

**Lab Manual- Container sample: Run a Dotnet web application**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLab992

**Contributor:**

Contents

[1. Objective 2](#_Toc155376141)

[2. Port Mapping in Docker 3](#_Toc155376142)

[**Key Concepts in Port Mapping:** 3](#_Toc155376143)

[3. Pull .NET 8, ASP.NET Core Composite images 4](#_Toc155376144)

[4. Run Container and expose it on port 8000 4](#_Toc155376145)

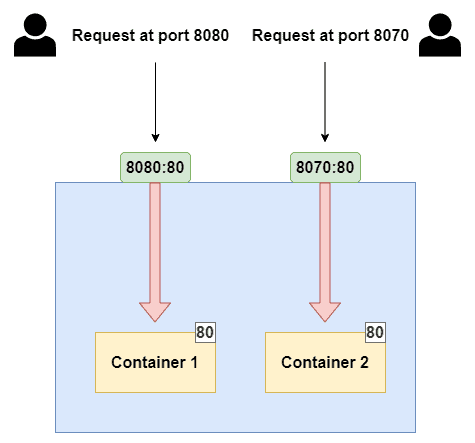
[5. Launch web app in Browser 4](#_Toc155376146)

# Objective

In this Lab, you'll learn how to use .NET and Docker together and also understand port mapping.

# Port Mapping in Docker

In Docker, a container is an isolated environment that runs applications and services, similar to a physical or virtual machine. Just like these machines, containers have their own set of ports. However, by default, these ports are not directly accessible from outside the container. This is where port mapping, also known as port forwarding, comes into play. Port mapping allows you to expose the ports within a Docker container, making the services running inside the container accessible to the host system or to other containers within the Docker environment.

****

**Key Concepts in Port Mapping:**

**Port:** Ports are numeric identifiers used to differentiate between different network services running on the same host. Ports are categorized into two groups: well-known ports (ranging from 0 to 1023) and dynamic or private ports (ranging from 1024 to 49151). Well-known ports are reserved for commonly used services like HTTP (port 80) and HTTPS (port 443).

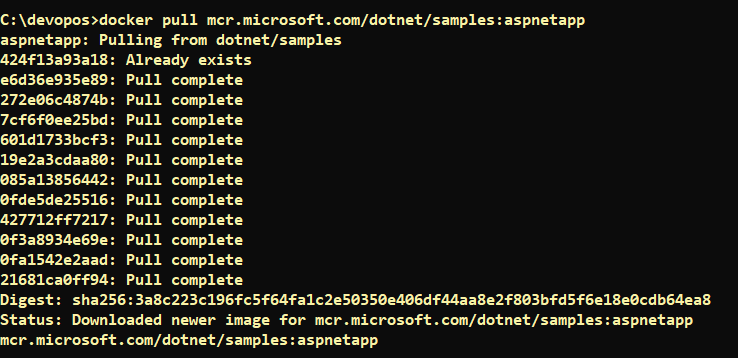
**Host Port vs. Container Port:**

* **Host Port:** This is a port on the host system, which is the machine where Docker is running. The host port is used to access the service or application running inside the Docker container from the host machine or from external systems.
* **Container Port:** This is the port on which the service or application inside the Docker container is listening. It may be a specific port required by the application, for example, a web server listening on port 80.

**Dynamic Mapping:** If you omit the host\_port, Docker will automatically assign an available host port. This is useful when you want to avoid port conflicts on the host system.

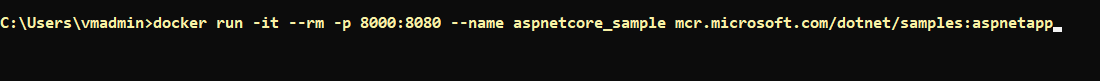
# Pull .NET 8, ASP.NET Core Composite images

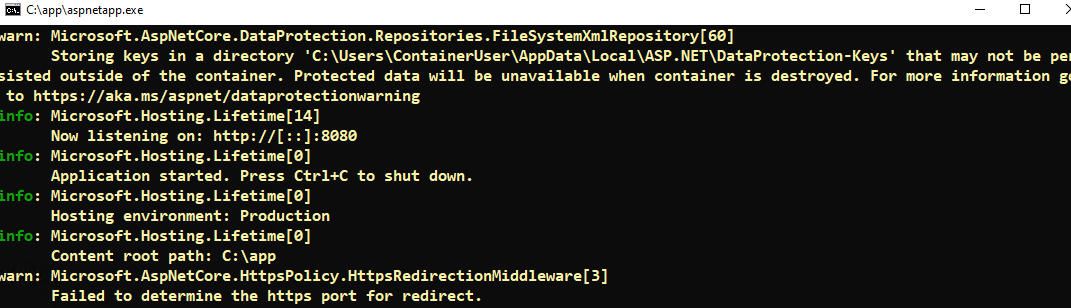
docker pull mcr.microsoft.com/dotnet/samples:aspnetapp



# Run Container and expose it on port 8000

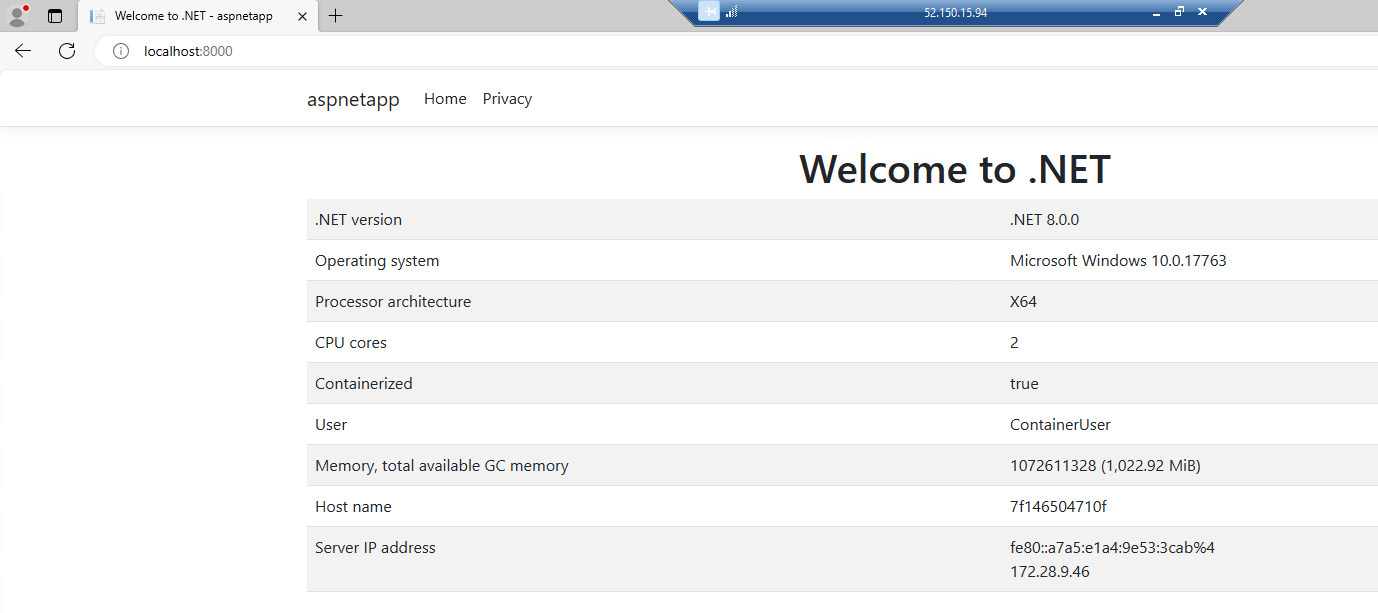
docker run -it --rm -p 8000:8080 --name aspnetcore\_sample mcr.microsoft.com/dotnet/samples:aspnetapp





# Launch web app in Browser

[**http://localhost:8000/**](http://localhost:8000/)

****