Docker Practice

1. Docker Desktop

* It is a Service Daemon running on an Operating System (Windows, MacOS or Linux)
* In Windows it uses a Hyper-V Virtual Machine as it’s runtime platform (can also use WSL2 for Linux container), the Hyper-V VM is in folder C:\ProgramData\DockerDesktop\vm-data

Graphical user interface, text, application, Word

Description automatically generated

1. Docker images and Containers

* Docker containers are instances of Docker images, whether running or stopped. In fact, the major difference between Docker containers and images is that containers have a writable layer.
* When you create a Docker container, you’re adding a writable layer on top of the Docker image. You can run many Docker containers from the same Docker image. You can see a Docker container as an instance of a Docker image.

1. Assume we already completed development of a project, and everything is stored in git, the following Windows PowerShell script

|  |
| --- |
| docker run --name repo alpine/git clone https://github.com/docker/getting-started.git |

Will:

* Find and download a docker image (int this case is alpine/git) from Docker Hub
* Download the source code into app/src folder for later docker build

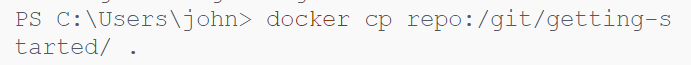
Text

Description automatically generated

Text

Description automatically generated

1. Download source file locally



Open C:\Users\john\getting-started

Graphical user interface, application

Description automatically generated

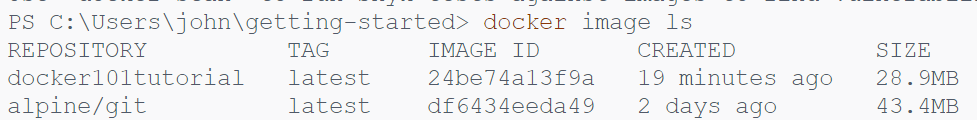
1. Build an Dock Image from a dock file

A picture containing text

Description automatically generated

* Since dockfile is local, we need to run docker build in the local folder
* But the application image was built on the container

1. Show images



* Alpine/git is the Linux based OS that the container will be run on
* Docker101tutorial is the application image that will be run on the container

1. Run the application on a new container called “docker-tutorial”



* -p 8080:80 means use port 8080 on the host machine to map port 80 on the container
* --name specified the name of the new container
* “docker101tutorial” is the image name that is going to run on the new container

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Start the container



1. Stop the container



1. Remove the container



1. Docker for .NET Core

Table

Description automatically generated

* Better to include \*.sln and \*.csproj into the same folder so that Dockerfile and .dockerignore file will be in the same folder
* In Solution Explorer window, right click Project 🡪 Add 🡪 Docker Support

A screenshot of a computer

Description automatically generated with medium confidence

* Pick OS, either Linux or Windows and this will generate the Dockerfile
* Build

|  |
| --- |
| docker build -t webappdockertest -f Dockerfile . |

* Check Images

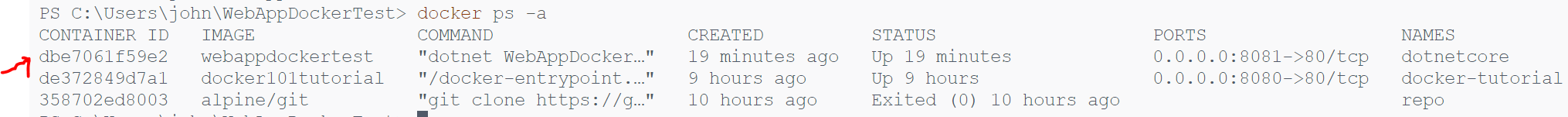
Text

Description automatically generated

* Create container and run the application using docker run:



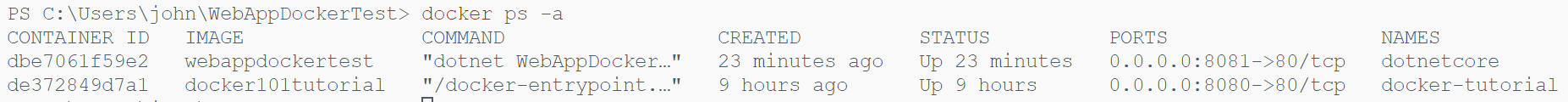
* Check containers



* Remove an idle container



* Check whether the container has been removed



* And the applications still working

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated