# Overview

Containers are a recent feature of Windows that allow you to run a minimal version of the operating system. This allows you to run web applications that do not require much of the operating system software that normally accompanies Windows. Thus, it provides a system with a small attack surface that can be easily scaled for greater performance and reliability. In this guided practice, you will install Docker for Windows.

# Objectives

* Create containers.
* Manage containers with docker.

## Skills Reviewed

* Installing applications.
* Using PowerShell.

## New Skills

* Install Docker for Windows.
* Installing Modules and Packages in PowerShell.
* Viewing the Docker version.
* Pulling a container image.
* Running a container image.

## References

Install Docker Desktop on Windows <https://docs.docker.com/docker-for-windows/install/>

# Initial Conditions

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

* Virtual machine created with Windows Server 2019 and Hyper-V installed and configured for nested virtualization.

# Final Conditions

At the end of this exercise, you will have:

* Docker enterprise installed on the **<*lastname*>-VM-Host**.
* **Hello-world** docker image downloaded locally.

# Instructions

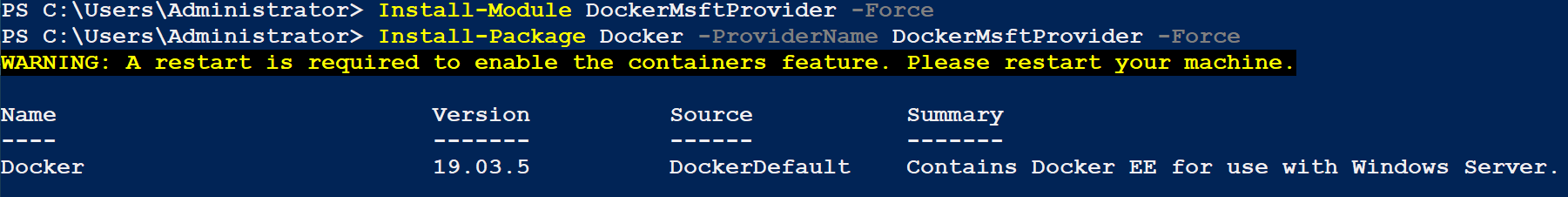
## Installing Docker for Windows

To install Docker for Windows, perform the following:

1. Login to your **Host** virtual machine
2. Openan **elevated PowerShell** session and type the following commands:

Install-Module DockerMsftProvider -Force

Install-Package Docker -ProviderName DockerMsftProvider -Force

1. You should see the output shown below.
2. If prompted, **restart** your **virtual machine** to complete the installation.

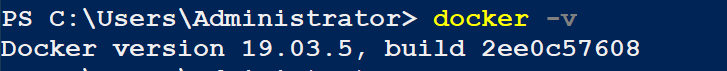
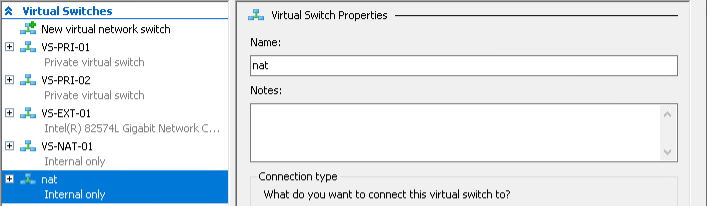
## Verifying Docker Installation

Once the installation is complete, you will want to verify that everything is working correctly.

To verify the installation of Docker, perform the following:

1. **Open** **PowerShell**, and type the following:

docker -v

1. You should see output like the screen below. Do not worry if the version number does not match, Docker is updated often, and this will change.
2. Open **Hyper-V management** console and open the **Virtual Switch Manager**. You should see a new virtual switch named **nat** created for Docker as shown below. If you do not see the **nat** switch, restart your **<*lastname*>-VM-Host**.

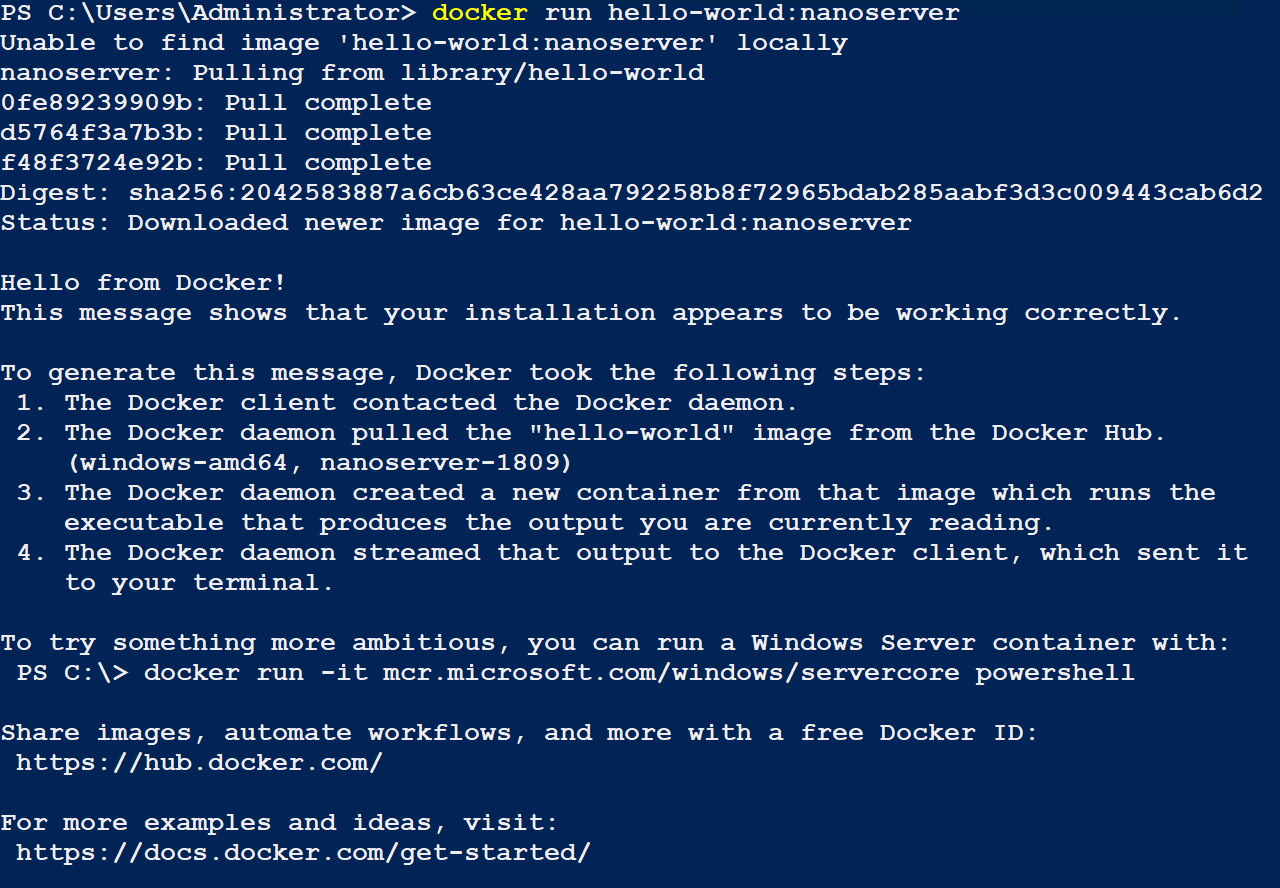
## Testing Docker

Since the whole purpose of Docker is to give you the ability to download and run containers. To test docker you should download and run a container to verify everything is running correctly. Microsoft provides a **hello-world** container based on Windows **Nanoserver** that is used to test your installation.

To test Docker functionality, perform the following:

1. Open a **PowerShell** session and type the following:

Docker run hello-world:nanoserver

1. You should see the output below.
2. **Note**: you could also use the command **docker container run** command. Docker has recently grouped the commands into a hierarchy as the number of commands have grown.

# 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\GP23-Installing\_Docker\_ on\_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\GP23-Installing\_Docker\_ on\_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):

|  |  |  |
| --- | --- | --- |
| **PowerShell/Docker Commands** | | |
| **Command** | Example | Description |
| **Get-Childitem** | Get-Childitem -Path C:\ | Displays the files in the C:\ directory |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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