# Overview

For containers to be useful, they will need to access storage to retrieve and save data and connect to networks to communicate with other devices. In this guided practice, you will explore some of the storage and networking capabilities available to containers.

# Objectives

* Manage container storage and networking.

## Skills Reviewed

* Creating and running containers.
* Creating folders.
* Configuring file system security.

## New Skills

* Modifying the free space on a container.
* Creating container bind mounts.
* Creating docker volumes.
* Attaching docker volumes to containers.

## References

* Networking - <https://docs.microsoft.com/en-us/virtualization/windowscontainers/container-networking/architecture>
* Storage - <https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/container-storage>

# Initial Conditions

Your virtual machine should be in this state prior to beginning this guided practice:

* Docker Enterprise is installed on a Windows Server host.
* The **iis-core-template** image has been created.

# Final Conditions

At the end of this exercise, you will have:

* A web server with a web site created using a container named **Volume-Test** that uses a persistent volume and the **iis-core-template** image.
* **WebSite** docker volume created.

# Instructions

**Remove** any **containers** that still exist, before proceeding.

## Exploring docker Storage

By default, every container is given a small amount of storage to use when running an application. The storage in the container is ephemeral, meaning that it will disappear when the container is deleted.

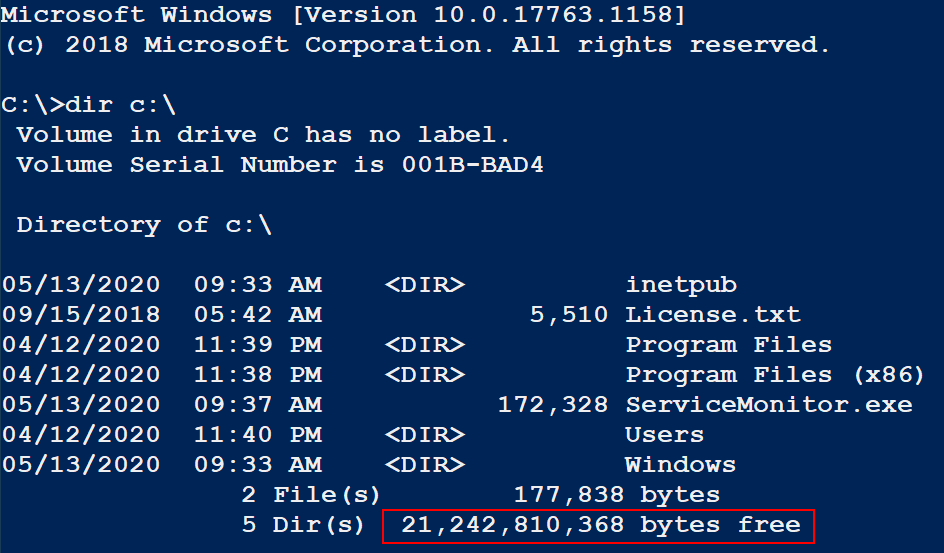
You can see the available storage using the following procedure:

1. Login to your **Host** machine.
2. Open an **elevated** **PowerShell** session.
3. Type the following **command** to create and run a container:

docker run --rm -it iis-core-template cmd

1. This will create a container using the **iis-core-template** base image and start a command prompt. The **--rm** option will remove the container when it stops. By not including the **--name** option Docker will create a name for the container.
2. To view the available free space in the container, type the following command:

dir c:\

1. You should see the output shown below. Note the available free space, the default for a Windows container is 20GB.
2. Type **exit** to exit and remove the container.
3. At times, the default free space will not be enough. You can specify the amount of space by using the **--storage-opt** parameter.
4. To increase the free space on the container to 50GB, **type** the following **command**:

docker run --rm -it --storage-opt “size=50GB” iis-core-template cmd

1. Verify that the **free** **space** has changed and then **type** **exit** to exit and remove the container.

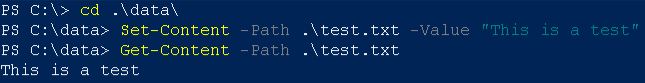
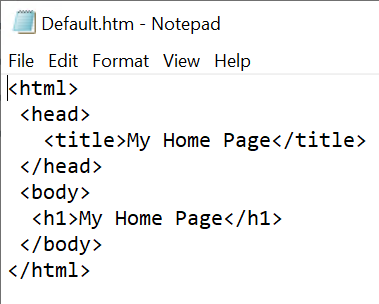
## Creating a Bind Mount

A Bind mount allows a container to share a folder with the host. If your container needs data to persist when a container is deleted and restarted this is one method that can be used. When a bind mount is created, a folder that exists on the host is mapped to a folder on the container. The folder on the container does not need to exist prior to running the container.

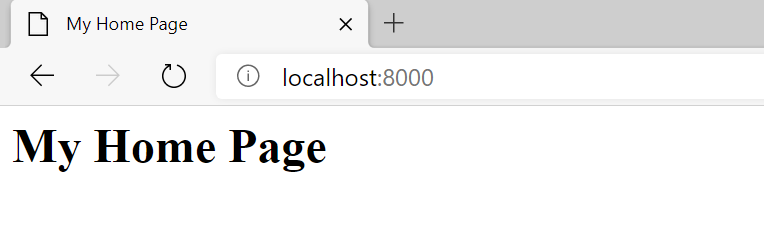
To create a bind mount on a container, perform the following:

1. Login to your **Host** machine.
2. Create a folder named **WebData** in the root of the **D**:\ drive.
3. Give the **Authenticated Users** group **modify** permissions on the **D:\WebData** folder.
4. Type the following **command** to create a bind mount on a container:

docker run -it --rm -v D:\WebData:C:\data:RW --name Storage-Test iis-core-template PowerShell

1. This command will launch a container and map the **D:\WebData** folder into the container in the **C:\Data** folder.
2. To test the storage, change to the **C:\Data** folder and create a file with a message as shown in the figure below.
3. Exit the container and verify that the **test.txt** file exists in the **D:\WebData** folder on the ***lastname*-VM-Host** machine.
4. Delete the **test.txt** file.
5. Create a file named **default.htm** in the **D:\WebData** folder on the host machine.
6. Open the file in **notepad** and add the **html code** shown in the figure. Save the file when you are done.
7. To use the **D:\WebData** folder to host the files for the website on your container, type the following:

docker run -it -p 8000:80 -v D:\WebData:C:\inetpub\wwwroot:RW --name Storage-Test iis-core-template PowerShell

1. Open the **browser** on your host machine and browse to <http://localhost:8000>. You should see the screen shown below.
2. Exit the **PowerShell** session to stop the container.

## Creating a Named Volume

Another way to add persistent storage to a container is by using named volumes. Named volumes allow you to create and name a volume and then use the volume by referring to the name instead of having to use a file system path.

To create a named volume, perform the following:

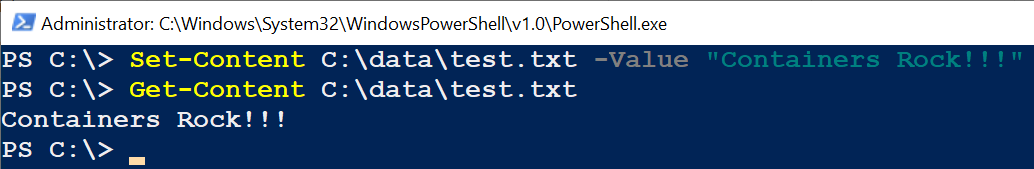
1. Open an **elevated** **PowerShell** Session on your **Host** system.
2. Type the following command to create a volume named **WebSite**:

docker volume create WebSite

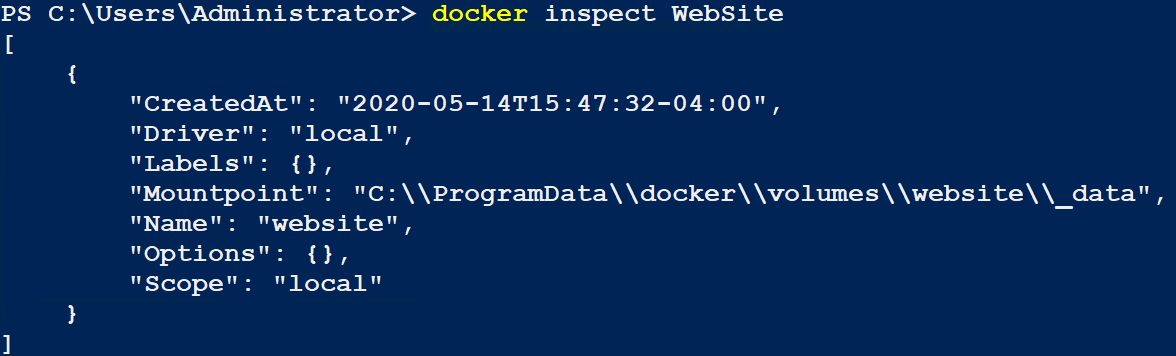
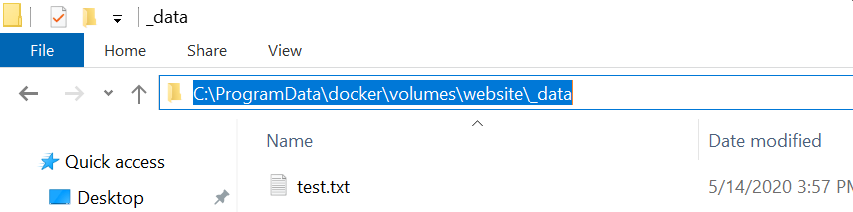
1. To use this volume with a virtual machine, **type** the following **command**:

docker run -it --rm -v WebSite:c:\data --name Volume-Test iis-core-template PowerShell

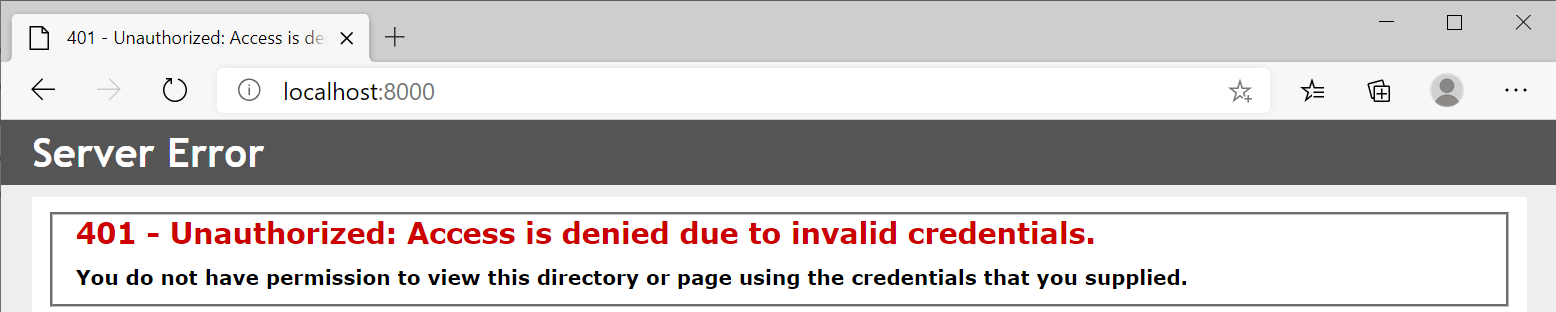
This will start a container named **Volume-Test** with the **WebSite** volume mounted in the **C:\Data** folder and open a PowerShell session on the container.

1. In the container**, create** a file named **test.txt** that containssometext in the **C:\Data** folder. You can use the steps shown in the figure below.
2. Exit and remove the container by typing the **exit** command.
3. To verify that the file still exists, you need to locate where the volume is stored on your system. Type the following command to identify the storage location for the **WebSite** volume:

docker inspect WebSite

1. You should see the output shown below. The location is specified in the Mountpoint.
2. Open File Explorer and browse to the specified location. You should see the **test.txt** file in the location as shown below.
3. Keep the folder above open and **open** another **File** **Explorer** window and **browse** to the **D:\WebData** folder.
4. Copy the **default.htm** file into the same location as the **test.txt** file.
5. Run the following **command** to map the **WebSite** volume to the **c:\inetpub\wwwroot** folder:

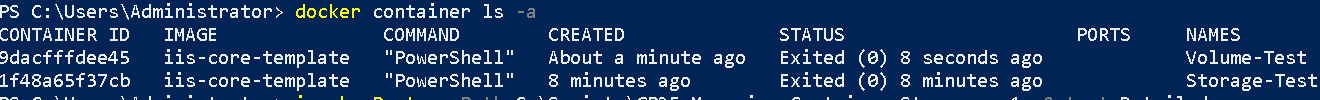
docker run -it -p 8000:80 -v WebSite:C:\inetpub\wwwroot --name Volume-Test iis-core-template PowerShell

1. This is the same command you ran previously with a few changes. Instead of specifying a folder you specified a volume. IN addition, **:RW** was removed after **c:\inetpub\wwwroot** as these are the default permissions in the container for the volume.
2. You should be able to browse to <http://localhost:8000> and see your web site as shown below.
3. Oh no!!! I’ll leave this error for you to figure out.
4. When you have fixed the issue, exit the container to exit PowerShell and stop the container.

# Submission Requirements

1. Verify that you have two containers: **Volume-Test** & **Storage-Test**. If both containers do not exit, the grading script will fail with ALL red. **Volume-Test must** be running.

docker container ls -a



1. **Download** the **grading** **script** from the assignment page to the **C:\Scripts** folder.
2. Check your lab by running the following command:

Invoke-Pester -Path C:\Scripts\GP25-Managing\_Container\_ Storage.Test.ps1

**Note**: You will see a security warning when running the script. Enter **R** to run the script.

If you want to see more detail, add **-Output Detailed** to the command. This may assist you with troubleshooting

Invoke-Pester -Path C:\Scripts\GP25-Managing\_Container\_ Storage.Test.ps1 -Output Detailed

1. You should not see any red in the output. Red in the PowerShell way of telling you that an error condition exists. Most of the time, the output will tell you what is wrong. If it is not obvious, contact your teacher and ask for assistance. You will be learning PowerShell during this term. **Correct** any **errors** you may have and run the script until all the output has no red. You should see the output like the images below.

Text, chat or text message

Description automatically generated

1. Capture a snippet that shows the PowerShell Command and all its output. If you must use more than one snippet to capture the output, you must have at least **one line of overlap** in the snippets. The text in the snippets **must be legible** when pasted into the Word document. Paste the snippet(s) into a **new** **Word** **document**
2. **Fill** **in** the **information** in the following table. Copy the following table into the **Word** **document** and fill in the information about all the **new** commands used in this lab (the example provided is not a new command and should be deleted):

|  |  |  |
| --- | --- | --- |
| Docker Storage Commands | | |
| Command | Example | Description |
| *docker run* | *docker run -it --rm iis-core-template PowerShell* | *Creates a container using the iis-core-template ….* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Upload** the **document** in the submission area of the assignment.