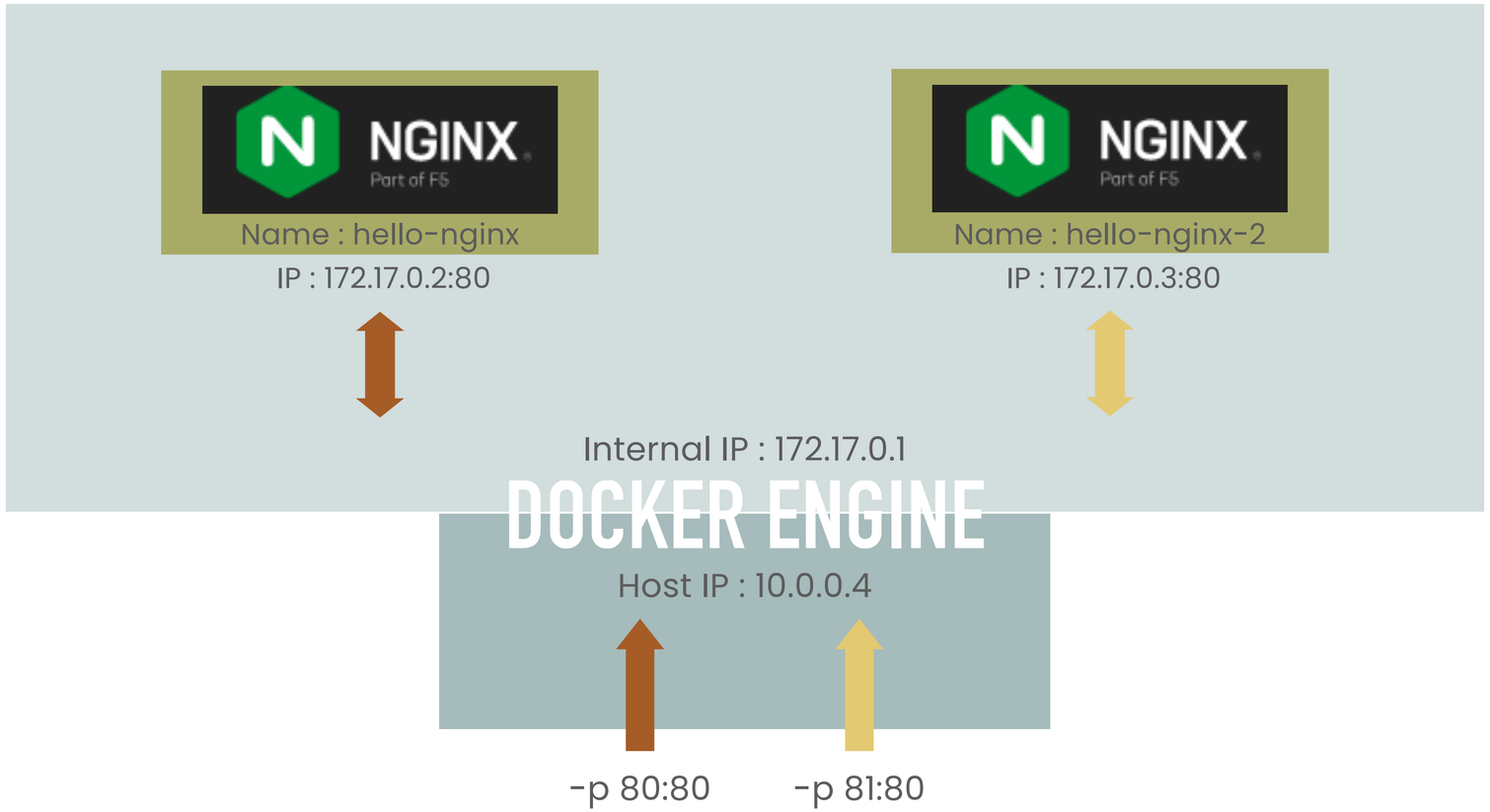

IP-Address of hello-nginx-2


```
"Name": "hello-nginx-2",
"EndpointID": "5f4d9d9b35bab4f6b738881c0036e
"MacAddress": "02:42:ac:11:00:03",
"IPv4Address": "172.17.0.3/16",
"IPv6Address": ""
```


IP-Address of hello-nginx

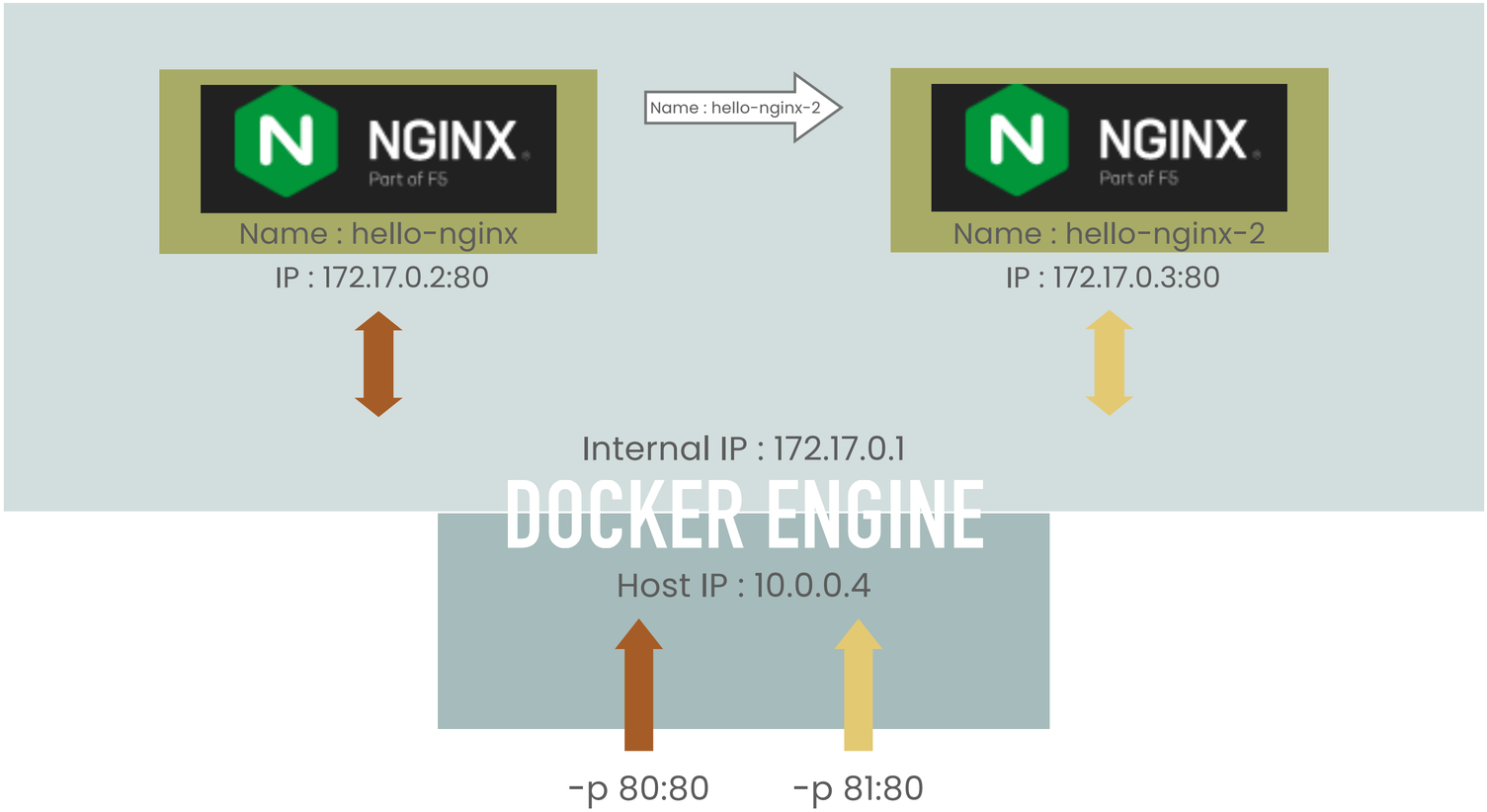

```
"Name": "hello-nginx",
"EndpointID": "306a62f3249ab8cfad056da39ccc8
"MacAddress": "02:42:ac:11:00:02",
"IPv4Address": "172.17.0.2/16",
"IPv6Address": ""
```

DOCKER - NO DNS FOR DEFAULT BRIDGE NETWORK



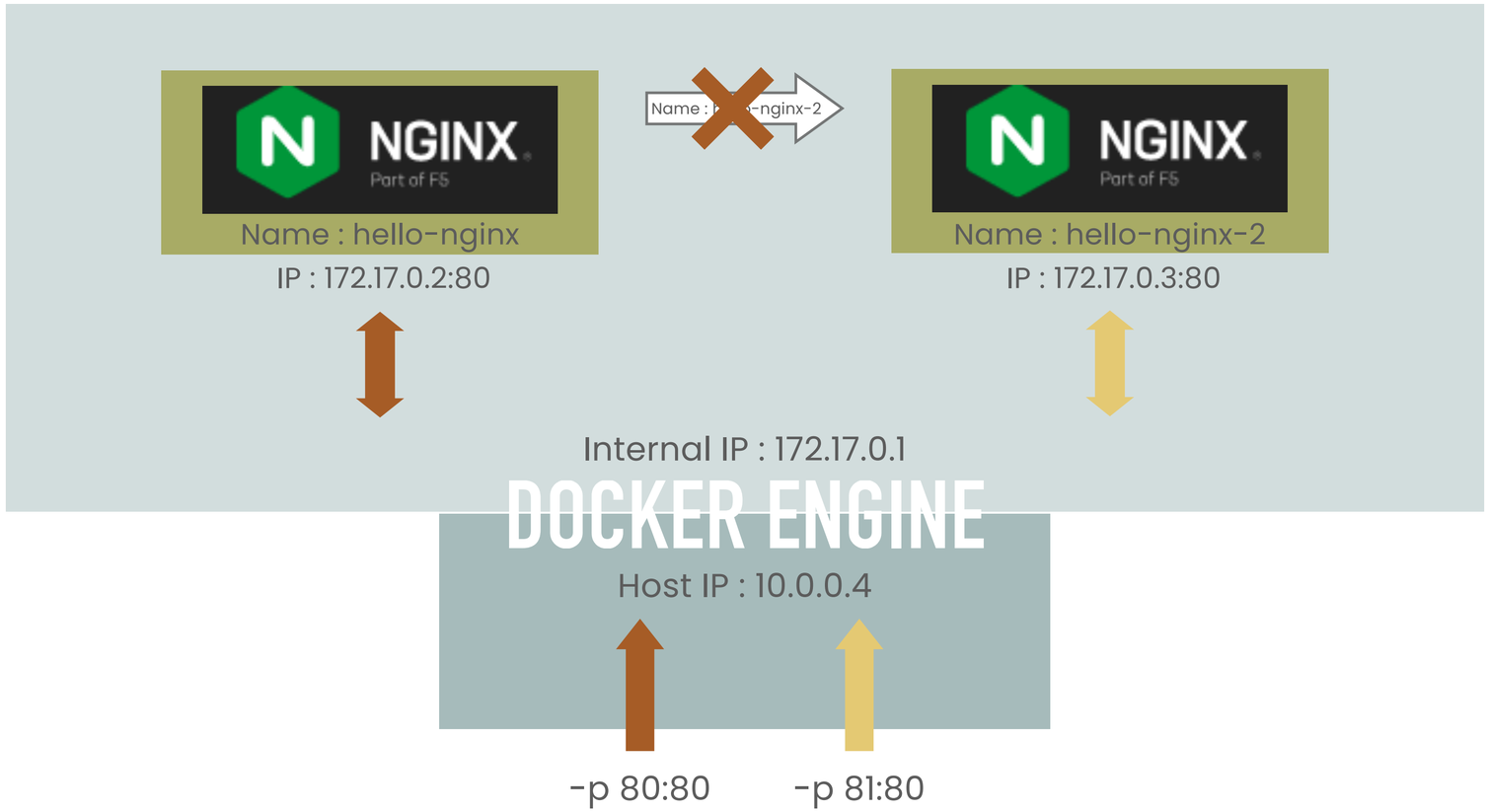
DOCKER - NO DNS FOR DEFAULT BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx curl hello-nginx-2
```



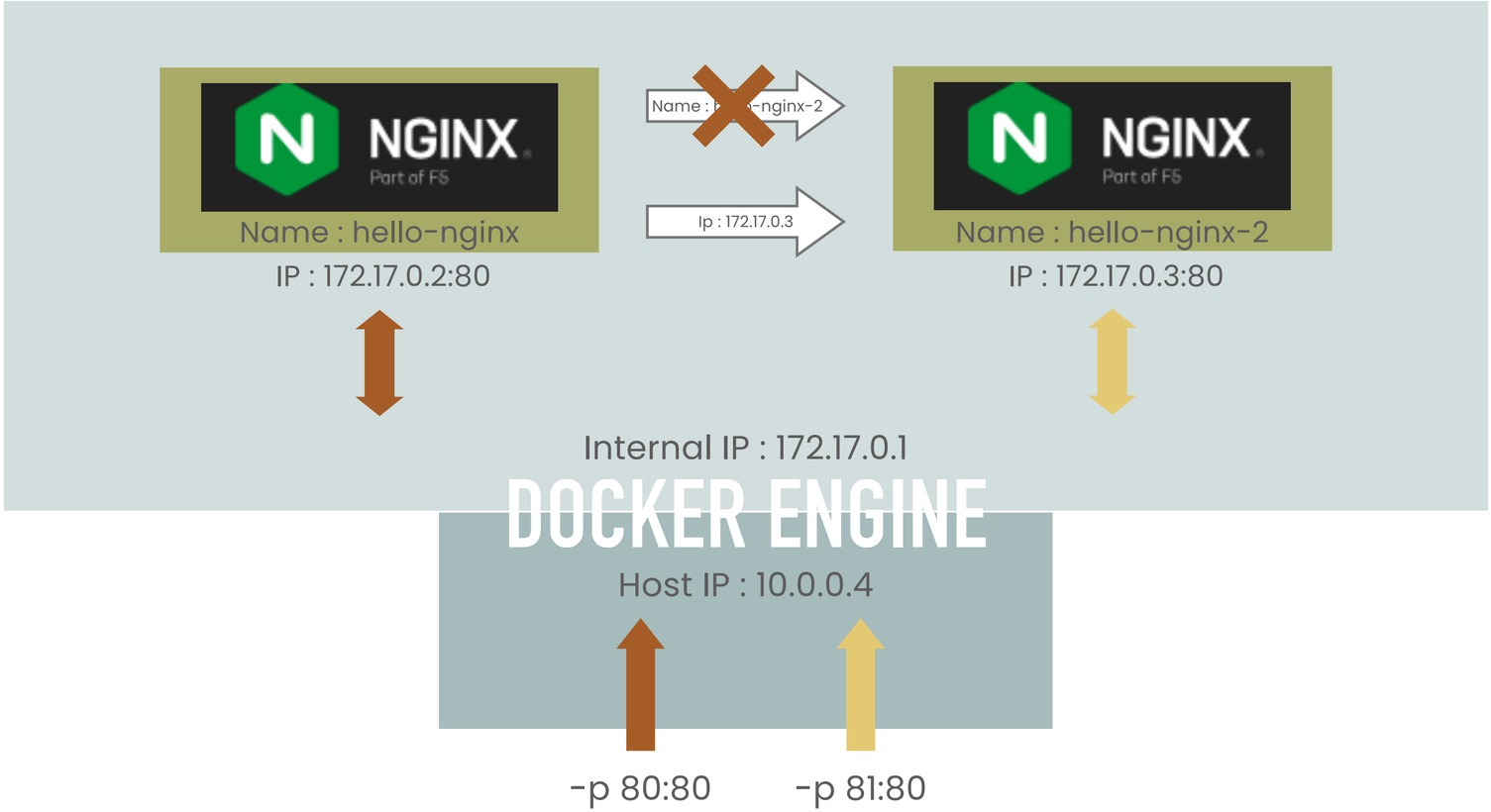
DOCKER - NO DNS FOR DEFAULT BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx curl hello-nginx-2
```



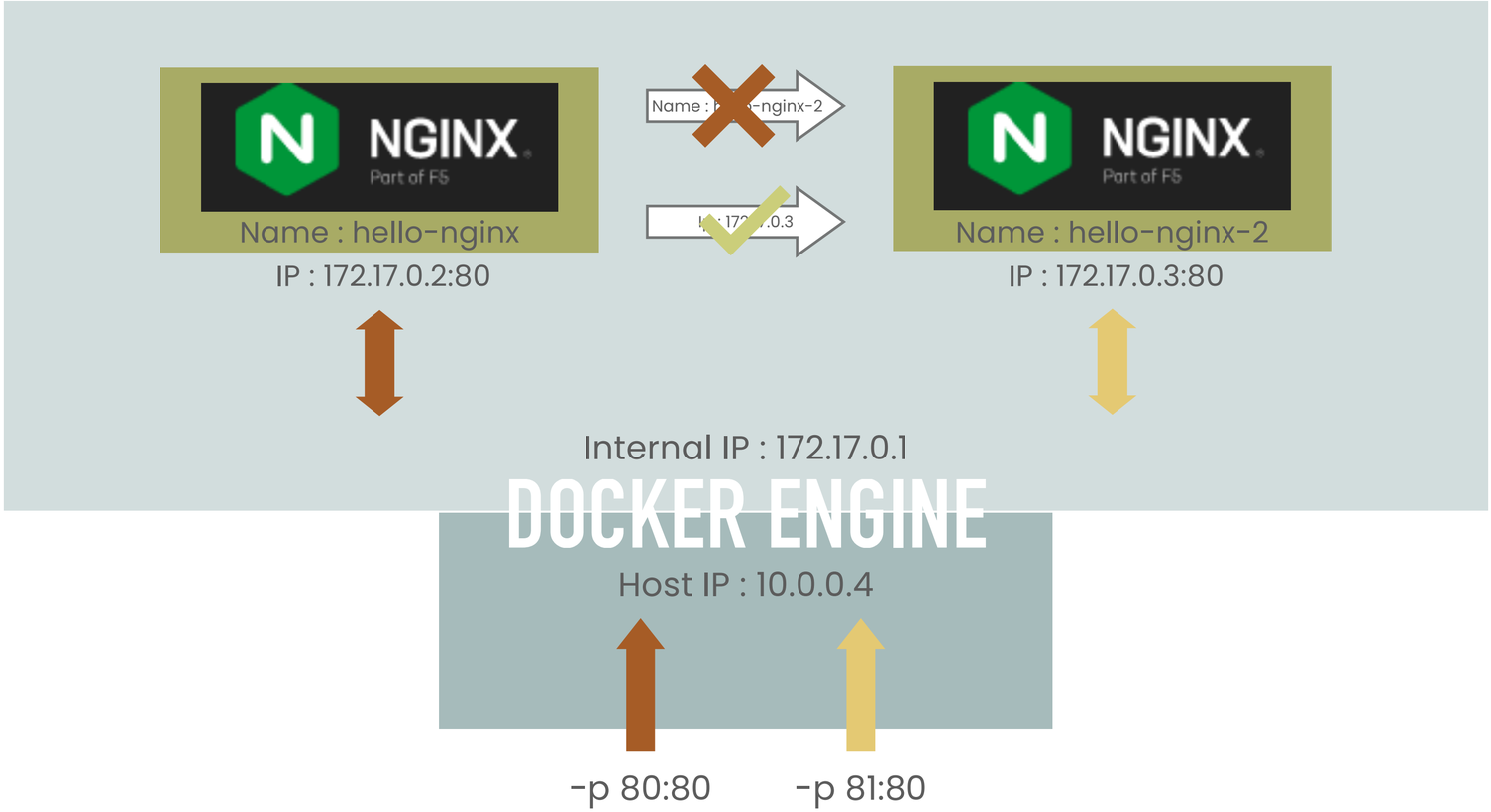
DOCKER - NO DNS FOR DEFAULT BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx curl 172.17.0.3
```



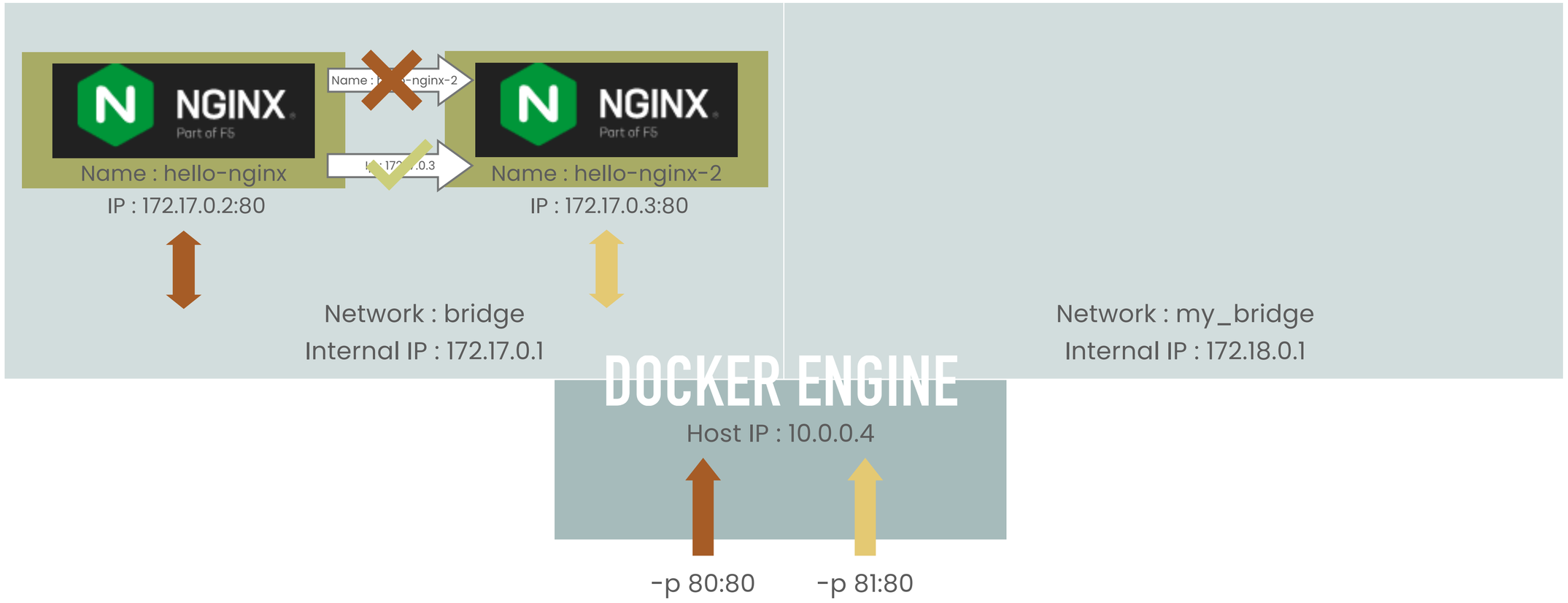
DOCKER - NO DNS FOR DEFAULT BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx curl 172.17.0.3 ✓
```



CREATE YOUR OWN (USER-DEFINED) BRIDGE NETWORK

```
$ docker network create -d bridge my_bridge
```



CREATE YOUR OWN (USER-DEFINED) BRIDGE NETWORK

CREATE YOUR OWN (USER-DEFINED) BRIDGE NETWORK

```
$ docker network ls
```

```
NETWORK ID     NAME        DRIVER    SCOPE
5c3bb2502623   bridge      bridge    local
6877ad312030   host        host      local
900ed84b8b07   my_bridge   bridge    local
d3d51f296ce4   none        null      local
```

CREATE YOUR OWN (USER-DEFINED) BRIDGE NETWORK

$ docker network ls

```
NETWORK ID     NAME        DRIVER    SCOPE
5c3bb2502623   bridge      bridge    local
6877ad312030   host        host      local
900ed84b8b07   my_bridge   bridge    local
d3d51f296ce4   none        null      local
```

$ docker network inspect my_bridge

```
C:\Users\karan>docker network inspect my_bridge
[
    {
        "Name": "my_bridge",
        "Id": "900ed84b8b071491d929553deaa44844d88a859c2e79f0b0e203bf33f3f26d6b",
        "Created": "2021-05-26T18:20:04.8395706Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
```

DOCKER INTERNAL NETWORK

```
"Config": [
    {
        "Subnet": "172.18.0.0/16",
        "Gateway": "172.18.0.1"
    }
]
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

```
$ docker network inspect my_bridge
```

```
C:\Users\karan>docker network inspect my_bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "22da2499c01e42465eedef4b6fb12826e866845c2efaaa978040b3de8a7aa13d": {
                "Name": "hello-nginx-4",
                "EndpointID": "9d7bd8571111806849d7730460ccfcffa700bfd1e79f06f87f625be45513432e",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "416bdb1eb345380ad67605d9c1b51c666110ef65edfe4be54d6ea91d32770687": {
                "Name": "hello-nginx-3",
                "EndpointID": "4b1b43a31cf13c3c655254569a8a754190297997352f550f606ae4e5e80c185b",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

```
$ docker network inspect my_bridge
```

```
C:\Users\karan>docker network inspect my_bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "22da2499c01e42465eedef4b6fb12826e866845c2efaaa978040b3de8a7aa13d": {
                "Name": "hello-nginx-4",
                "EndpointID": "9d7bd8571111806849d7730460ccfcffa700bfd1e79f06f87f625be45513432e",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "416bdb1eb345380ad67605d9c1b51c666110ef65edfe4be54d6ea91d32770687": {
                "Name": "hello-nginx-3",
                "EndpointID": "4b1b43a31cf13c3c655254569a8a754190297997352f550f606ae4e5e80c185b",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

```
$ docker network inspect my_bridge
```

```
C:\Users\karan>docker network inspect my_bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "22da2499c01e42465eedef4b6fb12826e866845c2efaaa978040b3de8a7aa13d": {
                "Name": "hello-nginx-4",
                "EndpointID": "9d7bd8571111806849d7730460ccfcffa700bfd1e79f06f87f625be45513432e",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "416bdb1eb345380ad67605d9c1b51c666110ef65edfe4be54d6ea91d32770687": {
                "Name": "hello-nginx-3",
                "EndpointID": "4b1b43a31cf13c3c655254569a8a754190297997352f550f606ae4e5e80c185b",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

$ docker network inspect my_bridge

```
C:\Users\karan>docker network inspect my_bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "22da2499c01e42465eedef4b6fb12826e866845c2efaaa978040b3de8a7aa13d": {
                "Name": "hello-nginx-4",
                "EndpointID": "9d7bd8571111806849d7730460ccfcffa700bfd1e79f06f87f625be45513432e",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "416bdb1eb345380ad67605d9c1b51c666110ef65edfe4be54d6ea91d32770687": {
                "Name": "hello-nginx-3",
                "EndpointID": "4b1b43a31cf13c3c655254569a8a754190297997352f550f606ae4e5e80c185b",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

$ docker network inspect bridge

```
C:\Users\karan>docker network inspect bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "8c6406e9cb0f8decc57e12c1875ca4cfd9d2bc7de833b992e9150f9f555072cf": {
                "Name": "hello-nginx-2",
                "EndpointID": "5f4d9d9b35bab4f6b738881c0036efa91f40ad1ce56294b5a047c450e710cf98",
                "MacAddress": "02:42:ac:11:00:03",
                "IPv4Address": "172.17.0.3/16",
                "IPv6Address": ""
            },
            "bd087eff1eaf79f353c1448924bed9985438382ff822638dc125b8c7b0a3ea83": {
                "Name": "hello-nginx",
                "EndpointID": "306a62f3249ab8cfad056da39ccc81d93f328bf8151640b539e803a4ad32db6c",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "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",
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

$ docker network inspect my_bridge

```
C:\Users\karan>docker network inspect my_bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "22da2499c01e42465eedef4b6fb12826e866845c2efaaa978040b3de8a7aa13d": {
                "Name": "hello-nginx-4",
                "EndpointID": "9d7bd8571111806849d7730460ccfcffa700bfd1e79f06f87f625be45513432e",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "416bdb1eb345380ad67605d9c1b51c666110ef65edfe4be54d6ea91d32770687": {
                "Name": "hello-nginx-3",
                "EndpointID": "4b1b43a31cf13c3c655254569a8a754190297997352f550f606ae4e5e80c185b",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

$ docker network inspect bridge

```
C:\Users\karan>docker network inspect bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "8c6406e9cb0f8decc57e12c1875ca4cfd9d2bc7de833b992e9150f9f555072cf": {
                "Name": "hello-nginx-2",
                "EndpointID": "5f4d9d9b35bab4f6b73888[illegible]c0036efa91f40ad1ce56294b5a047c450e710cf98",
                "MacAddress": "02:42:ac:11:00:03",
                "IPv4Address": "172.17.0.3/16",
                "IPv6Address": ""
            },
            "bd087eff1eaf79f353c1448924bed9985438382ff822638dc125b8c7b0a3ea83": {
                "Name": "hello-nginx",
                "EndpointID": "306a62f3249ab8cfad056da39ccc81d93f328bf8151640b539e803a4ad32db6c",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "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",
```

ADD CONTAINER TO USER-DEFIND BRIDGE NETWORK

```
$ docker container run —net=my_bridge --name hello-nginx-3 -d -p 83:80 nginx
```

```
$ docker container run —net=my_bridge --name hello-nginx-4 -d -p 84:80 nginx
```

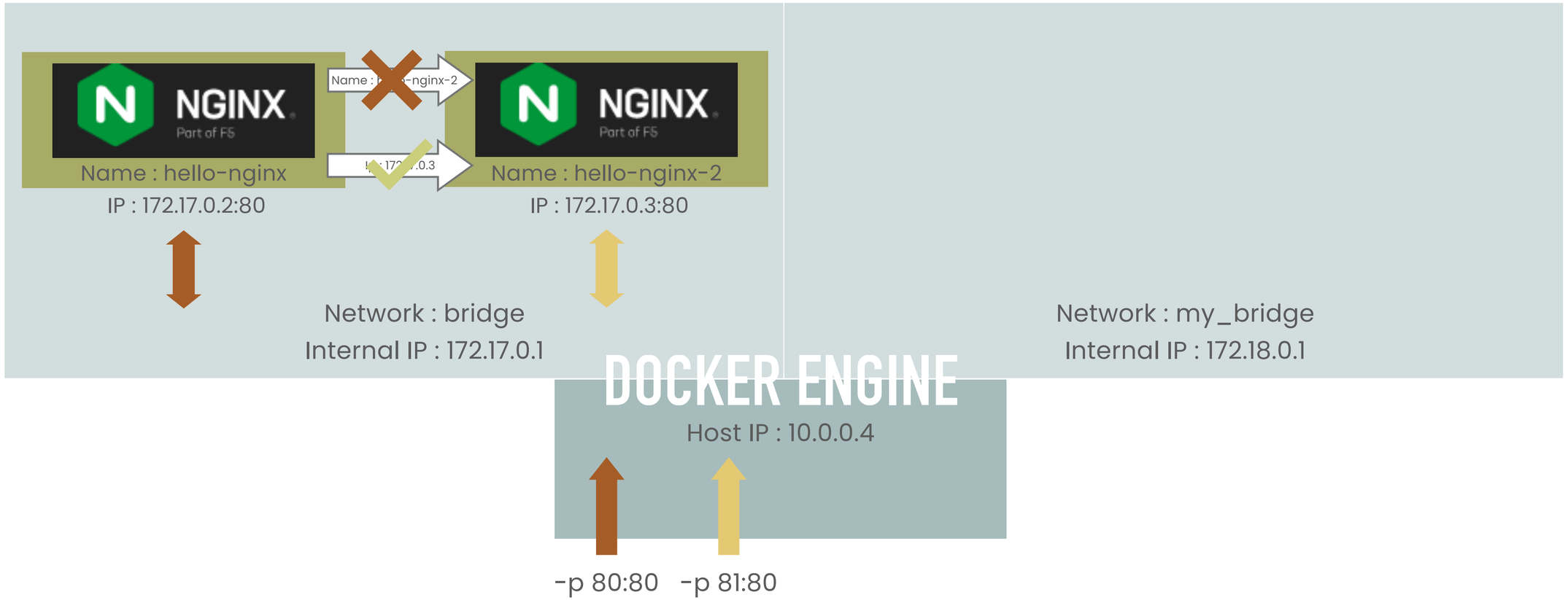
$ docker network inspect my_bridge

```
C:\Users\karan>docker network inspect my_bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "22da2499c01e42465eedef4b6fb12826e866845c2efaaa978040b3de8a7aa13d": {
                "Name": "hello-nginx-4",
                "EndpointID": "9d7bd8571111806849d7730460ccfcffa700bfd1e79f06f87f625be45513432e",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            },
            "416bdb1eb345380ad67605d9c1b51c666110ef65edfe4be54d6ea91d32770687": {
                "Name": "hello-nginx-3",
                "EndpointID": "4b1b43a31cf13c3c655254569a8a754190297997352f550f606ae4e5e80c185b",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

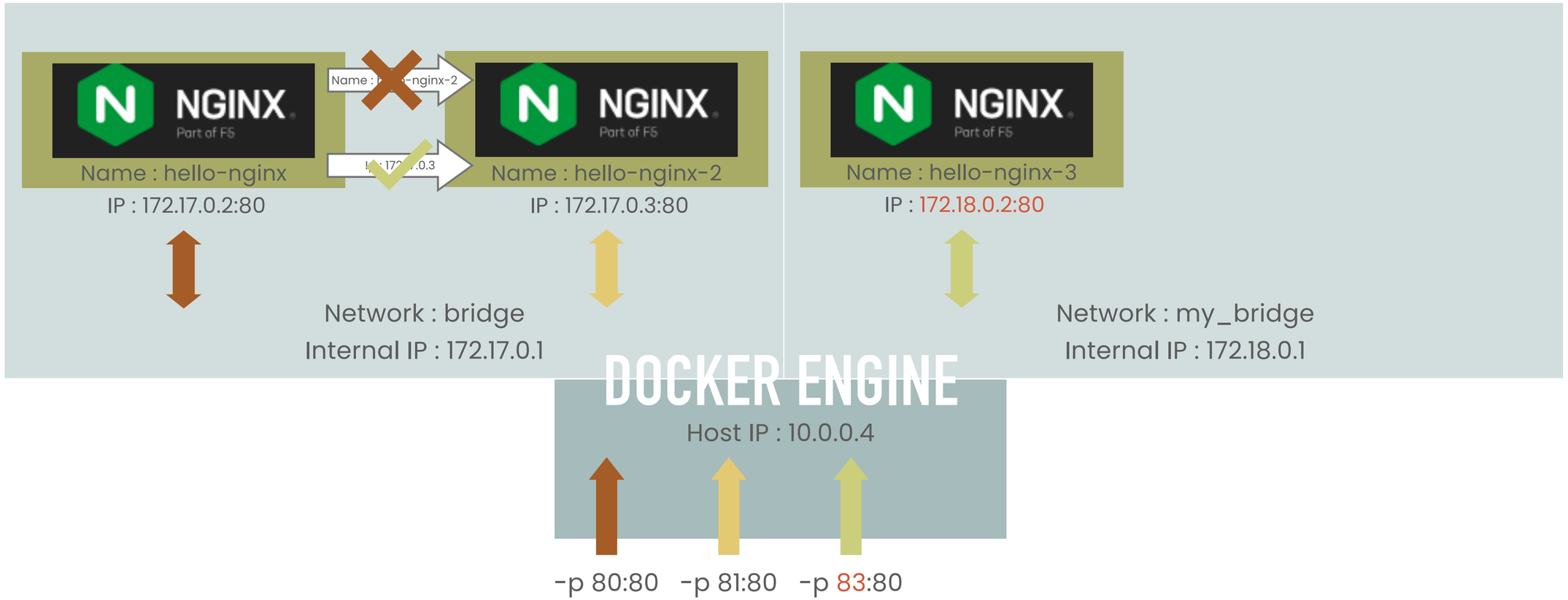
$ docker network inspect bridge

```
C:\Users\karan>docker network inspect bridge
...
...
...
        "ConfigOnly": false,
        "Containers": {
            "8c6406e9cb0f8decc57e12c1875ca4cfd9d2bc7de833b992e9150f9f555072cf": {
                "Name": "hello-nginx-2",
                "EndpointID": "5f4d9d9b35bab4f6b73888[illegible]c0036efa91f40ad1ce56294b5a047c450e710cf98",
                "MacAddress": "02:42:ac:11:00:03",
                "IPv4Address": "172.17.0.3/16",
                "IPv6Address": ""
            },
            "bd087eff1eaf79f353c1448924bed9985438382ff822638dc125b8c7b0a3ea83": {
                "Name": "hello-nginx",
                "EndpointID": "306a62f3249ab8cfad056d[illegible]39ccc81d93f328bf8151640b539e803a4ad32db6c",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "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",
```

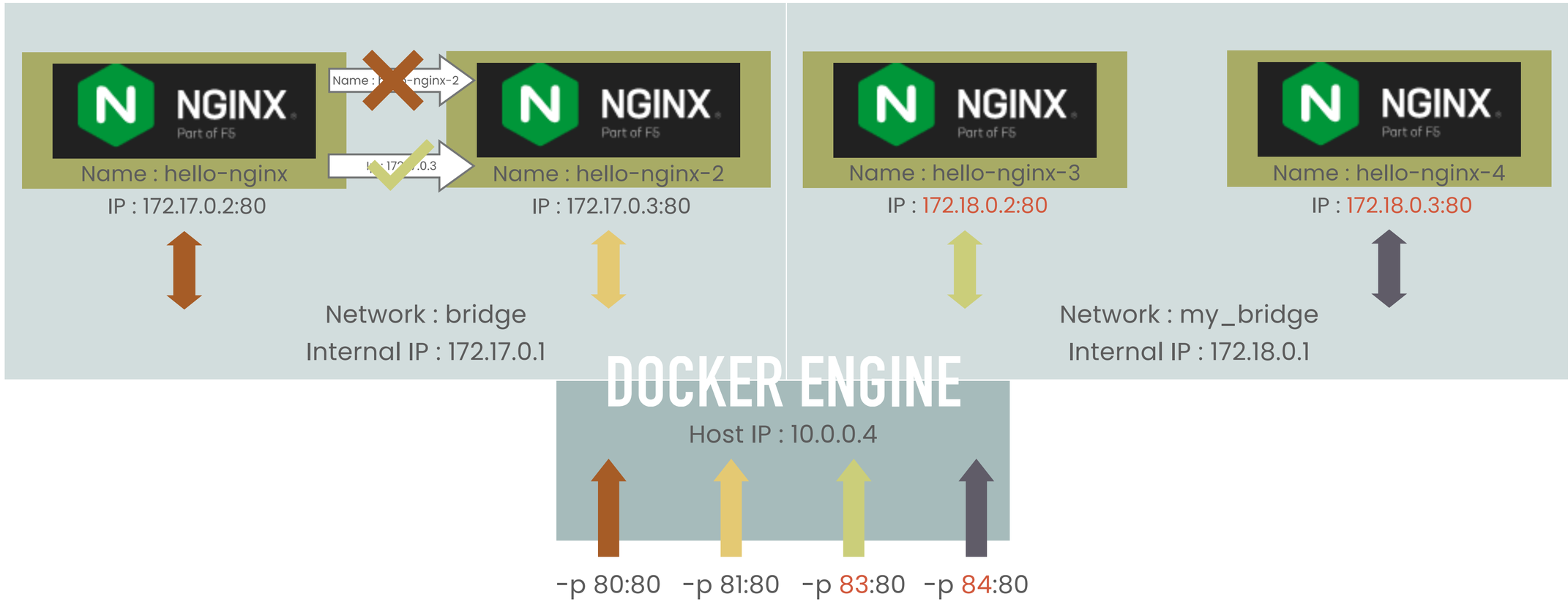
ADD CONTAINER TO USER-DEFINED BRIDGE NETWORK



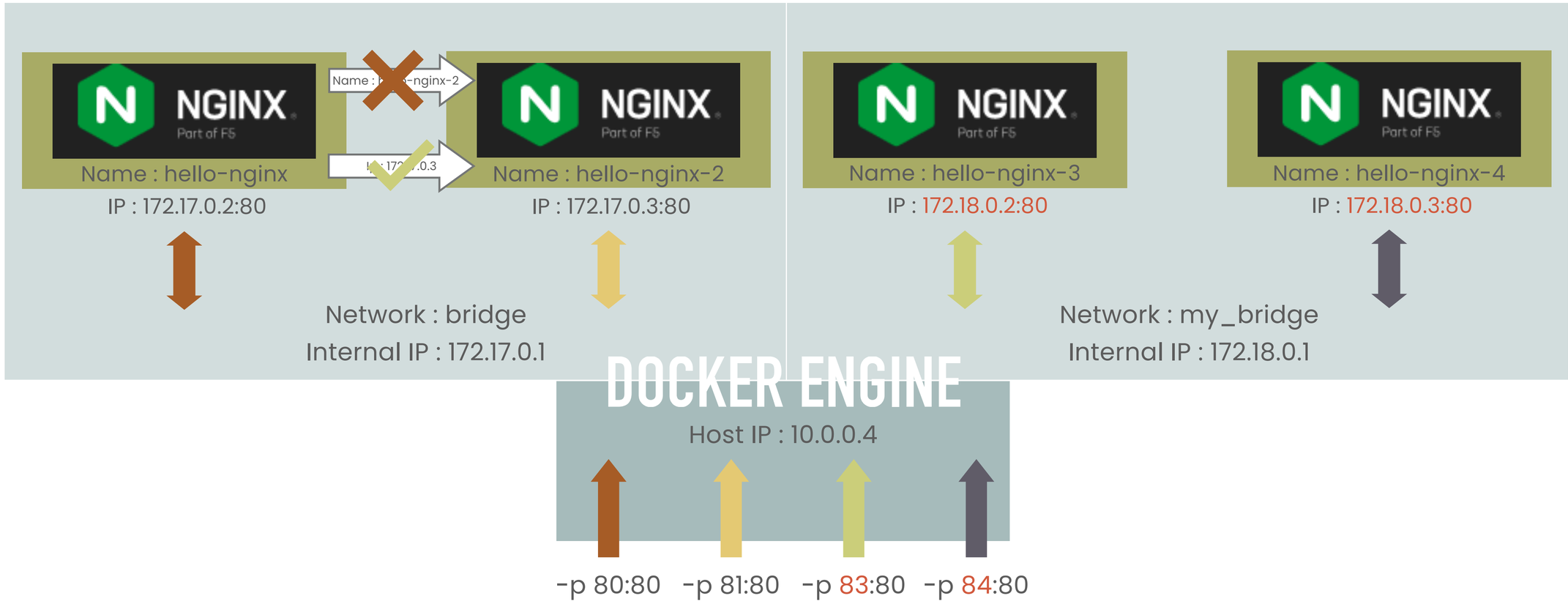
ADD CONTAINER TO USER-DEFINED BRIDGE NETWORK



ADD CONTAINER TO USER-DEFINED BRIDGE NETWORK

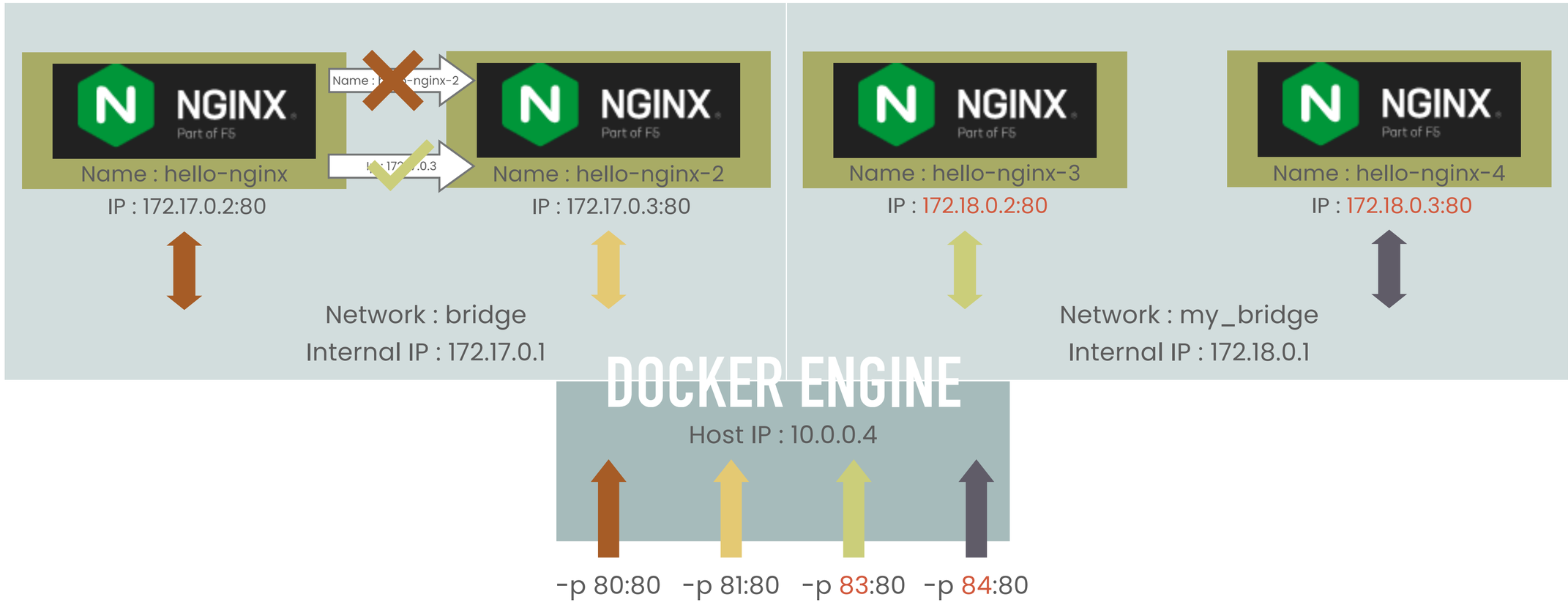


DNS IN USER-DEFINED BRIDGE NETWORK



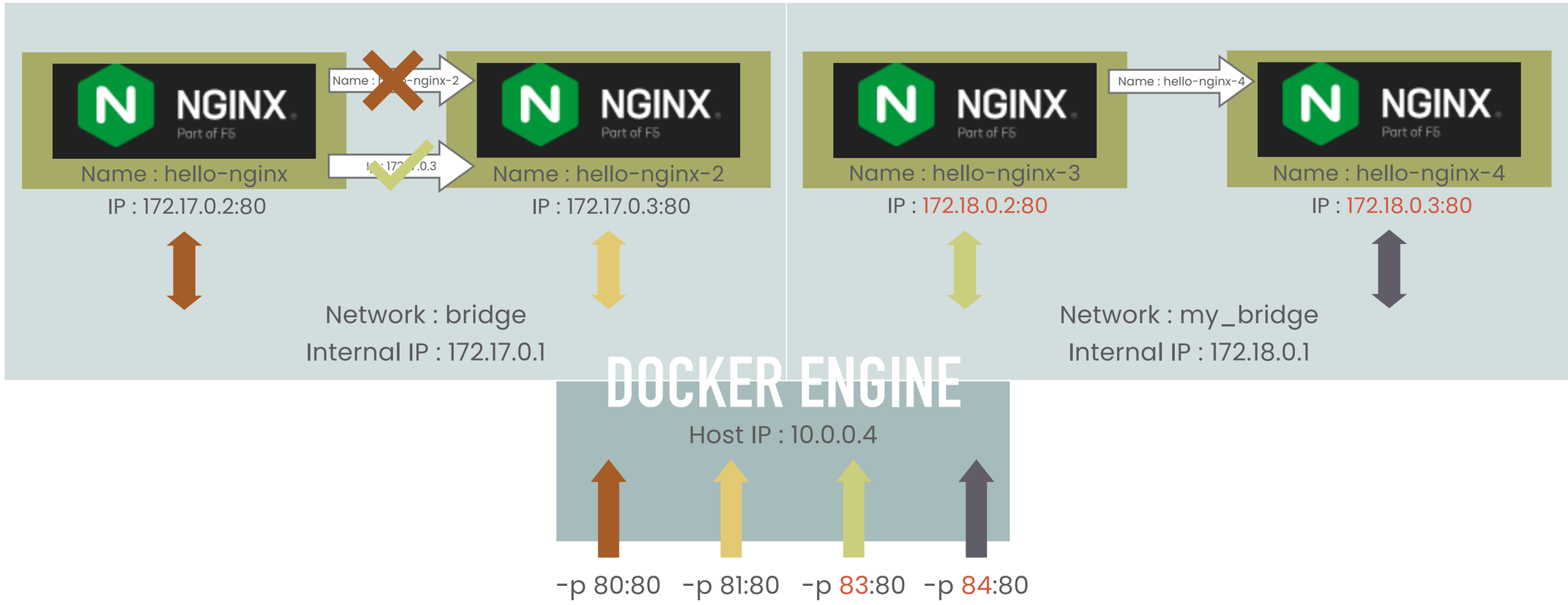
DNS IN USER-DEFINED BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx-3 curl hello-nginx-4
```



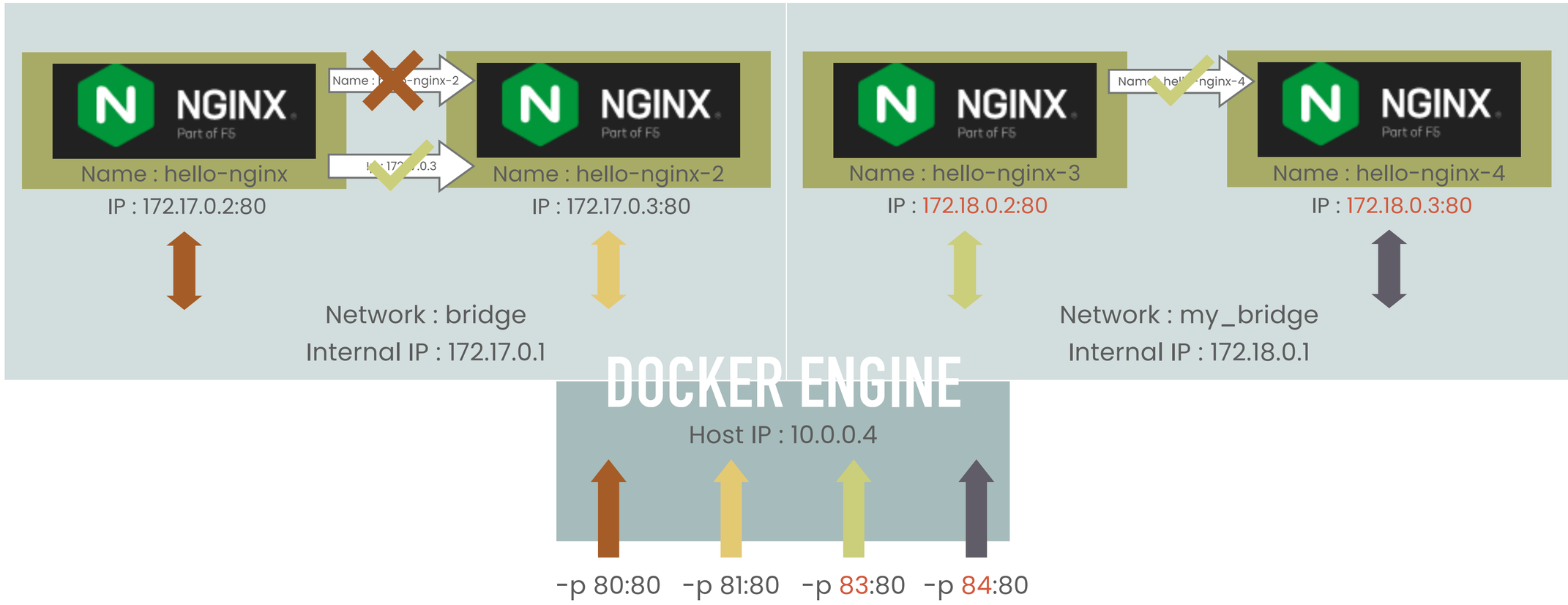
DNS IN USER-DEFINED BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx-3 curl hello-nginx-4
```



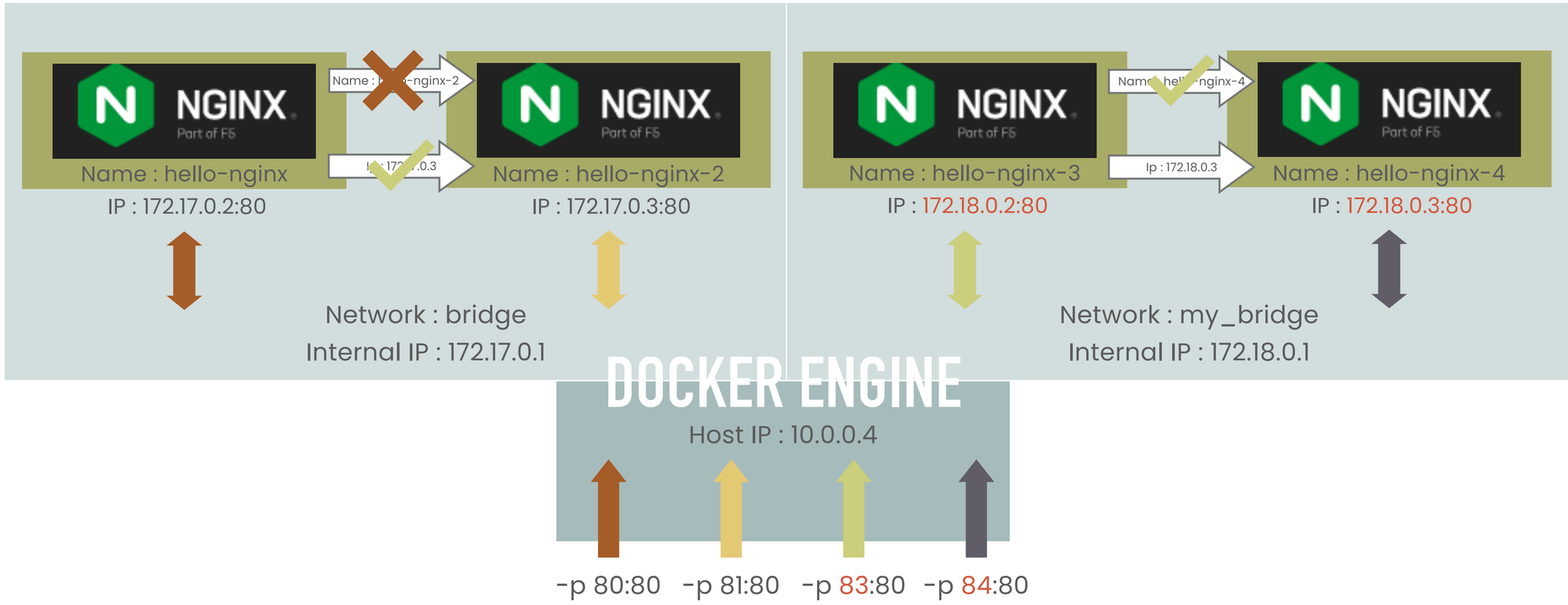
DNS IN USER-DEFINED BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx-3 curl hello-nginx-4
```



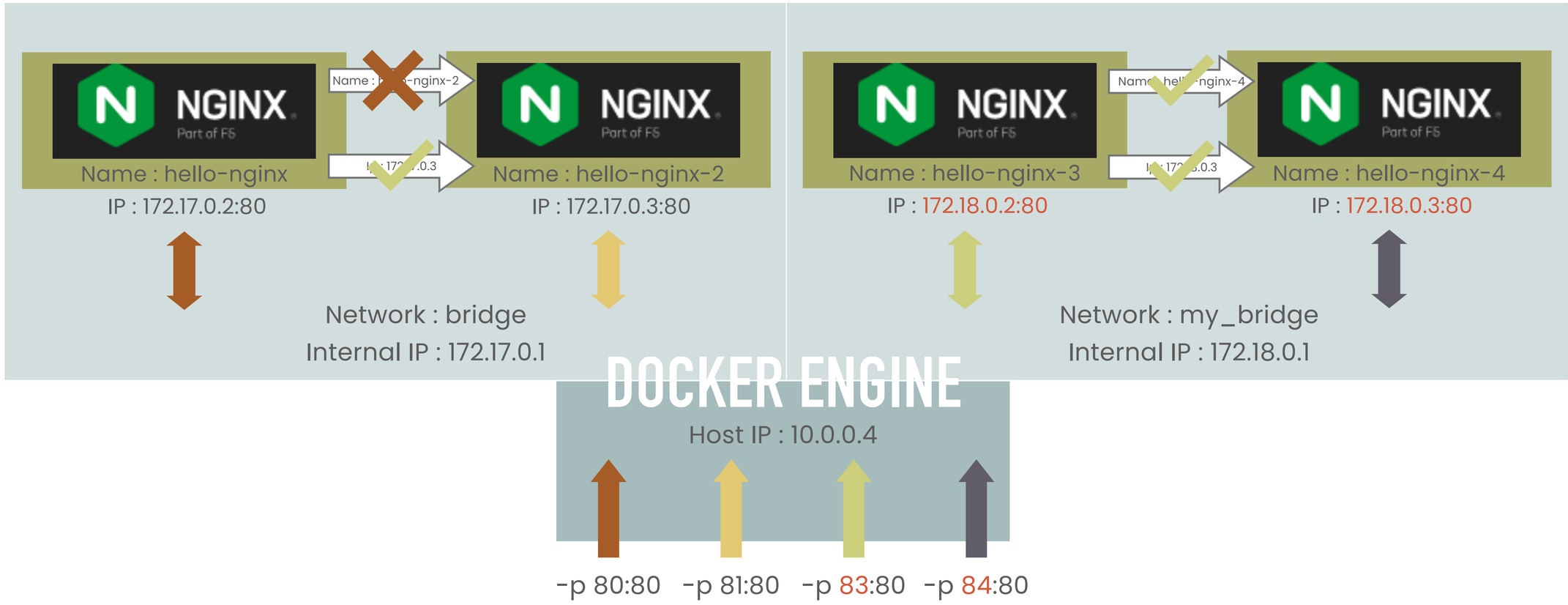
DNS IN USER-DEFINED BRIDGE NETWORK

```
$ docker container exec  -it hello-nginx-3 curl hello-nginx-4
```

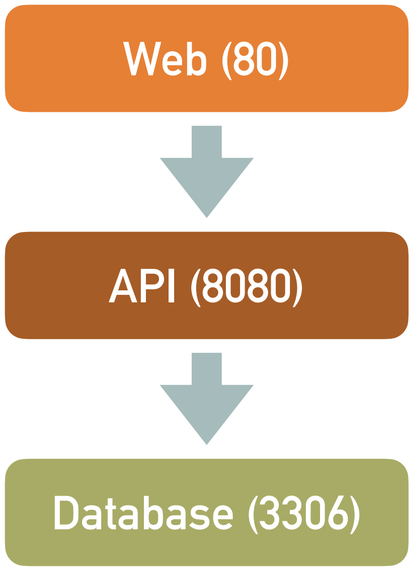


DNS IN USER-DEFINED BRIDGE NETWORK

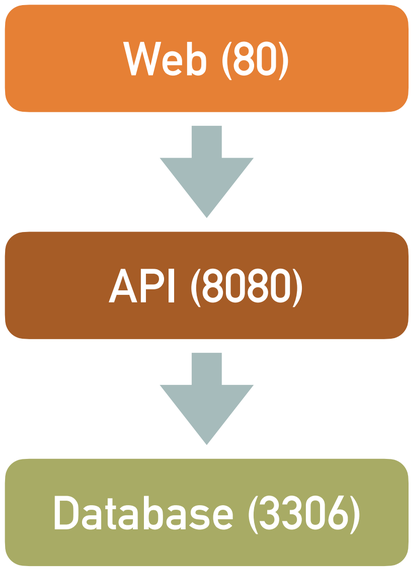
```
$ docker container exec  -it hello-nginx-3 curl hello-nginx-4
```



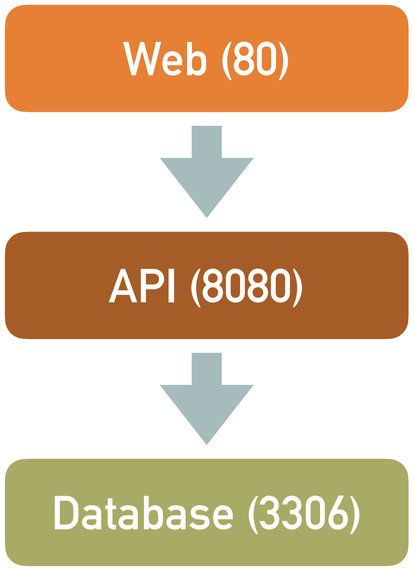
EXERCISE : HOW CAN I SETUP THIS ARCHITECTURE IN DOCKER ?



EXERCISE : HOW CAN I SETUP THIS ARCHITECTURE IN DOCKER ?



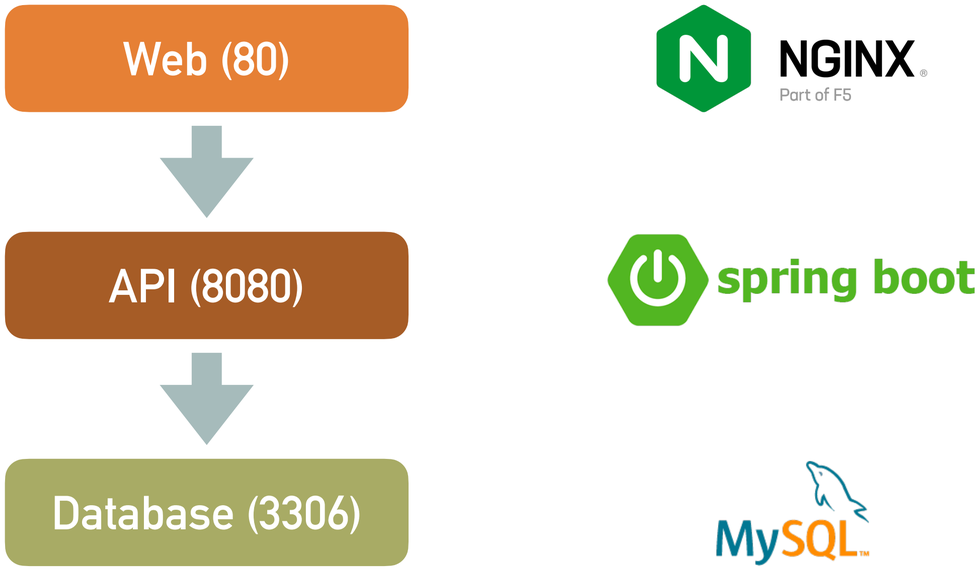
EXERCISE : HOW CAN I SETUP THIS ARCHITECTURE IN DOCKER ?



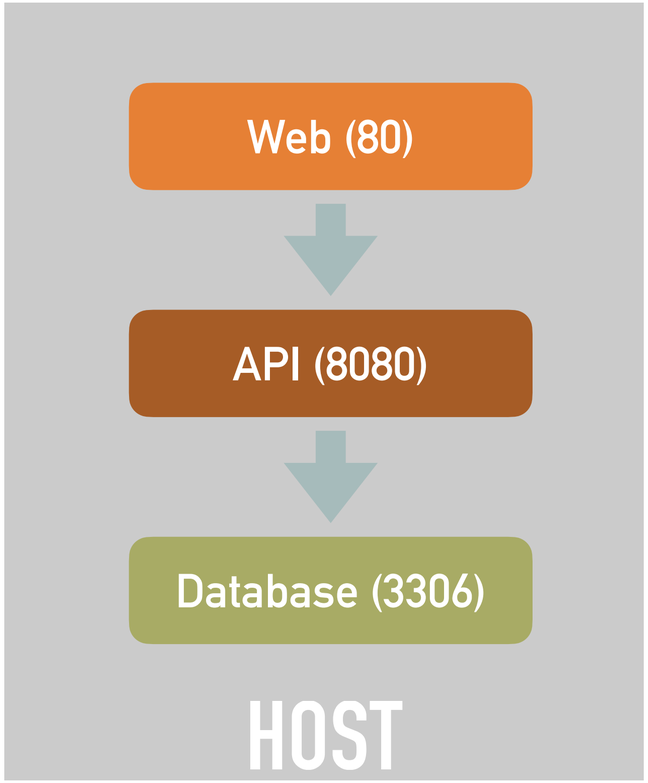
NGINX
Part of F5

spring boot

EXERCISE : HOW CAN I SETUP THIS ARCHITECTURE IN DOCKER ?



EXERCISE : HOW CAN I SETUP THIS ARCHITECTURE IN DOCKER ?

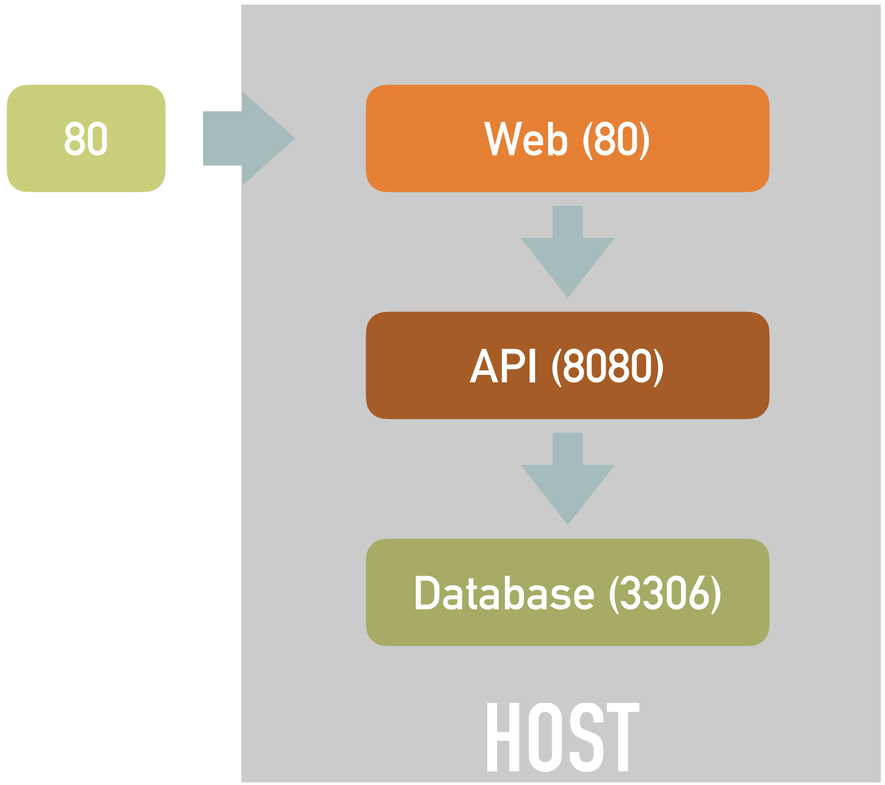


NGINX
Part of F5

spring boot

MySQL

EXERCISE : HOW CAN I SETUP THIS ARCHITECTURE IN DOCKER ?



NGINX
Part of F5

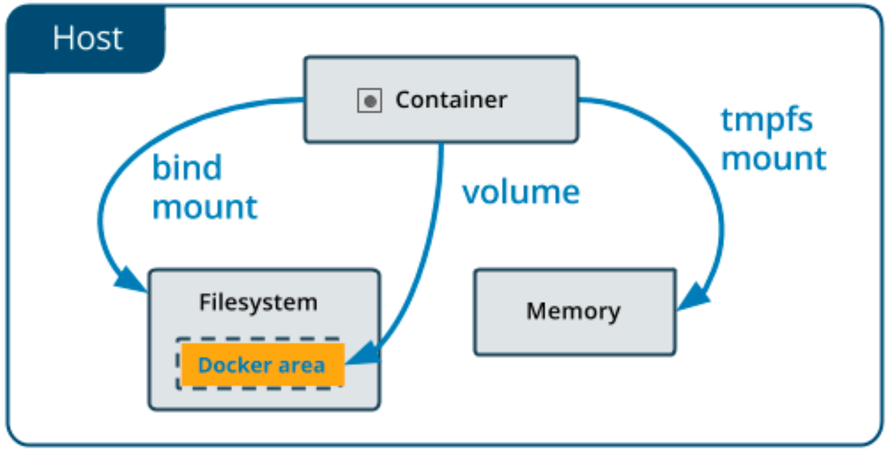
spring boot

MySQL

WORKING WITH VOLUMES

DOCKER VOLUMES

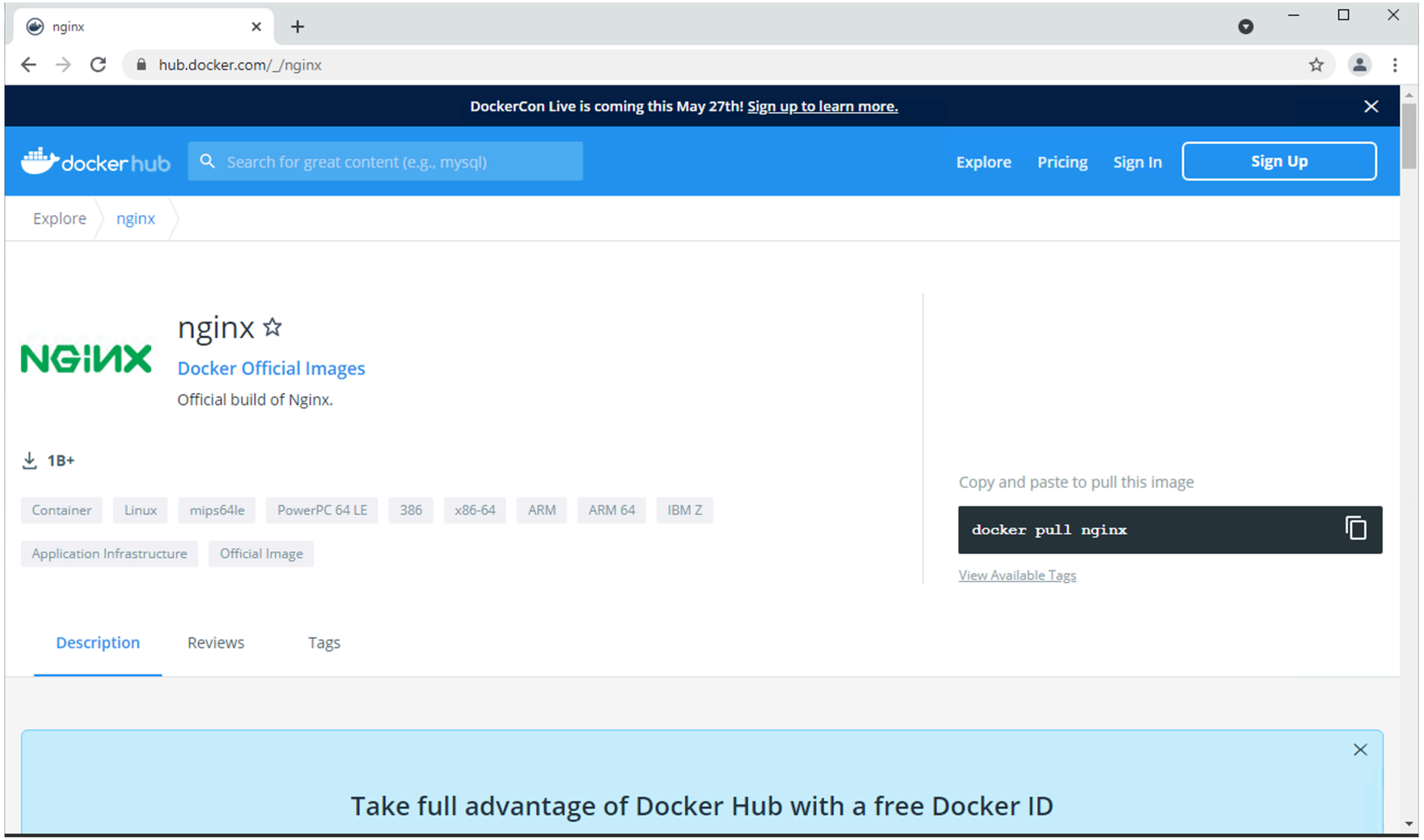
Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. While bind mounts are dependent on the directory structure and OS of the host machine, volumes are completely managed by Docker.



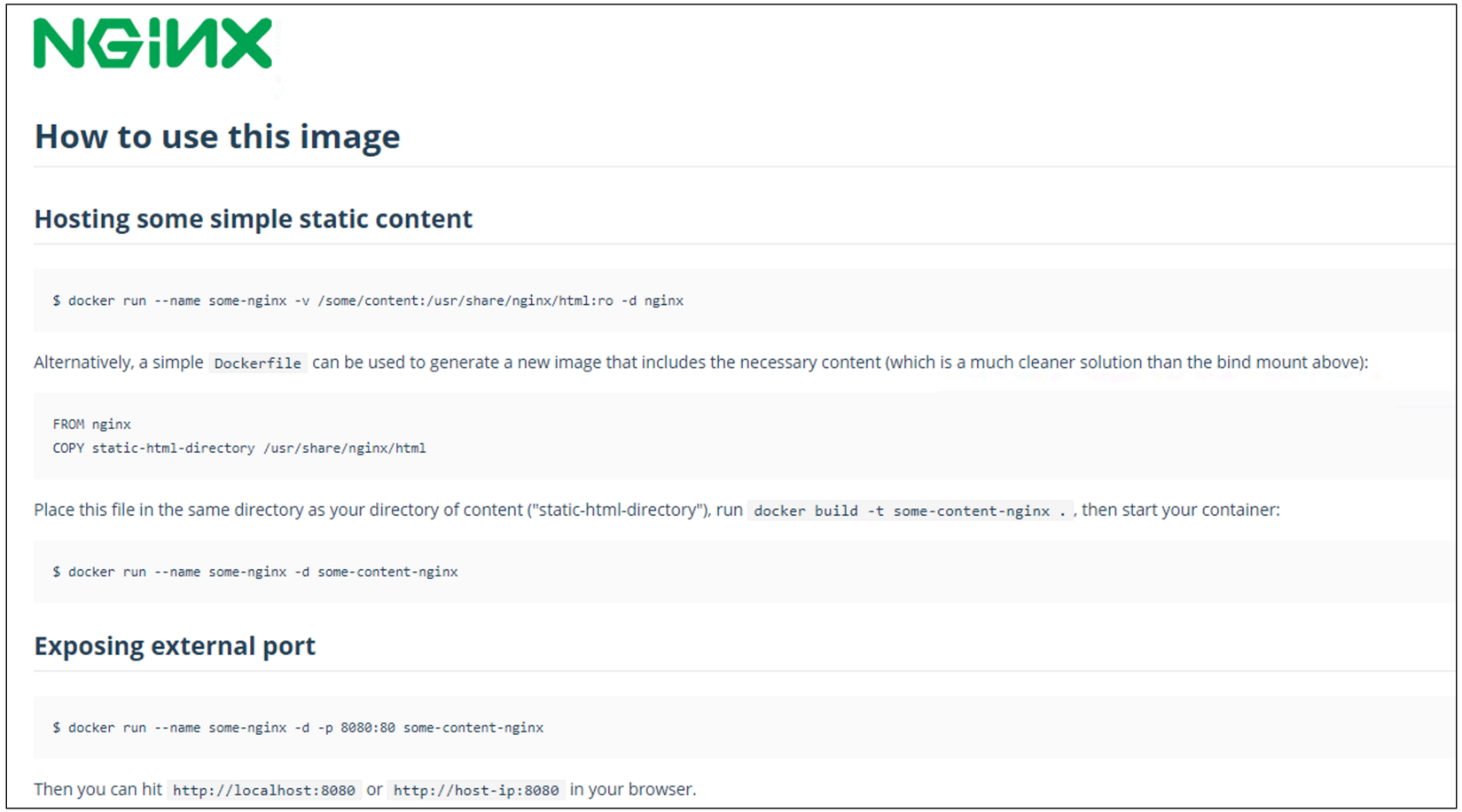
ADVANTAGES OF VOLUMES

- Volumes are easier to back up or migrate than bind mounts.
- You can manage volumes using Docker CLI commands or the Docker API.
- Volumes work on both Linux and Windows containers.
- Volumes can be more safely shared among multiple containers.
- Volume drivers let you store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
- New volumes can have their content pre-populated by a container.
- Volumes on Docker Desktop have much higher performance than bind mounts from Mac and Windows hosts.

NGINX WITH VOLUMES



HOW TO USE NGINX DOCKER IMAGE



How to use this image

Hosting some simple static content

```
$ docker run --name some-nginx -v /some/content:/usr/share/nginx/html:ro -d nginx
```

Alternatively, a simple `Dockerfile` can be used to generate a new image that includes the necessary content (which is a much cleaner solution than the bind mount above):

```
FROM nginx
COPY static-html-directory /usr/share/nginx/html
```

Place this file in the same directory as your directory of content ("static-html-directory"), run `docker build -t some-content-nginx .`, then start your container:

```
$ docker run --name some-nginx -d some-content-nginx
```

Exposing external port

```
$ docker run --name some-nginx -d -p 8080:80 some-content-nginx
```

Then you can hit `http://localhost:8080` or `http://host-ip:8080` in your browser.

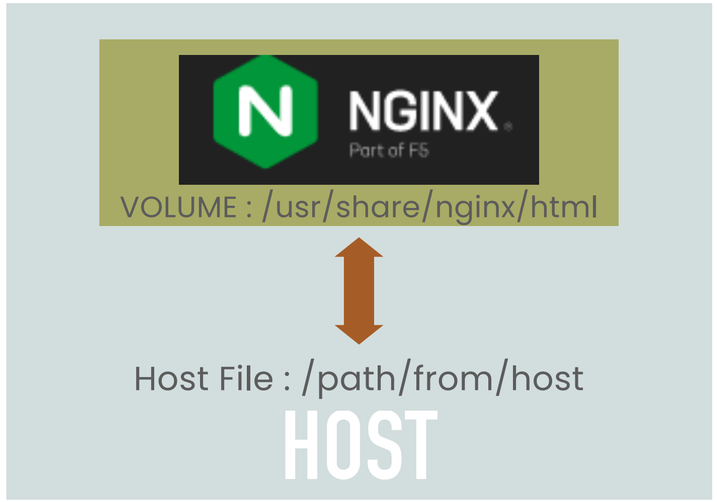
VOLUME – HOST AND CONTAINER

VOLUME – HOST AND CONTAINER

VOLUME - HOST AND CONTAINER

VOLUME - HOST AND CONTAINER

VOLUME - HOST AND CONTAINER



CREATE STATIC WEBSITE FOR NGINX

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

Only for Powershell

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

Only for Powershell

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

Only for Powershell

Host File/Directory

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

Only for Powershell

Host File/Directory

CREATE STATIC WEBSITE FOR NGINX

```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

Only for Powershell

Host File/Directory

Container File/Directory

CREATE STATIC WEBSITE FOR NGINX

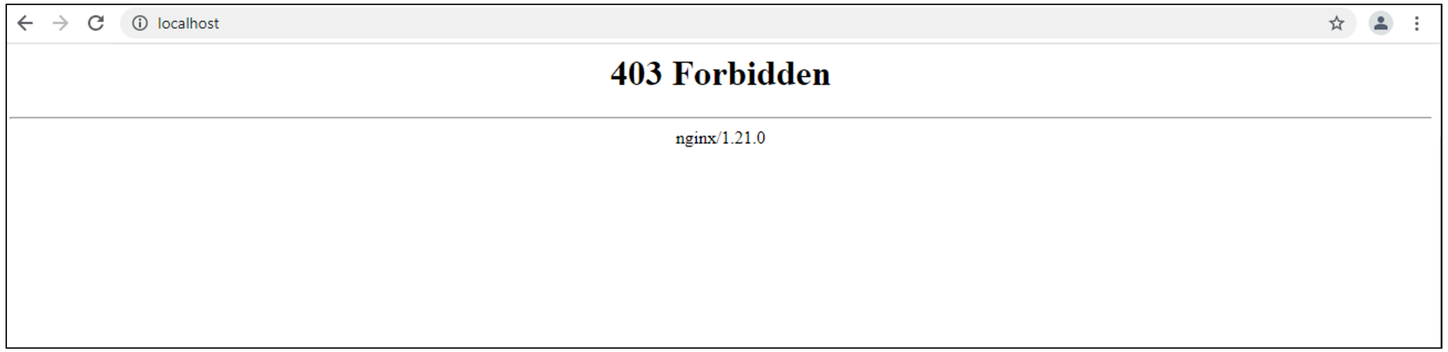
```
$ mkdir web
```

```
$ docker container run --name hello-nginx -d -p 80:80
                -v ${PWD}/web:/usr/share/nginx/html
                nginx
```

Only for Powershell

Host File/Directory

Container File/Directory



CREATE STATIC WEBSITE FOR NGINX

CREATE STATIC WEBSITE FOR NGINX

```
$cd web
$echo “Hello World” > index.html
```

CREATE STATIC WEBSITE FOR NGINX

```
$cd web
$echo “Hello World” > index.html
```

THANK YOU