# Overview

Containers can be very useful when you understand how they work. In this guided practice, you will explore the basic process that is used when developing applications using containers

# Objectives

* Deploy and manage windows containers
  + Create containers.
  + Manage containers with docker.

## Skills Reviewed

* Using the docker run command.

## New Skills

* Use docker command to search and pull an image in a registry.
* Create, view, modify, save, and delete docker images.
* Create, view, manage, and delete docker containers.

# Initial Conditions

* Docker Enterprise installed on a Windows Server virtual machine.

# Final Conditions

* Container base images for Windows **Nanoserver** and **Windows Server Core** pulled.
* Container base image **iis-core-template** created with IIS installed.
* Working container created from the **iis-core-template image.**

# Instructions

The first part of a typical container workflow is to locate and download (pull) a container base image to base your container on. In this next section, you will learn how to locate, download, and view the container base images on your machine.

## Using Container Base Images

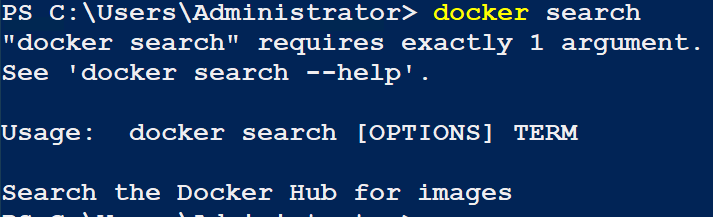
Container images are in registries on the Internet. The default registry for Docker is Docker hub <http://hub.docker.com>. You can create an account on the registry to search for images or push (upload) your own images. You can also create a private registry for your organization for internal use.

### Finding a Container Base Image

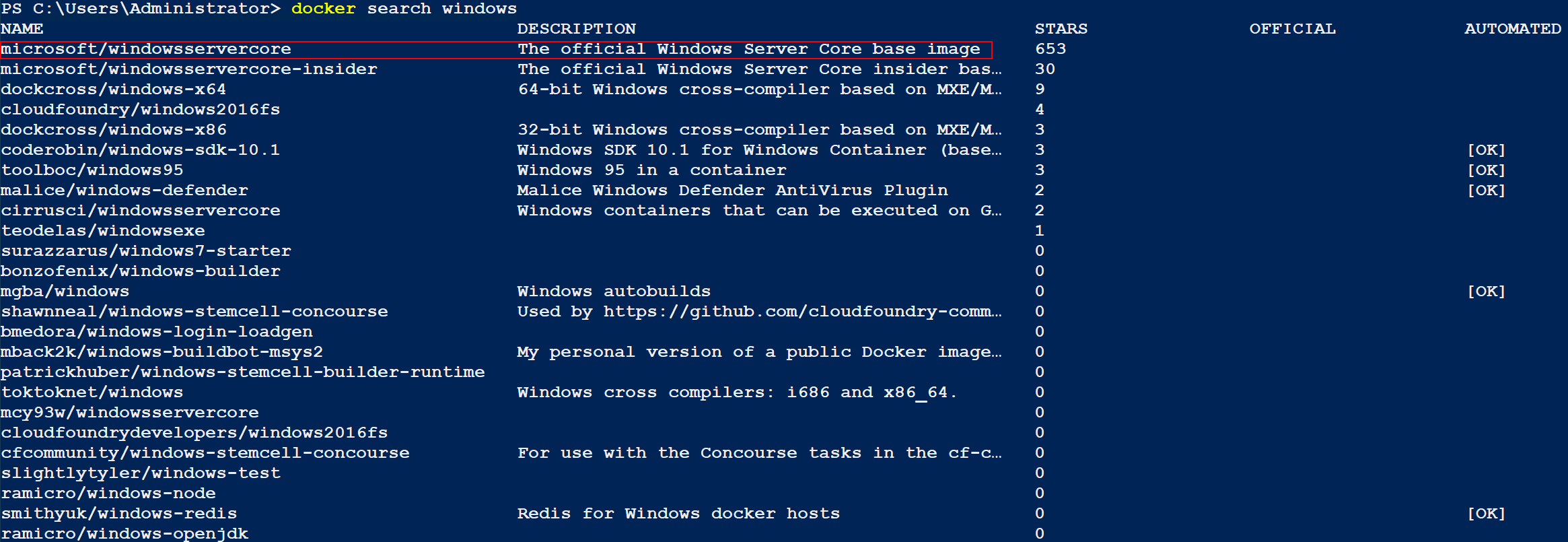
The first step in creating a container is to locate a container base image. To locate a container base image, perform the following:

1. Login to your host machine
2. Open an **elevated** **PowerShell** session.
3. Type the following command to search for a base image:

Docker search

1. You should see the following.
2. The output tells you that you can find more help by using the –help parameter and that you need to supply a search TERM and additional options are optional.
3. To find a container image using windows as the search term, type the following

docker search windows

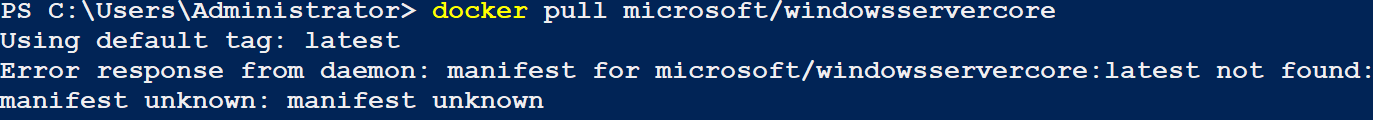
1. You should see the following output.
2. Note the official designation in the description, you will want to stick to these images for use in production.
3. To pull (download) an image you need to know the name of the image. The name is in the first column.

### Pulling a Container Base Image

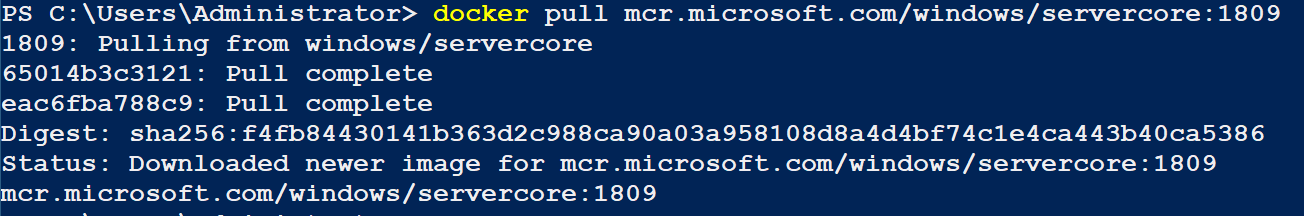
Now that you have located an image, the next step is to pull (download) the image. To pull an image, perform the following:

1. To pull the official Window Server Core base image from the Docker registry, type the following command in PowerShell:

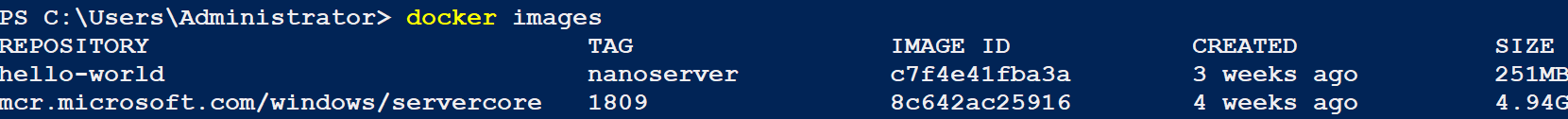
docker pull microsoft/windowsservercore

1. You should see the error shown below. This error is shown to you because several tutorials on the web show this command and you should know how to resolve this issue. This error has to do with the fact that Microsoft has changed the way that they tag images. You can find more information here 🡪 [https://techcommunity.microsoft.com/t5/containers/windows-server-2019-now-available/ba-p/382430#](https://techcommunity.microsoft.com/t5/containers/windows-server-2019-now-available/ba-p/382430)
2. Type the following command to pull the latest Windows Server 2019 Core image:

docker pull mcr.microsoft.com/windows/servercore:1809

1. You should see the output below.
2. You can view the container base images on your system by typing the following command:

docker images

1. You should see the output below.
2. Pull the **Windows nanoserver** **image**. Its name is **mcr.microsoft.com/windows/nanoserver:1809**
   1. **Note:**  The latest version of the **hello-world** container might be built from the **Nanoserver** image. You may get a message that it is already installed. If this is the case, just proceed to the next step.

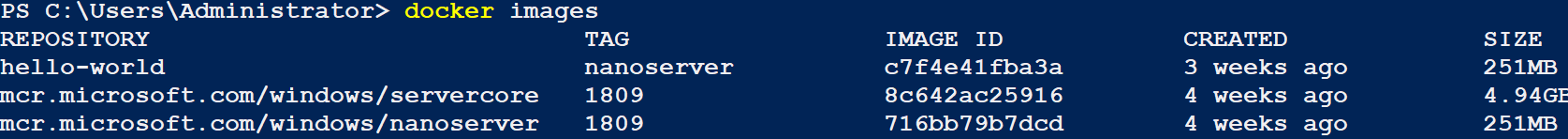
## Removing a Container Base Image

Images are often updated with patches, hotfixes, and newer features. When this is done you will need to rebuild your containers using the latest images and then delete the old images.

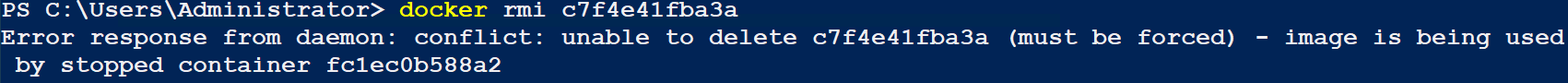
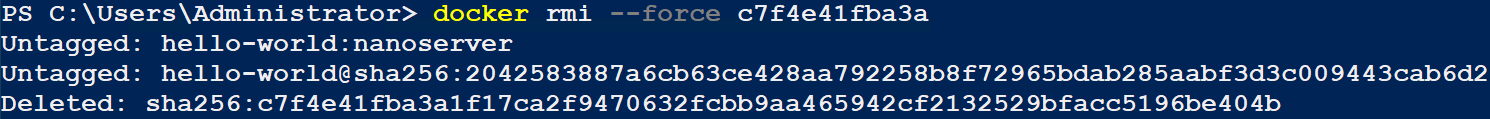
To delete a docker image, perform the following:

1. In PowerShell, type the following command to view the images:

docker images

1. You should see the output shown below.
2. To remove the hello-world image using the Image ID, type the following (your images will be different):

docker rmi c7f4e41fba3a

1. You may see the output below. If so, rerun the command with the **--force** parameter
2. You should see output like the screen below.

**Note**: You can download, create, and run containers at the same time using the docker run command as it will check to see if the image is present and download it if it is not. It is more likely when planning a container infrastructure that you will gather all your images before starting.

## Creating and Using Containers

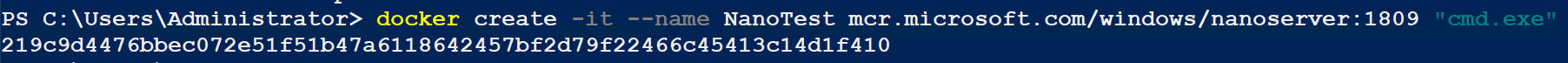
Once you have container base images you can start creating containers. There are several ways that you can create a new container, the first time through the process you should do every step individually to better understand the process and commands used.

### Creating a container

To **create** a **container** using the **nanoserver** image, perform the following:

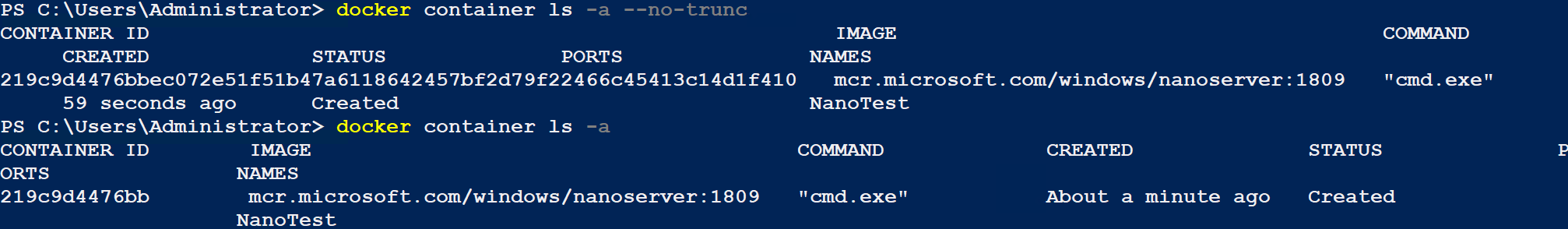
1. **Open** an **elevated** **PowerShell** session and **type** the following: **Note**: If your image uses a different tag (e.g latest) then that should be used after the colon instead of 1809.

docker create -it --name NanoTest mcr.microsoft.com/windows/nanoserver:1809 “cmd.exe”

1. You should see the output like the screenshot shown below.
2. The command used above created a container named **NanoTest** (--**name** **NanoTest**) with an **interactive** **terminal** (--**interactive** or **-i** and --**tty** or **-t**) using the image **mcr.microsoft.com/windows/nanoserver** that runs the program **cmd.exe**.
3. You can view the container you just created by typing the following command:

docker container ls -a --no-trunc

docker container ls -a

1. You should see output like the image below.
2. The two commands show you information about the container, one truncated and the other not. Most often you will not user the **--no-trunc** parameter, but you should know it just in case some of the container information that is removed is needed.
3. Note the name of the container. In docker if you do not specify a name when creating a container, the docker command will create one for you. You should name your containers whenever possible.

### Starting a Container

When you list a container, you will see a value in the COMMAND column. This value is a command that is run when the container is started. The container is stopped when the command exits, or you manually stop the container.

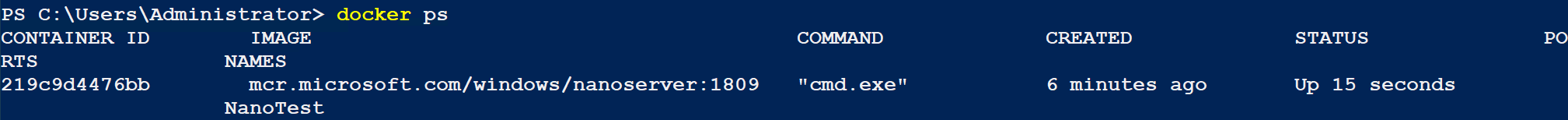
To start a container, perform the following:

1. In **PowerShell**, **type** the following command to start the container using its name:

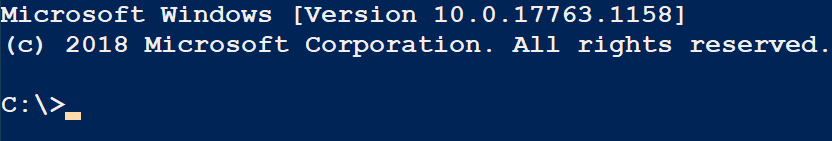
docker start NanoTest

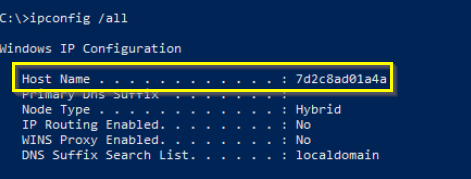
1. You should see the output below. This just shows that the container was started.
2. To see the running containers, type the following:

docker ps

1. You should output like the figure below.
2. To attach to the container and interact with it, type the following command:

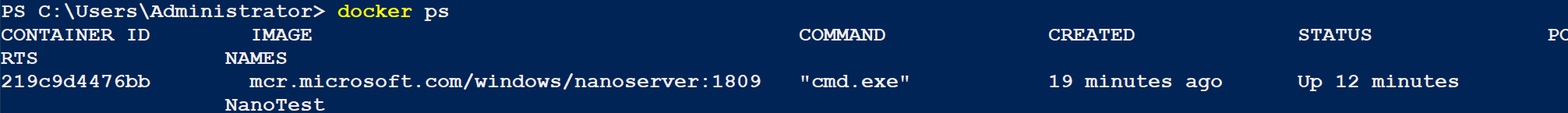
docker attach NanoTest

1. You should be placed into the cmd.exe (command prompt) in the nanoserver image as shown below.
2. **Issue** the **ipconfig /all** command. Note that **Host Name**. It is not the hostname of your **lastname-VM-Host** server

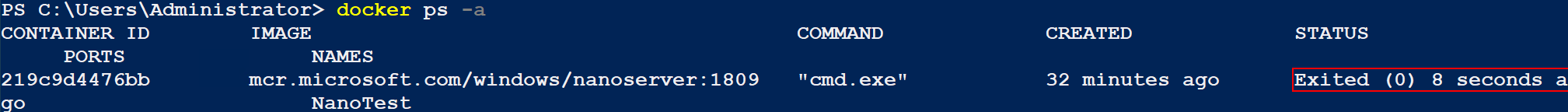
****

1. Typing exit in the command prompt would end the cmd.exe session and stop the container. To **exit** the **container** and **keep** it **running** **type** **ctrl-p** and then **ctrl-q.**
2. To verify that the container is still running, type the following:

docker ps

1. You should see output like the screenshot below:
2. To **stop** the **container**, **attach** to the **container** as you did in a previous step and **type** **exit** in the **command prompt**. This will terminate the command prompt and stop the container.
3. To verify the container is stopped. Type the following:

docker

1. You should see output like the image below.
2. Notice the container exited 8 seconds ago. The container runs until its application stops.

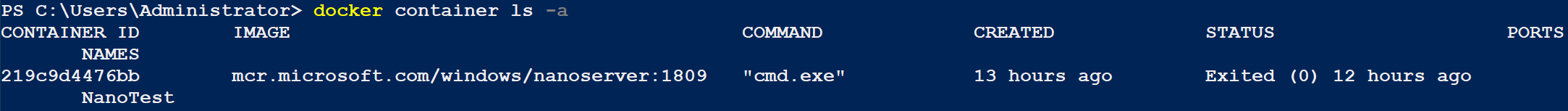
## Removing a Container

The next step would normally be to modify the container and create an image from it. In the latest nanoserver images, this is done offline and requires learning and using specialized tools that you have not learned yet.

To remove the container you just created, perform the following:

1. **Open** an **elevated** **PowerShell** session.
2. **Type** the following **command** to list the containers:

docker container ls -a

1. You should see output like the screenshot below. You will need the name or the ID to remove the container.
2. Type the following command to remove the container:

docker container rm NanoTest

1. You should see output like the screenshot below.
2. Like many of the docker commands, the only output is the name of the container or image. To verify the container was deleted you can run the docker ls -a command again and see that the image no longer exists.

## Starting and running a container

Now that you are familiar with the process of creating, starting, and deleting containers, you will do this all at once using the docker run command.

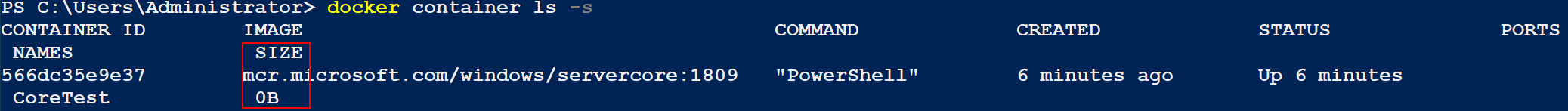
To create and run a docker container using the Window Server core image, perform the following:

1. **Open** an **elevated** **PowerShell** session.
2. **Type** the following **command**:

docker run -it --hostname svr-core-test --name CoreTest mcr.microsoft.com/windows/servercore:1809 “PowerShell”

1. You should be shown a PowerShell prompt. You can test it out by running a few commands as shown in the screenshot below.
2. **Exit** the **container** by typing **ctrl-p ctrl-q.** (Note the change in the Powershell prompt). Issue the **hostname** command so show that you are back on the VM-Host.
3. Type the following command to determine the size of the container:

docker container ls -s

1. You should see output similar to the screenshot below.
2. Notice that the size of the container is 0 bytes. This is because we have not added anything to the container. The container uses the storage like a Hyper-V differencing disk with the image being the parent disk and the container being the child disk.

## Modifying a Container

Now that you have a container, you can modify it to run your service or application.

To modify the container to be a web server running IIS, perform the following:

1. **Open** an **elevated** **PowerShell** session.
2. **Attach** **to** the **CoreTest** **container** so that you have a PowerShell session inside the container.
3. **Verify** that you have attached and are in the container by issuing the **whoami** command**.**



1. In the **container** **PowerShell** session, use PowerShell to **add** the **web** **server** **role** (Web-Server).

Install-WindowsFeature -Name Web-Server

1. **Type** the following **command** to **download** the **ServiceMonitor.exe** program that will be used to start the web server service:

Invoke-WebRequest -UseBasicParsing -Uri "https://dotnetbinaries.blob.core.windows.net/servicemonitor/2.0.1.6/ServiceMonitor.exe" -OutFile "C:\ServiceMonitor.exe"

1. **List** the **c:\** directory to **verify** that the **ServiceMonitor**.**exe** program **exists**.
2. Your container now has all the code it needs to run IIS.
3. **Exit** the **container** by typing **ctrl-p ctrl-q.**

## Creating an Image from a Container

Now that the container has been created and modified to suit your needs, you can convert it to an image using the docker commit command.

To commit the container changes to an image, perform the following:

1. **Open** an **elevated** **PowerShell** session.
2. **Stop** the **container** by typing the following command:

docker container stop CoreTest

1. Commit the changes to an image using the following command (change the author to your name):

docker commit -a "Kevin Azevedo" -m "IIS Installed" CoreTest iis-core-template

1. The command should output the image id for the new image that was created. You should now have a new image named iis-core-template. You can verify this by typing the following command:

docker images

1. You should see output like the screenshot below.
2. To test your image in a new container, type the following:

docker run -d --name IIS-Test -p 8000:80 iis-core-template ServiceMonitor.exe w3svc

1. Once again, you should see the ID of the created container. The two options specified -d runs the container detached (i.e. not interactive) and -p 8000:80 will expose port 80 inside the container to port 8000 on the host. You can test this by **opening** a **web browser** and **browsing** to **http://localhost:8000** you should see the output below which is the default web page for IIS.

# Submission Requirements

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\GP24-Using\_Containers\_in \_Windows.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\GP24-Using\_Containers\_in \_Windows.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

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 Management Commands | | |
| Command | Example | Description |
| *docker pull* | *Docker pull mcr.microsoft.com/windows/servercore* | *Pull (download) an image from a registry* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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