Hands-On Lab: NAV on Docker

Who should complete this HOL?

This Hands-On Lab is designed to help you understand what Docker is and what NAV on Docker can do for you. After completing the HOL, you should be able to determine if Docker and especially NAV on Docker is useful in your organization. The HOL will use the Workshop VMs as a foundation for the HOL to have a uniform platform for all.

When you have completed this HOL, you can find more info on the nav-docker project on github: http://www.github.com/microsoft/nav-docker. This is also the place you should be filing issues and comments.

What is Docker?

If you are new to Docker and Containers, you might want to scan through this document before heading into the workshop:

https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/

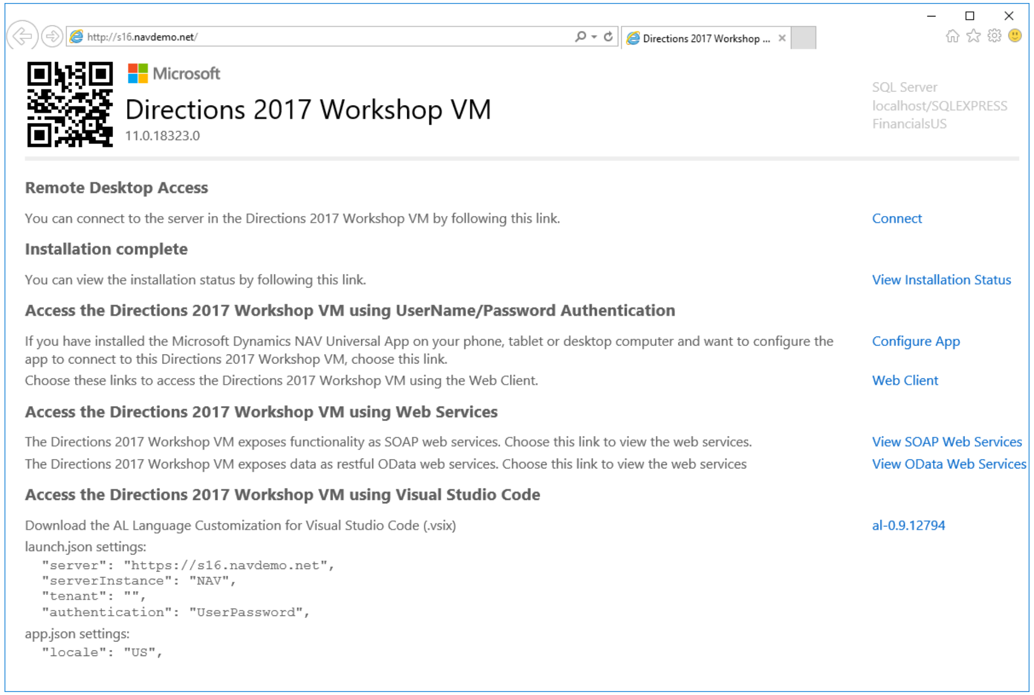
This should give you a better understanding of what Docker is.

For the remaining of the workshop, you will be going through some scenarios, using Docker, on how to Deploy NAV.

When you connect to your learning environment, you are presented with a website, which looks like the image on the right side. What you might not be aware is, that when you are viewing this, you are already using Docker. This website is hosted inside a Container, which is running inside an Azure Virtual Machine, called the Docker host and in this case, your Workshop VM.

Clicking the Connect link will download the .rdp file, which launches Remote Desktop to the Workshop VM.

Note that all other workshops at Directions are using the same Workshop VM, but they just don't care that NAV is running in a Container.



Directions 2017 Workshop VM
11.0.18323.0

SQL Server
localhost/SQLEXPRESS
FinancialsUS

Remote Desktop Access

You can connect to the server in the Directions 2017 Workshop VM by following this link. Connect

Installation complete

You can view the installation status by following this link. View Installation Status

Access the Directions 2017 Workshop VM using UserName/Password Authentication

If you have installed the Microsoft Dynamics NAV Universal App on your phone, tablet or desktop computer and want to configure the app to connect to this Directions 2017 Workshop VM, choose this link. Configure App

Choose these links to access the Directions 2017 Workshop VM using the Web Client. Web Client

Access the Directions 2017 Workshop VM using Web Services

The Directions 2017 Workshop VM exposes functionality as SOAP web services. Choose this link to view the web services. View SOAP Web Services

The Directions 2017 Workshop VM exposes data as restful OData web services. Choose this link to view the web services View OData Web Services

Access the Directions 2017 Workshop VM using Visual Studio Code

Download the AL Language Customization for Visual Studio Code (.vsix) al-0.9.12794

launch.json settings:

```
  "server": "https://s16.navdemo.net",
  "serverInstance": "NAV",
  "tenant": "",
  "authentication": "UserPassword",
```

app.json settings:

```
  "locale": "US",
```

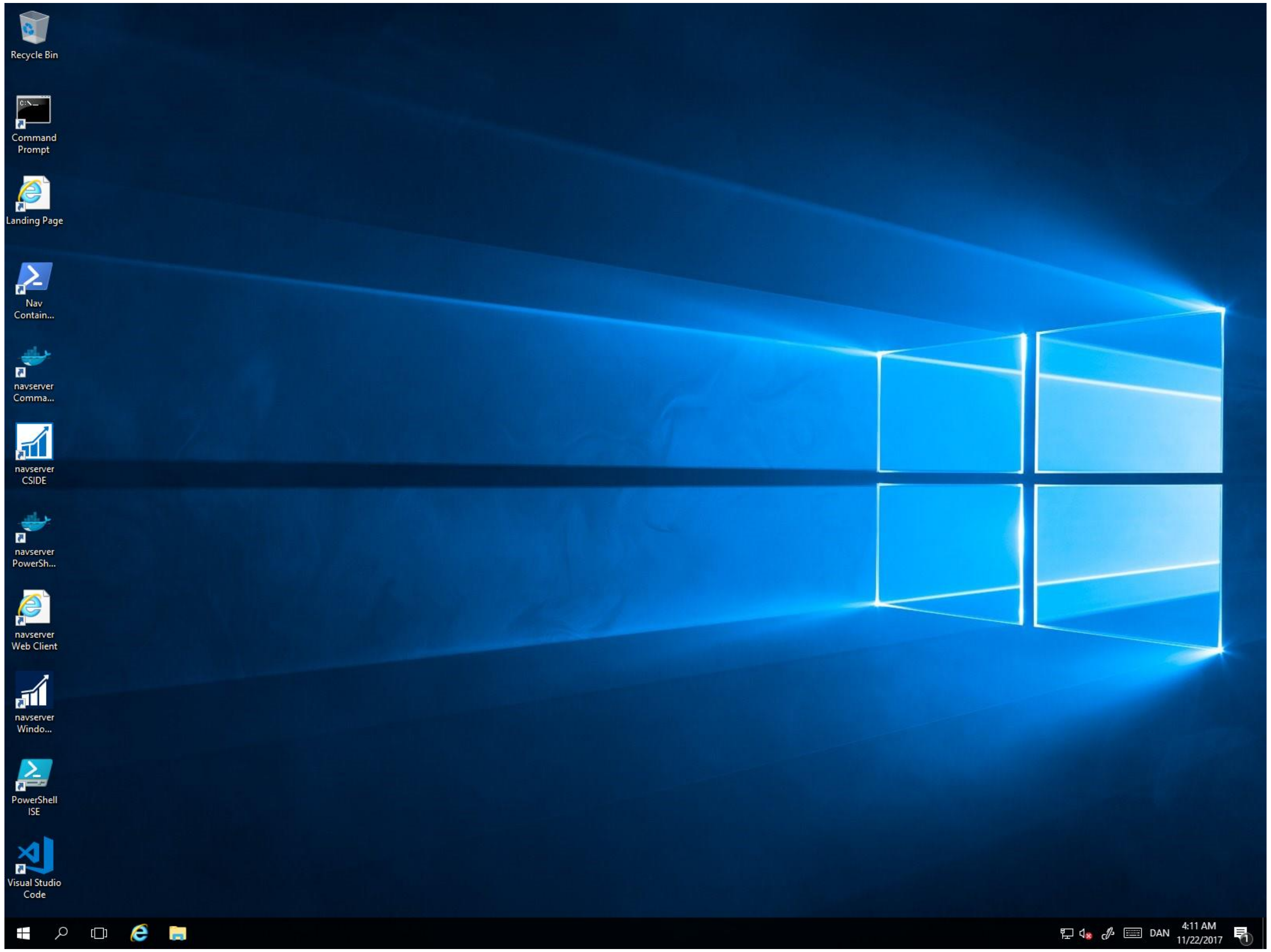
Note also that the Workshop VMs are very much like the NAV Developer Preview VMs with the September Update you can get from http://aka.ms/navdeveloperpreview.

Note also that you will not and cannot connect to a remote desktop in the Container. The Container is based on WindowsServerCore, which has no UI, no desktop.

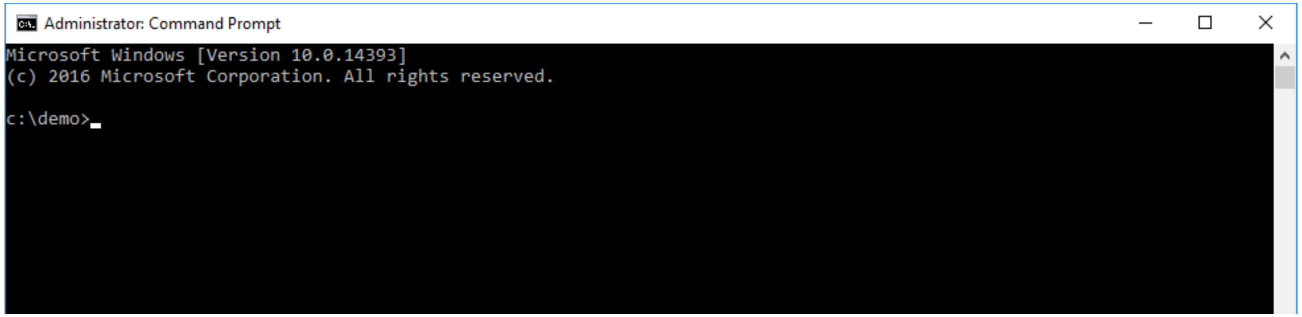
Connecting to the Workshop VM (the Docker host) will allow you to interact with the Docker Containers that are available on that machine by using various commands.

First thing we will do is to have a look at the Workshop VM desktop and what we can do with that.

Open your workshop landing page in a browser and press the Connect button to connect to the remote desktop of your workshop environment. Use the provided credentials to login.
Close Server manager and other windows that pop up and you should have a desktop, with a few shortcuts.



Command Prompt
The Command Prompt is the standard CMD.EXE running as Administrator.
This prompt is primarily there for running Docker commands or other executables.
We will be using the Command Prompt throughout this Hands On Lab.



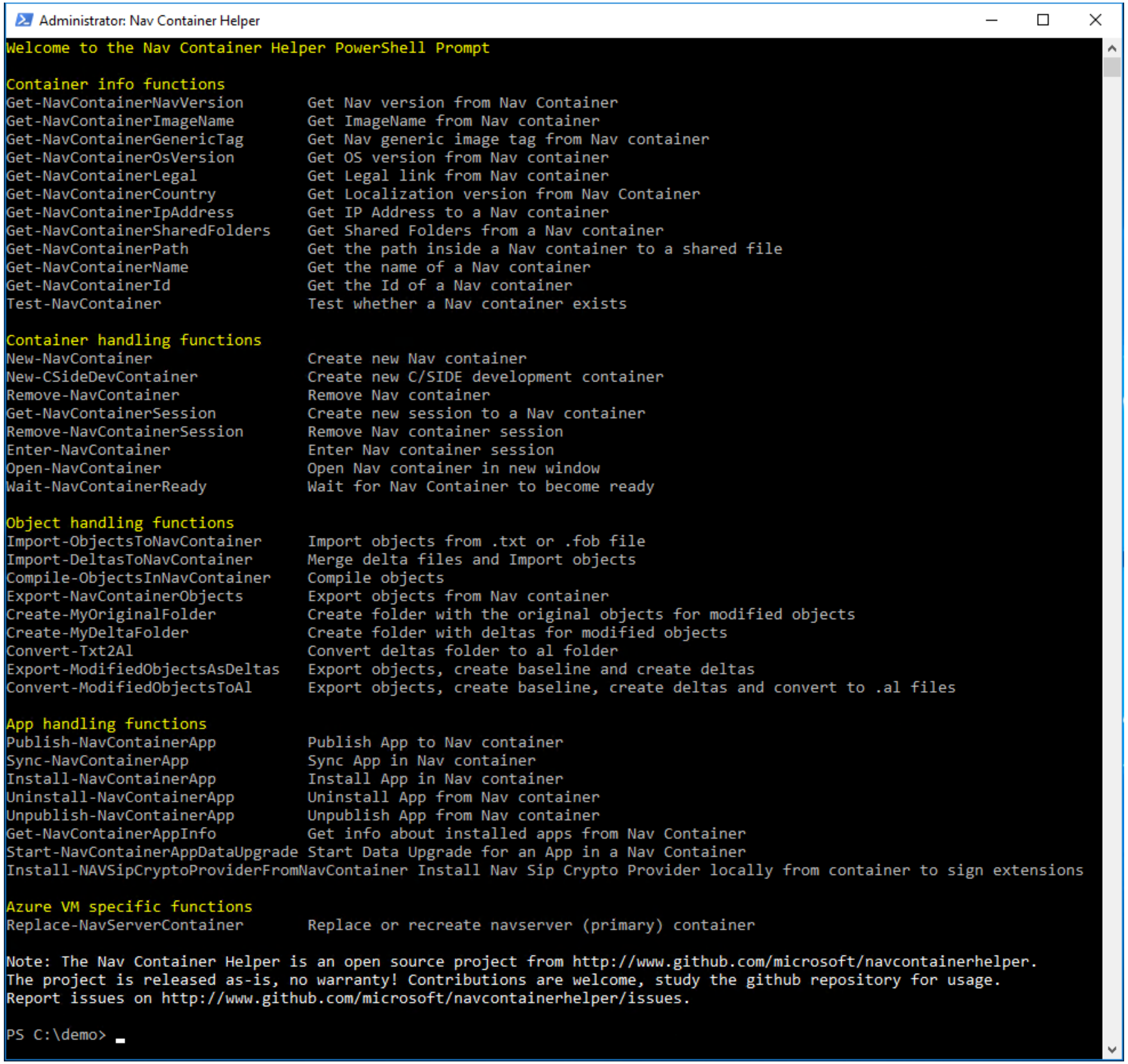
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\demo>_
```

Nav Container Helper
The navcontainerhelper is a set of functions, which will help you working with Nav Containers.

When you start the container helper, it will display a number of the available functions.

Note that the navcontainerhelper is an open source project from http://www.github.com/microsoft/navcontainerhelper and any issues regarding the navcontainerhelper should be added under issues in the github repo.

We will dive into the container helper later.



```
Welcome to the Nav Container Helper PowerShell Prompt

Container info functions
Get-NavContainerNavVersion       Get Nav version from Nav Container
Get-NavContainerImageName        Get ImageName from Nav container
Get-NavContainerGenericTag       Get Nav generic image tag from Nav container
Get-NavContainerOsVersion        Get OS version from Nav container
Get-NavContainerLegal            Get Legal link from Nav container
Get-NavContainerCountry          Get Localization version from Nav Container
Get-NavContainerIpAddress        Get IP Address to a Nav container
Get-NavContainerSharedFolders    Get Shared Folders from a Nav container
Get-NavContainerPath             Get the path inside a Nav container to a shared file
Get-NavContainerName             Get the name of a Nav container
Get-NavContainerId               Get the Id of a Nav container
Test-NavContainer                Test whether a Nav container exists

Container handling functions
New-NavContainer                 Create new Nav container
New-CSideDevContainer            Create new C/SIDE development container
Remove-NavContainer              Remove Nav container
Get-NavContainerSession          Create new session to a Nav container
Remove-NavContainerSession       Remove Nav container session
Enter-NavContainer               Enter Nav container session
Open-NavContainer                Open Nav container in new window
Wait-NavContainerReady           Wait for Nav Container to become ready

Object handling functions
Import-ObjectsToNavContainer     Import objects from .txt or .fob file
Import-DeltasToNavContainer      Merge delta files and Import objects
Compile-ObjectsInNavContainer    Compile objects
Export-NavContainerObjects       Export objects from Nav container
Create-MyOriginalFolder          Create folder with the original objects for modified objects
Create-MyDeltaFolder             Create folder with deltas for modified objects
Convert-Txt2Al                   Convert deltas folder to al folder
Export-ModifiedObjectsAsDeltas   Export objects, create baseline and create deltas
Convert-ModifiedObjectsToAl      Export objects, create baseline, create deltas and convert to .al files

App handling functions
Publish-NavContainerApp          Publish App to Nav container
Sync-NavContainerApp             Sync App in Nav container
Install-NavContainerApp          Install App in Nav container
Uninstall-NavContainerApp        Uninstall App from Nav container
Unpublish-NavContainerApp        Unpublish App from Nav container
Get-NavContainerAppInfo          Get info about installed apps from Nav Container
Start-NavContainerAppDataUpgrade Start Data Upgrade for an App in a Nav Container
Install-NAVSipCryptoProviderFromNavContainer Install Nav Sip Crypto Provider locally from container to sign extensions

Azure VM specific functions
Replace-NavServerContainer       Replace or recreate navserver (primary) container

Note: The Nav Container Helper is an open source project from http://www.github.com/microsoft/navcontainerhelper.
The project is released as-is, no warranty! Contributions are welcome, study the github repository for usage.
Report issues on http://www.github.com/microsoft/navcontainerhelper/issues.

PS C:\demo> _
```

navserver Command Prompt

The navserver Command Prompt is the standard CMD.EXE running inside the navserver container.

When you run dir inside the navserver Command Prompt you will see the Container file system. Folders that are shared from the host to the container are shown as symbolic directory links (SYMLINKD).

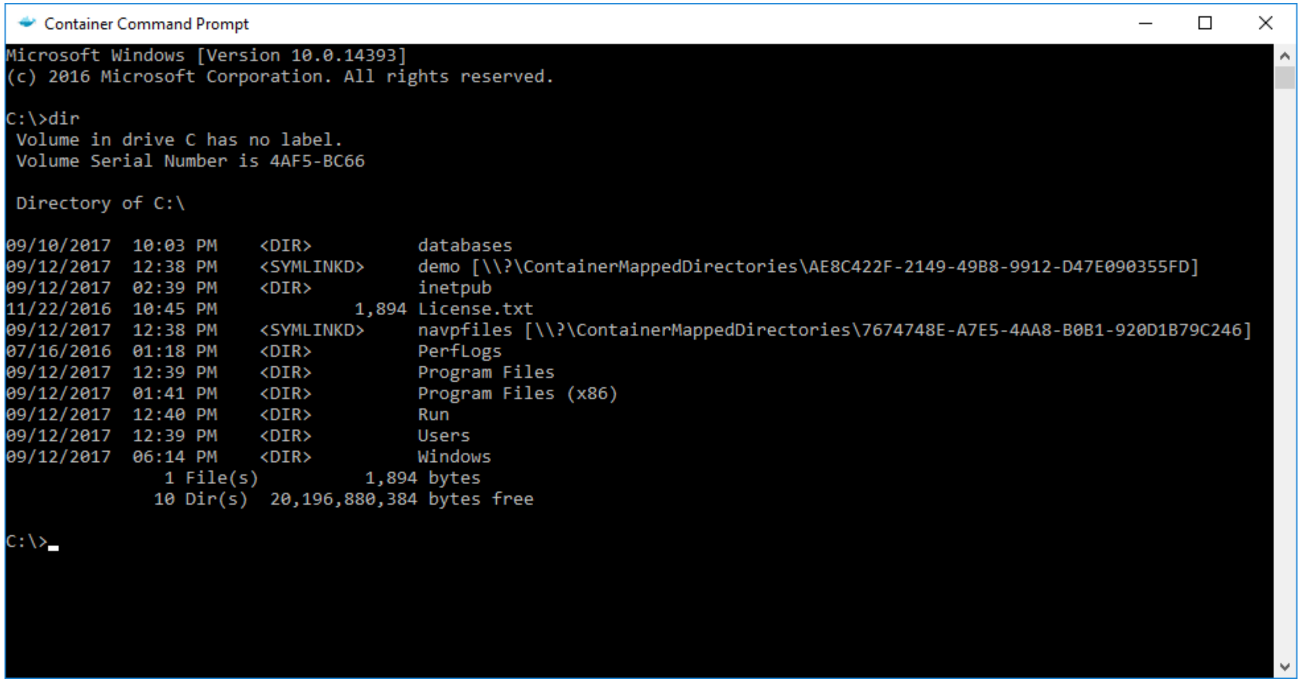
The file system inside the NAV Docker Image consists of a few special folders/files:

c:\run the run folder is the folder containing all the scripts, which are used to set up NAV in the container.

c:\run\my is the location, where you can place scripts which can override functionality of the run folder. Typical scenario is to share a folder from the host to the c:\run\my folder, containing various scripts that you want executed during start.

c:\run\start.ps1 is the entry point for the container.

c:\run\navstart.ps1 is the main script for setting up NAV and launching other setup scripts.



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\>dir
 Volume in drive C has no label.
 Volume Serial Number is 4AF5-BC66

 Directory of C:\

09/10/2017  10:03 PM    <DIR>          databases
09/12/2017  12:38 PM    <SYMLINKD>     demo [\\?\ContainerMappedDirectories\AE8C422F-2149-49B8-9912-D47E090355FD]
09/12/2017  02:39 PM    <DIR>          inetpub
11/22/2016  10:45 PM             1,894 License.txt
09/12/2017  12:38 PM    <SYMLINKD>     navpfiles [\\?\ContainerMappedDirectories\7674748E-A7E5-4AA8-B0B1-920D1B79C246]
07/16/2016  01:18 PM    <DIR>          PerfLogs
09/12/2017  12:39 PM    <DIR>          Program Files
09/12/2017  01:41 PM    <DIR>          Program Files (x86)
09/12/2017  12:40 PM    <DIR>          Run
09/12/2017  12:39 PM    <DIR>          Users
09/12/2017  06:14 PM    <DIR>          Windows
               1 File(s)          1,894 bytes
              10 Dir(s)  20,196,880,384 bytes free

C:\>_
```

navserver PowerShell Prompt

The navserver PowerShell Prompt is a PowerShell prompt running inside the container.

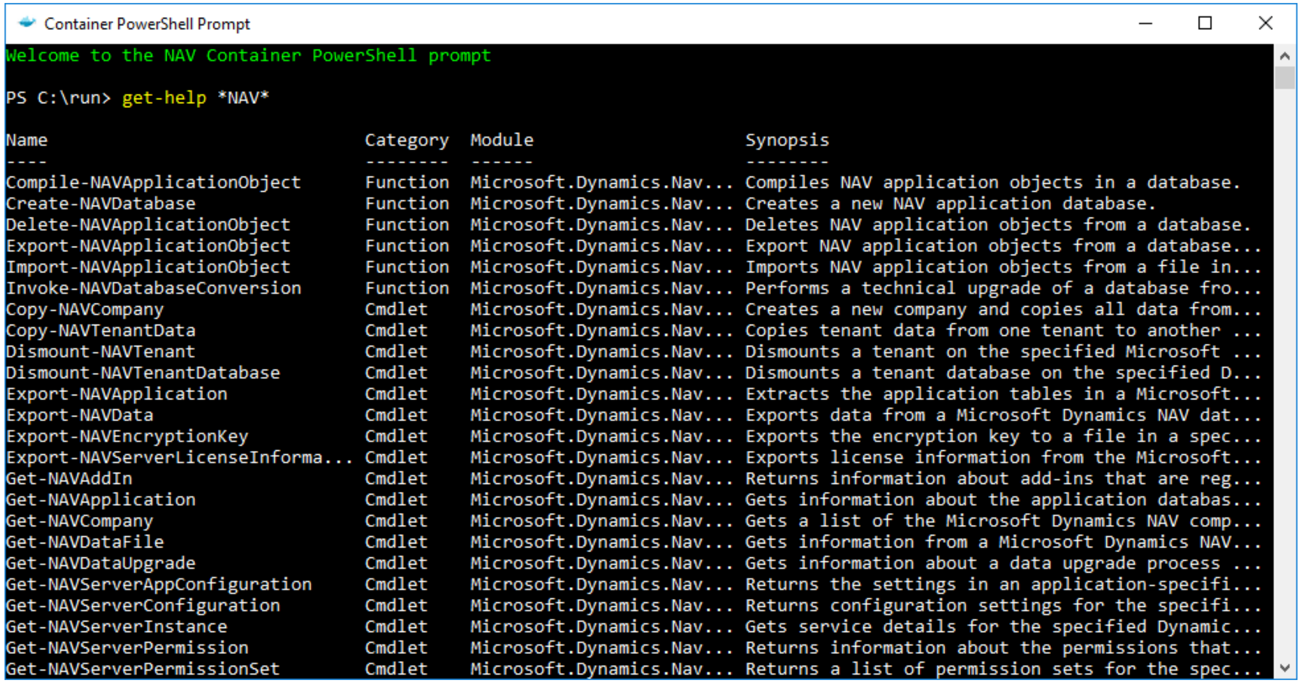
All NAV cmdlets are loaded inside the PowerShell prompt, ready to use.

Example:

Get-NavServerUser NAV

Will list all users in the NAV server instance (which is the default server instance in the container).

Note that **not** all commands will work inside the container. You cannot create a new server instance, for example – that is done by spinning up another container (the Docker way😉)



```
Welcome to the NAV Container PowerShell prompt

PS C:\run> get-help *NAV*

Name                              Category  Module                    Synopsis
----                              --------  ------                    --------
Compile-NAVApplicationObject      Function  Microsoft.Dynamics.Nav... Compiles NAV application objects in a database.
Create-NAVDatabase                Function  Microsoft.Dynamics.Nav... Creates a new NAV application database.
Delete-NAVApplicationObject       Function  Microsoft.Dynamics.Nav... Deletes NAV application objects from a database.
Export-NAVApplicationObject       Function  Microsoft.Dynamics.Nav... Export NAV application objects from a database...
Import-NAVApplicationObject       Function  Microsoft.Dynamics.Nav... Imports NAV application objects from a file in...
Invoke-NAVDatabaseConversion      Function  Microsoft.Dynamics.Nav... Performs a technical upgrade of a database fro...
Copy-NAVCompany                   Cmdlet    Microsoft.Dynamics.Nav... Creates a new company and copies all data from...
Copy-NAVTenantData                Cmdlet    Microsoft.Dynamics.Nav... Copies tenant data from one tenant to another ...
Dismount-NAVTenant                Cmdlet    Microsoft.Dynamics.Nav... Dismounts a tenant on the specified Microsoft ...
Dismount-NAVTenantDatabase        Cmdlet    Microsoft.Dynamics.Nav... Dismounts a tenant database on the specified D...
Export-NAVApplication             Cmdlet    Microsoft.Dynamics.Nav... Extracts the application tables in a Microsoft...
Export-NAVData                    Cmdlet    Microsoft.Dynamics.Nav... Exports data from a Microsoft Dynamics NAV dat...
Export-NAVEncryptionKey           Cmdlet    Microsoft.Dynamics.Nav... Exports the encryption key to a file in a spec...
Export-NAVServerLicenseInforma... Cmdlet    Microsoft.Dynamics.Nav... Exports license information from the Microsoft...
Get-NAVAddIn                      Cmdlet    Microsoft.Dynamics.Nav... Returns information about add-ins that are reg...
Get-NAVApplication                Cmdlet    Microsoft.Dynamics.Nav... Gets information about the application databas...
Get-NAVCompany                    Cmdlet    Microsoft.Dynamics.Nav... Gets a list of the Microsoft Dynamics NAV comp...
Get-NAVDataFile                   Cmdlet    Microsoft.Dynamics.Nav... Gets information from a Microsoft Dynamics NAV...
Get-NAVDataUpgrade                Cmdlet    Microsoft.Dynamics.Nav... Gets information about a data upgrade process ...
Get-NAVServerAppConfiguration     Cmdlet    Microsoft.Dynamics.Nav... Returns the settings in an application-specifi...
Get-NAVServerConfiguration        Cmdlet    Microsoft.Dynamics.Nav... Returns configuration settings for the specifi...
Get-NAVServerInstance             Cmdlet    Microsoft.Dynamics.Nav... Gets service details for the specified Dynamic...
Get-NAVServerPermission           Cmdlet    Microsoft.Dynamics.Nav... Returns information about the permissions that...
Get-NAVServerPermissionSet        Cmdlet    Microsoft.Dynamics.Nav... Returns a list of permission sets for the spec...
```

navserver CSIDE

C/SIDE a.k.a. the Classic Development Environment for the navserver container.

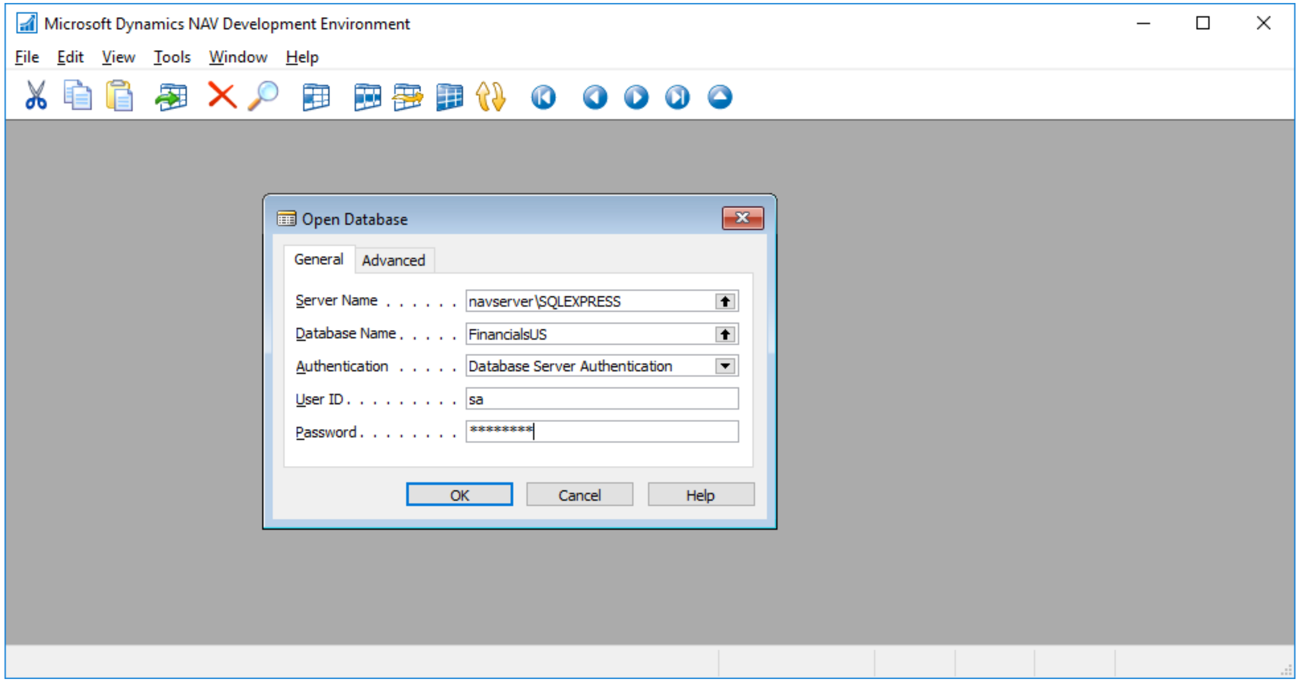
Note that C/SIDE is not there to support all classic development scenarios.

The primary reason for C/SIDE to be available is for the VS Code developer to be able to see and browse through the source of the base application.

Having said that, you can do the majority of classic development scenarios in C/SIDE.

Note that when you start C/SIDE you will be running Database Authentication and you have to login as **SA** and use the Workshop VM password.

Server name is **navserver\SQLEXPRESS** and the database name depends on which localization you are running.


Microsoft Dynamics NAV Development Environment
File Edit View Tools Window Help
Open Database
General
Advanced
OK
Cancel
Help

Server Name . . . . . . navserver\SQLEXPRESS

Database Name . . . . . FinancialsUS

Authentication . . . . . Database Server Authentication

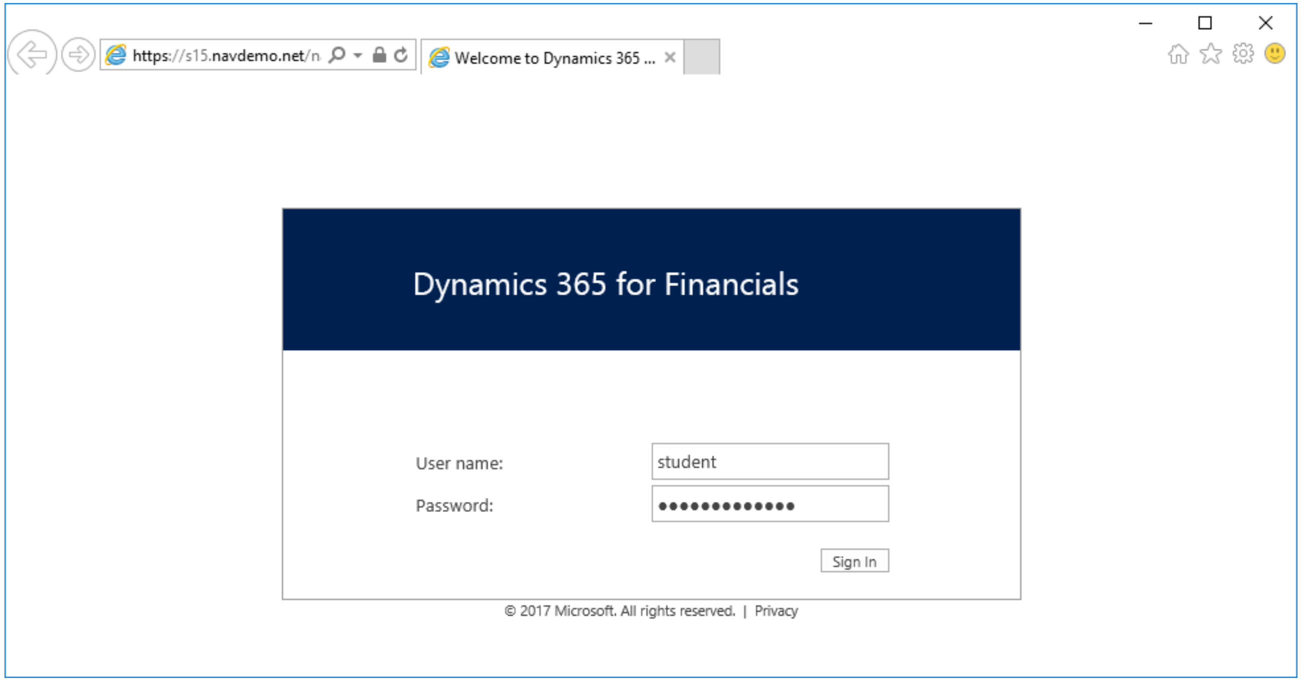
User ID . . . . sa

Password . . . . *********

navserver Web Client

Opens a browser with the Web client for the navserver container. The Web client is installed inside the container on IIS and the ports are exposed on the container and published to the host.

Login user name is **student** and the Workshop VM password is your password.


https://s15.navdemo.net/n
Welcome to Dynamics 365 ...
Dynamics 365 for Financials
Sign In
© 2017 Microsoft. All rights reserved. | Privacy

User name: student

Password: •••••••••••••

navserver Windows Client
Opens the Windows client for the navservercontainer.

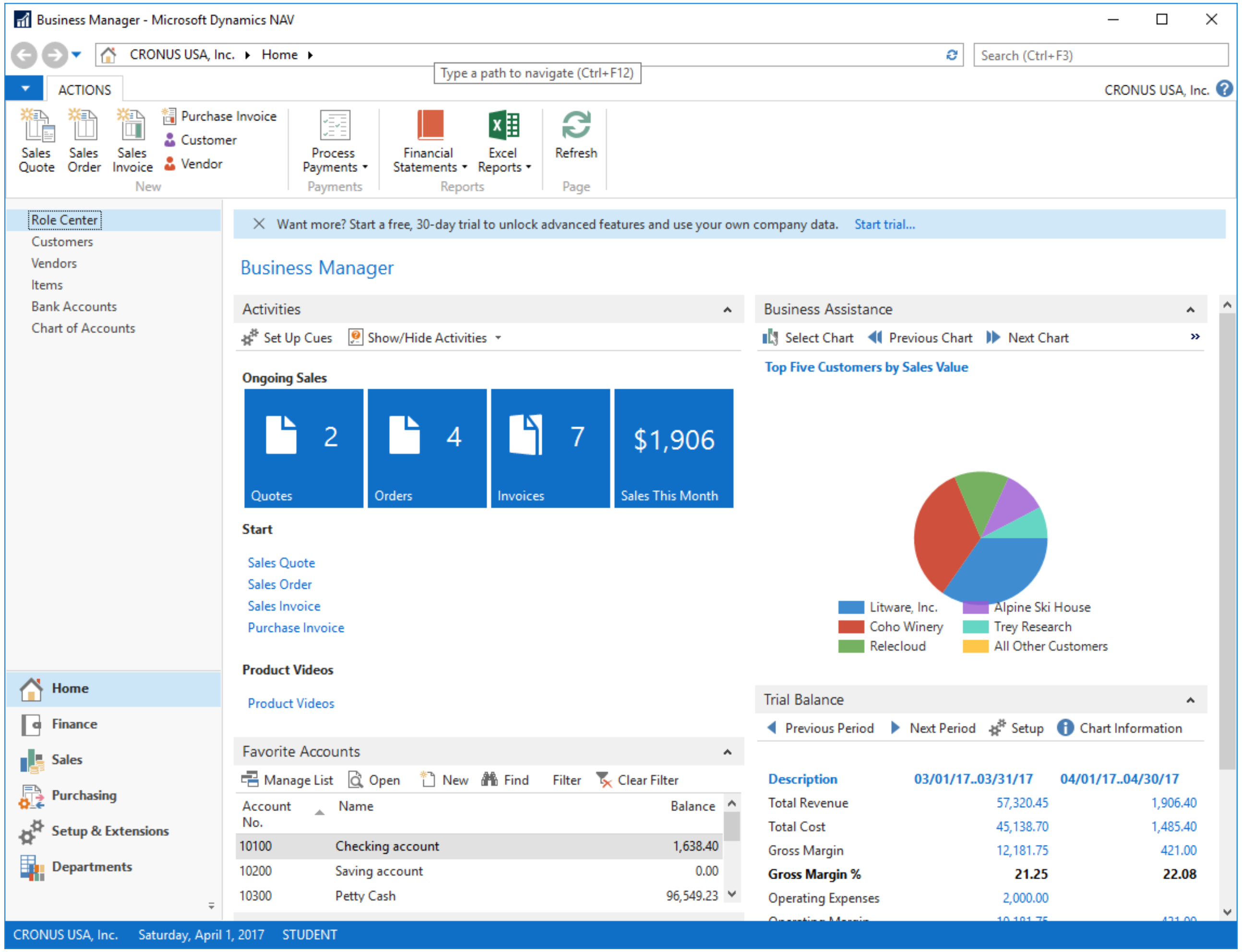
The Windows client is not installed on the Docker host even though it looks like it.

The Docker host shares a folder to the container called C:\Program Files (x86)\Microsoft Dynamics NAV – and the container then copies the files from that folder to the host.

This gives the best compatibility and allows the folder to be overridden if deploying a new container.

Login user name is **student** and the Workshop VM password is your password.

You can also install the Windows client using ClickOnce. There is a section about this later in the HOL.



Account No.	Name	Balance
10100	Checking account	1,638.40
10200	Saving account	0.00
10300	Petty Cash	96,549.23

Description	03/01/17..03/31/17	04/01/17..04/30/17
Total Revenue	57,320.45	1,906.40
Total Cost	45,138.70	1,485.40
Gross Margin	12,181.75	421.00
Gross Margin %	**21.25**	**22.08**
Operating Expenses	2,000.00	

Landing Page
The landing page was the starting point of your journey. You will find all info and links here necessary to connect and use the Workshop VM.

PowerShell ISE
PowerShell ISE running on the Docker host. This is every IT infrastructure gurus favorite tool and we will be using ISE throughout this Hands On Lab.

Visual Studio Code
Visual Studio Code is used for AL development and is not used in this Hands On Lab.
When launching the Workshop VM, the AL Language extension from the landing page is preinstalled. If you deploy a new NAV Container, you will have to uninstall and install a new AL Language extension.

Basic Docker commands

Let’s drill into some of the basic Docker commands to get a better understanding of what Docker is and how it works.

You can run these commands in PowerShell, but Docker is a simple Windows Executable and will run in a command prompt as well.

For simplicity reasons, we will use the Command Prompt.

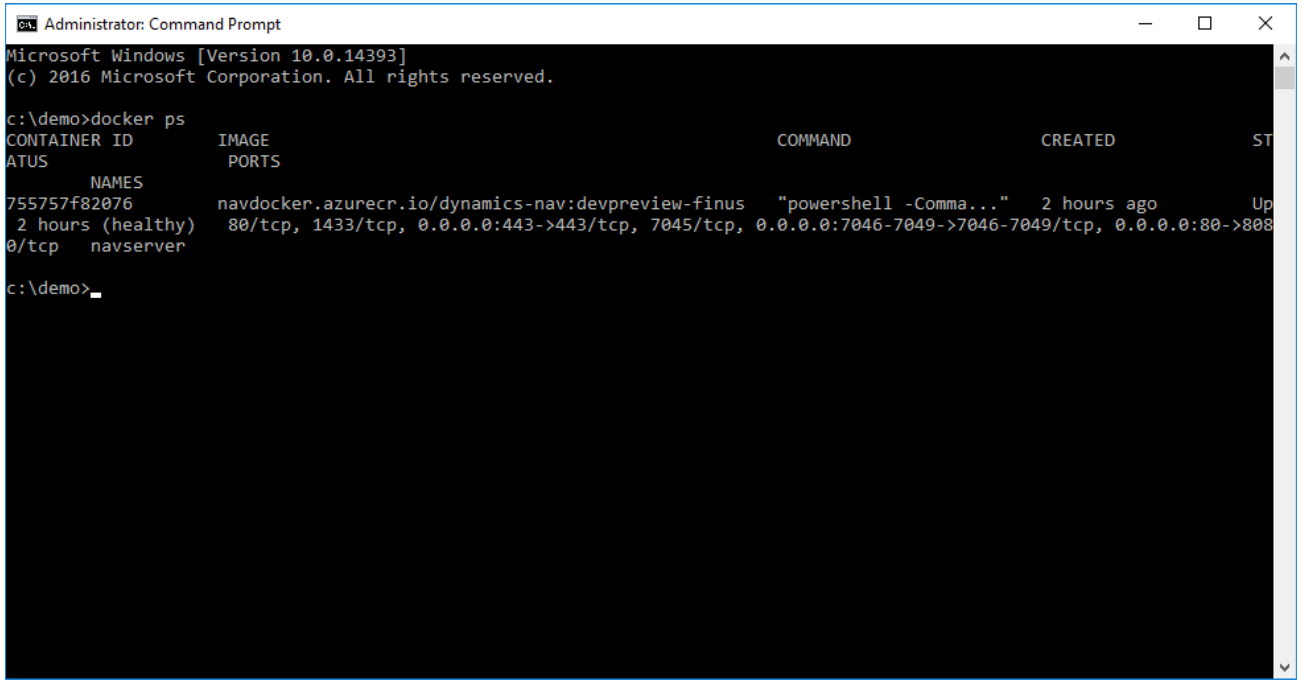
Open the Command Prompt and write:

docker ps

This gives you a list of all the running Docker Containers on your machine.

Take some time to Inspect the info:

- The container name is navserver.
- The container ID starts with 755757f82076.
- The container is based on the *microsoft/dynamics-nav:devpreview-finus* image.
- Ports 443 and 7046-7049 are all exposed on the Docker host, meaning you can access them from the outside.
- Port 8080 from the Docker container is published as port 80 on the Docker host.
- Ports 80, 1433 and 7045 are open for the host.

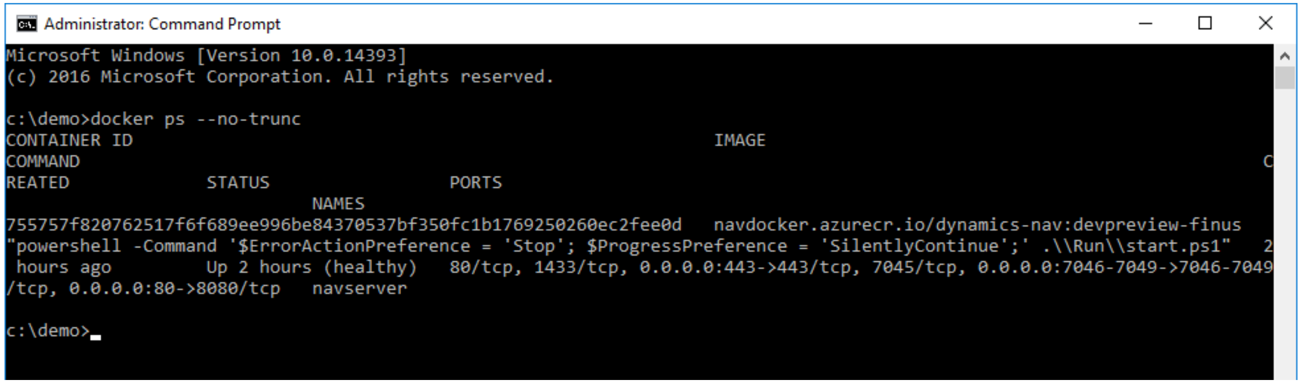


```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\demo>docker ps
CONTAINER ID        IMAGE                                                  COMMAND                  CREATED             ST
ATUS                 PORTS
        NAMES
755757f82076        navdocker.azurecr.io/dynamics-nav:devpreview-finus   "powershell -Comma..."   2 hours ago         Up
 2 hours (healthy)   80/tcp, 1433/tcp, 0.0.0.0:443->443/tcp, 7045/tcp, 0.0.0.0:7046-7049->7046-7049/tcp, 0.0.0.0:80->808
0/tcp   navserver

c:\demo>
```

You might wonder why the previous section says: *“The container ID starts with…”*. The reason for this is, that the ID really is a 64 digit globally unique hex identifier, but most time you can refer to the ID by specifying the first digits until your specification isn’t ambiguous.
You will get the full ID by typing:

docker ps --no-trunc

but if you only have one image you can identify it by writing the first digit – here: **7**



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\demo>docker ps --no-trunc
CONTAINER ID                                                         IMAGE
COMMAND                                                                                                                     C
REATED              STATUS                   PORTS
                                NAMES
755757f820762517f6f689ee996be84370537bf350fc1b1769250260ec2fee0d   navdocker.azurecr.io/dynamics-nav:devpreview-finus
"powershell -Command '$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';' .\\Run\\start.ps1"   2
 hours ago          Up 2 hours (healthy)   80/tcp, 1433/tcp, 0.0.0.0:443->443/tcp, 7045/tcp, 0.0.0.0:7046-7049->7046-7049
/tcp, 0.0.0.0:80->8080/tcp   navserver

c:\demo>
```

The next command to try is:

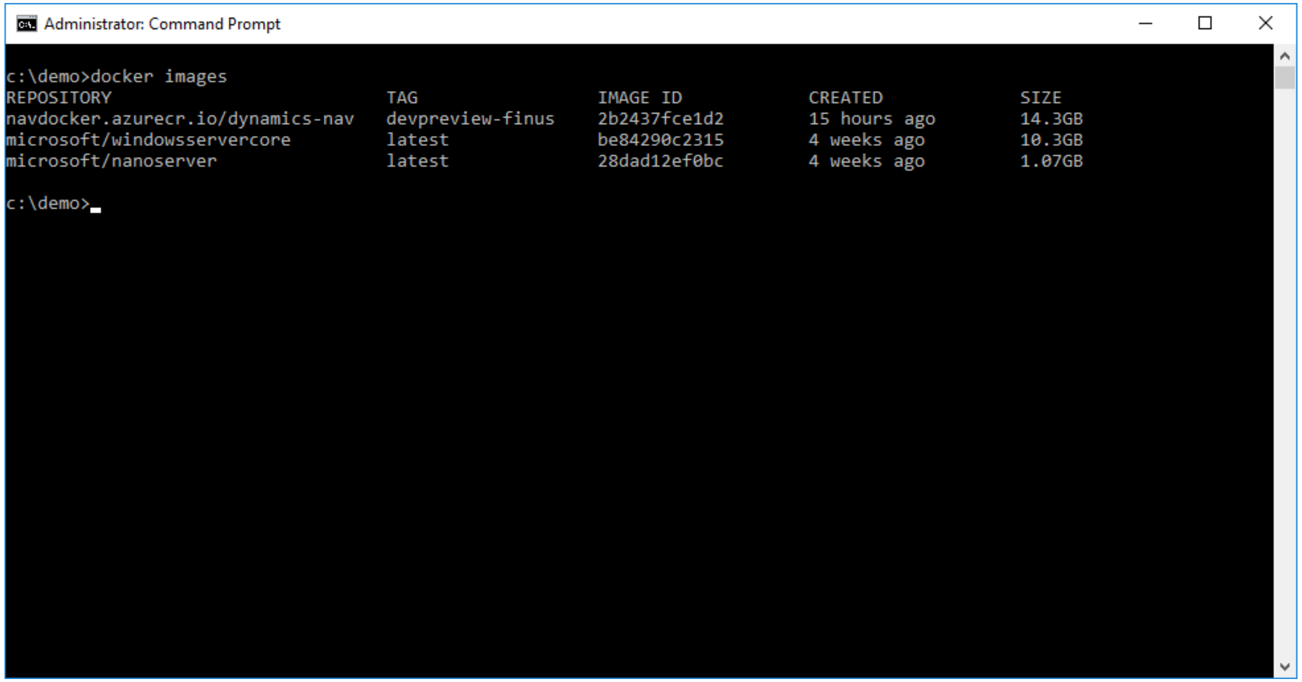
docker images

This gives you a list of all images available for you to run. In this picture there are the 2 Microsoft base images: Windows Server Core and Nano Server. Beside them, the latest devpreview image with Financials US localization.

A Docker image is really a set of services installed in a box (container) ready to run on demand.

A specific version of the NAV Docker image is a specific version (incl. localization) of NAV installed in a Container ready to run (ex. NAV 2017 CU7 DK).

The NAV Docker images are highly configurable and customizable.



```
c:\demo>docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
navdocker.azurecr.io/dynamics-nav   devpreview-finus    2b2437fce1d2        15 hours ago        14.3GB
microsoft/windowsservercore         latest              be84290c2315        4 weeks ago         10.3GB
microsoft/nanoserver                latest              28dad12ef0bc        4 weeks ago         1.07GB

c:\demo>
```

Now, try to run another instance of the dynamics-nav image you have available:

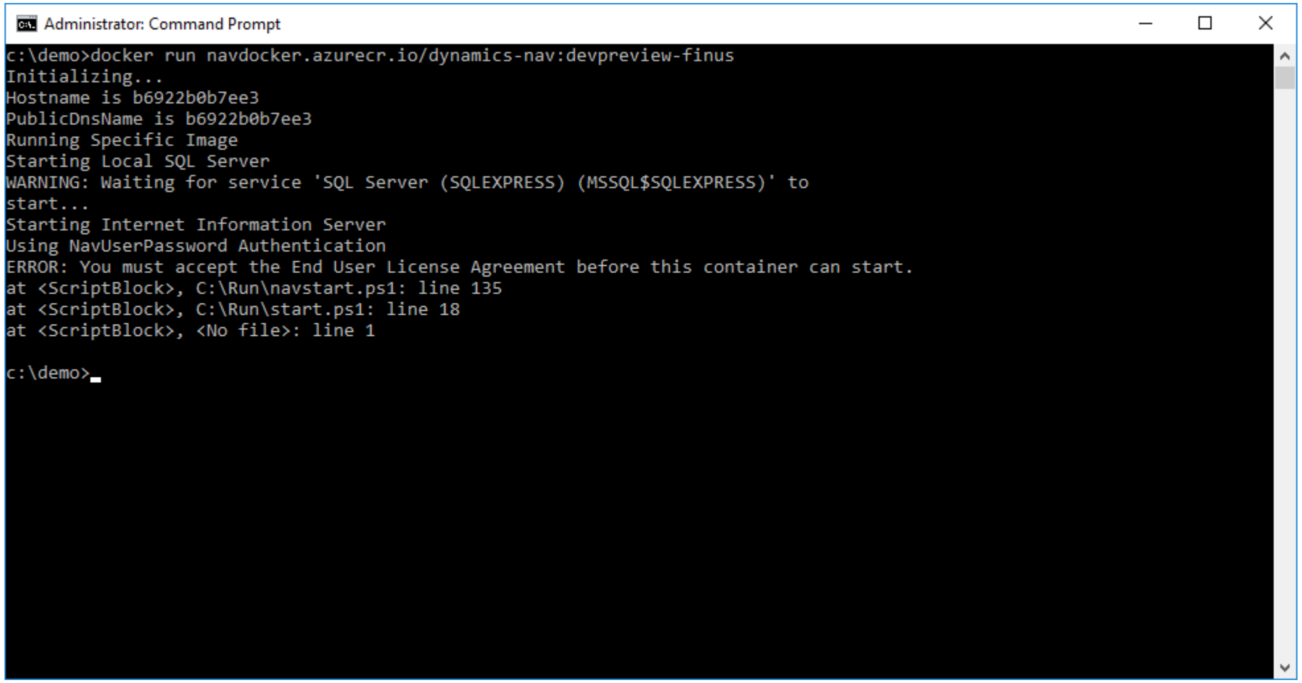
docker run microsoft/dynamics-nav:devpreview-finus

As the error indicates, you will have to accept the End User License Agreement before this container can start.

Use:

docker inspect --format='{{.Config.Labels.legal}}' microsoft/dynamics-nav:devpreview-finus

to view the legal documents for the preview.



```
c:\demo>docker run navdocker.azurecr.io/dynamics-nav:devpreview-finus
Initializing...
Hostname is b6922b0b7ee3
PublicDnsName is b6922b0b7ee3
Running Specific Image
Starting Local SQL Server
WARNING: Waiting for service 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' to
start...
Starting Internet Information Server
Using NavUserPassword Authentication
ERROR: You must accept the End User License Agreement before this container can start.
at <ScriptBlock>, C:\Run\navstart.ps1: line 135
at <ScriptBlock>, C:\Run\start.ps1: line 18
at <ScriptBlock>, <No file>: line 1

c:\demo>
```

Let’s run another instance of the image and accept the EULA:

docker run -e accept_eula=Y microsoft/dynamics-nav:devpreview-finus

Press Ctrl+C in the command prompt to exit the container and leave it running in the background.

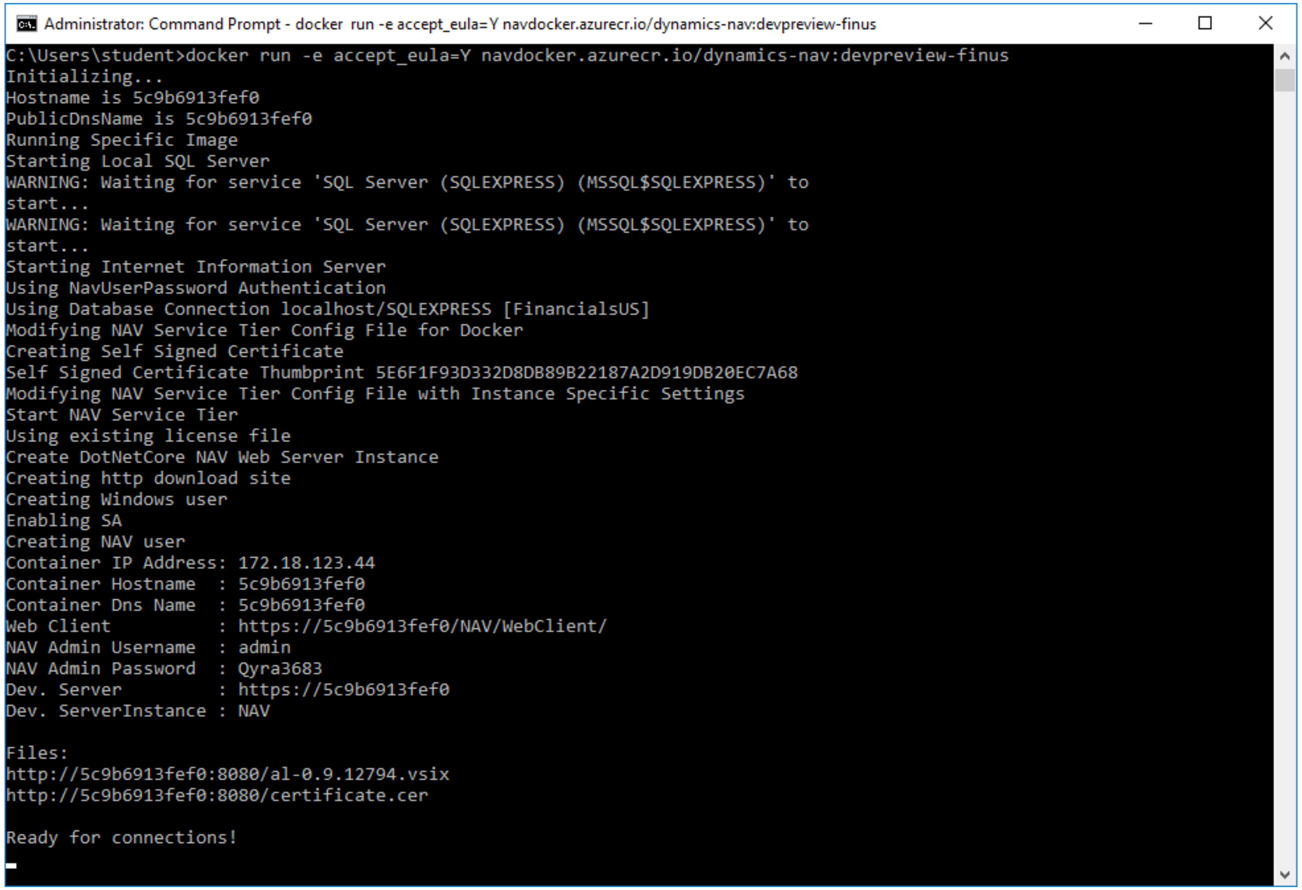
Now, run:

docker ps

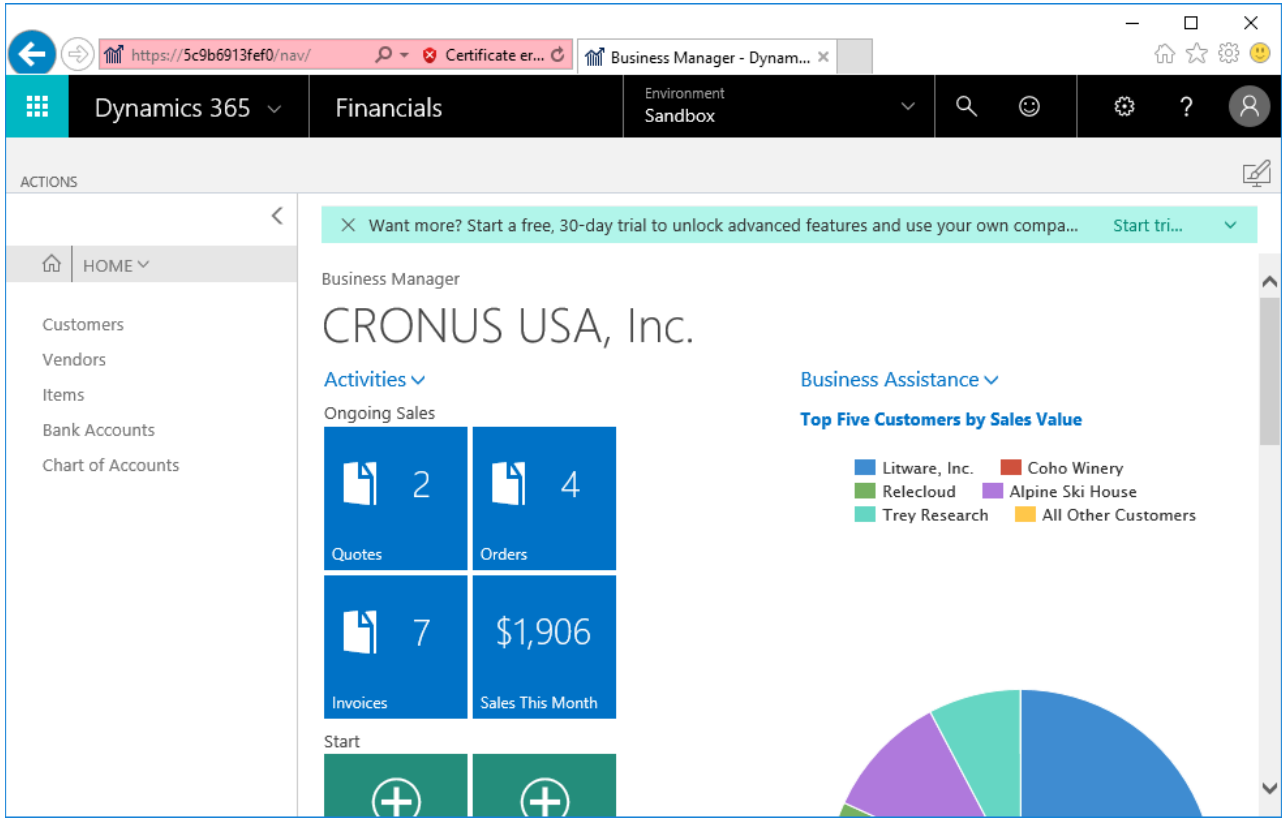
The command will show you two containers running. Inspect the difference in names, ports etc.

Note that Docker automatically assigns a readable name to the container if you don’t assign a name in the Docker run statement.

The original container will have ports exposed on the host, the new container will only have ports exposed on the container.


Administrator: Command Prompt - docker run -e accept_eula=Y navdocker.azurecr.io/dynamics-nav:devpreview-finus

```
C:\Users\student>docker run -e accept_eula=Y navdocker.azurecr.io/dynamics-nav:devpreview-finus
Initializing...
Hostname is 5c9b6913fef0
PublicDnsName is 5c9b6913fef0
Running Specific Image
Starting Local SQL Server
WARNING: Waiting for service 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' to
start...
WARNING: Waiting for service 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' to
start...
Starting Internet Information Server
Using NavUserPassword Authentication
Using Database Connection localhost/SQLEXPRESS [FinancialsUS]
Modifying NAV Service Tier Config File for Docker
Creating Self Signed Certificate
Self Signed Certificate Thumbprint 5E6F1F93D332D8DB89B22187A2D919DB20EC7A68
Modifying NAV Service Tier Config File with Instance Specific Settings
Start NAV Service Tier
Using existing license file
Create DotNetCore NAV Web Server Instance
Creating http download site
Creating Windows user
Enabling SA
Creating NAV user
Container IP Address: 172.18.123.44
Container Hostname  : 5c9b6913fef0
Container Dns Name  : 5c9b6913fef0
Web Client          : https://5c9b6913fef0/NAV/WebClient/
NAV Admin Username  : admin
NAV Admin Password  : Qyra3683
Dev. Server         : https://5c9b6913fef0
Dev. ServerInstance : NAV

Files:
http://5c9b6913fef0:8080/al-0.9.12794.vsix
http://5c9b6913fef0:8080/certificate.cer

Ready for connections!
```

Open the Web client in a browser. Ignore the certificate warnings for the self-signed certificate.	

https://5c9b6913fef0/nav/

Want more? Start a free, 30-day trial to unlock advanced features and use your own compa...

Now try to run

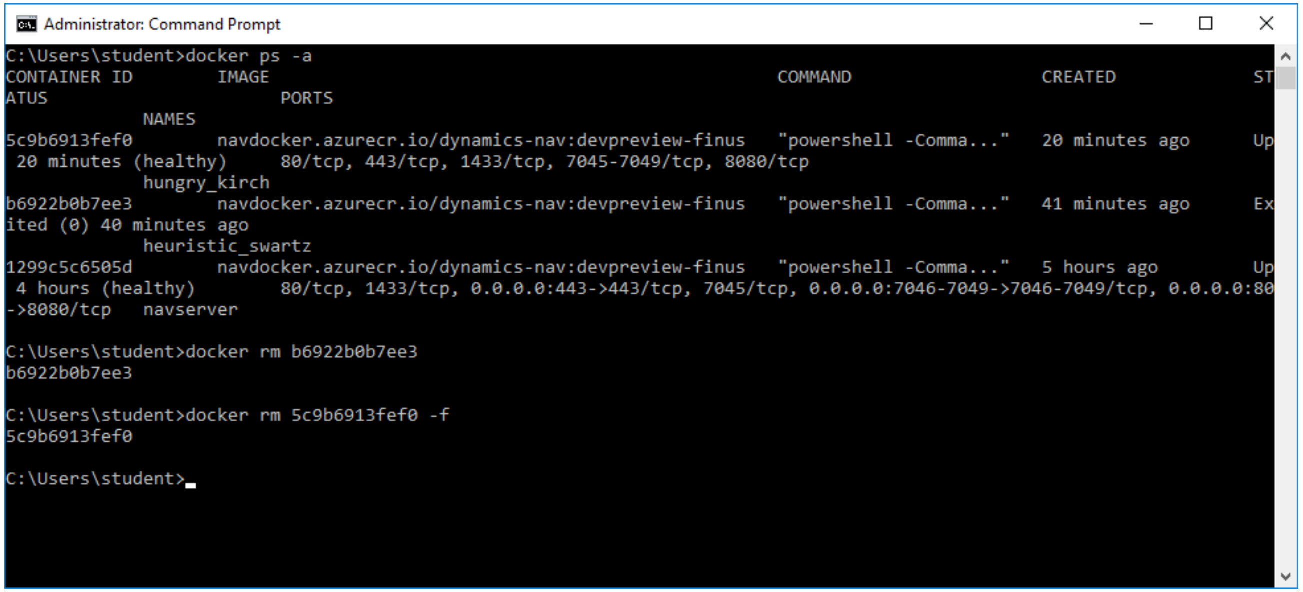
docker ps -a

which will show you all containers – running ones and exited ones. If you did try to run a container earlier without specifying the accept_eula=Y then you will have an exited container in the list.
Remove the dead container using

docker rm <containerid>

If you want to remove a running container you either need to stop it first or use the -f parameter:

docker rm <contianerid> -f



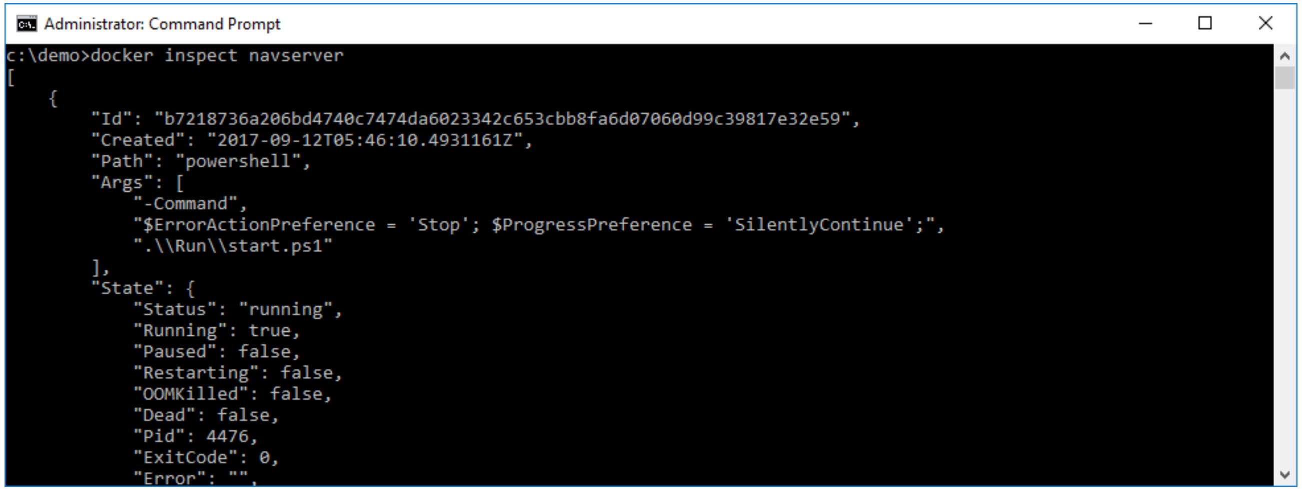
```
C:\Users\student>docker ps -a
CONTAINER ID        IMAGE                                                COMMAND                  CREATED             ST
ATUS                      PORTS
             NAMES
5c9b6913fef0        navdocker.azurecr.io/dynamics-nav:devpreview-finus   "powershell -Comma..."   20 minutes ago      Up
 20 minutes (healthy)     80/tcp, 443/tcp, 1433/tcp, 7045-7049/tcp, 8080/tcp
             hungry_kirch
b6922b0b7ee3        navdocker.azurecr.io/dynamics-nav:devpreview-finus   "powershell -Comma..."   41 minutes ago      Ex
ited (0) 40 minutes ago
             heuristic_swartz
1299c5c6505d        navdocker.azurecr.io/dynamics-nav:devpreview-finus   "powershell -Comma..."   5 hours ago         Up
 4 hours (healthy)        80/tcp, 1433/tcp, 0.0.0.0:443->443/tcp, 7045/tcp, 0.0.0.0:7046-7049->7046-7049/tcp, 0.0.0.0:80
->8080/tcp   navserver

C:\Users\student>docker rm b6922b0b7ee3
b6922b0b7ee3

C:\Users\student>docker rm 5c9b6913fef0 -f
5c9b6913fef0

C:\Users\student>_
```

Use

docker inspect navserver

to inspect settings, status, labels etc. on a container or an image.

You will also find network settings etc. if you look through the emitted JSON.

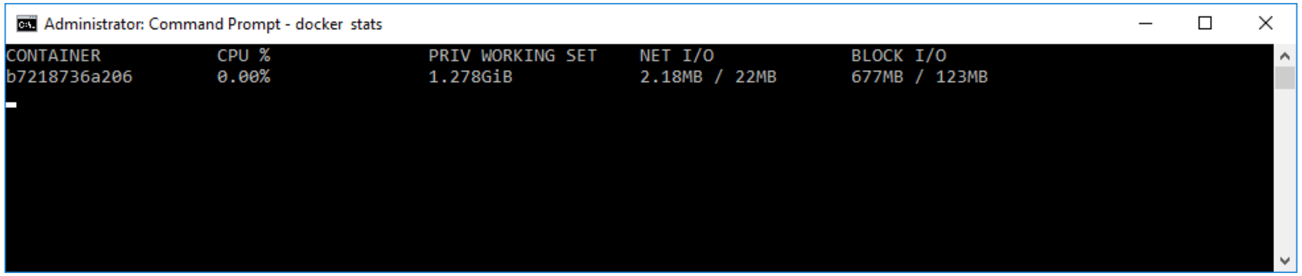


```
c:\demo>docker inspect navserver
[
    {
        "Id": "b7218736a206bd4740c7474da6023342c653cbb8fa6d07060d99c39817e32e59",
        "Created": "2017-09-12T05:46:10.4931161Z",
        "Path": "powershell",
        "Args": [
            "-Command",
            "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';",
            ".\\Run\\start.ps1"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 4476,
            "ExitCode": 0,
            "Error": "",
```

Use

docker stats

to get statistics from the currently running containers.



```
CONTAINER           CPU %               PRIV WORKING SET    NET I/O             BLOCK I/O
b7218736a206        0.00%               1.278GiB            2.18MB / 22MB       677MB / 123MB
_
```

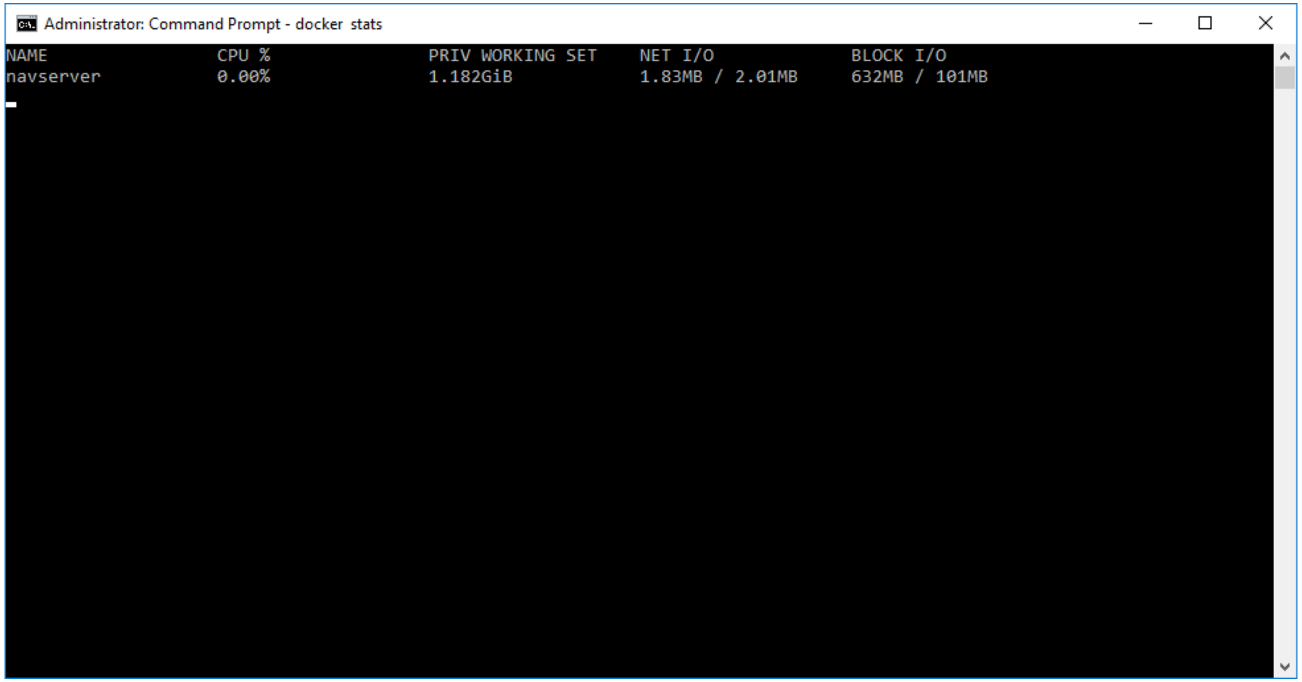
If you dislike the format of Docker stats (if you would like the container name included) you can modify the output by specifying a statsFormat property in the c:\users\student\.docker\config.json file.
In VSCode, create a new file with this content:

```
{
   "statsFormat": "table {{.Name}}\t{{.CPUPerc}}\t{{.MemUsage}}\t{{.NetIO}}\t{{.BlockIO}}"
}
```

And save it in c:\users\student\.docker\config.json.
Now re-run

docker stats

and you will see the info requested.
You can also add a section for psFormat etc.



```
NAME                CPU %               PRIV WORKING SET    NET I/O             BLOCK I/O
navserver           0.00%               1.182GiB            1.83MB / 2.01MB     632MB / 101MB
```

Use PowerShell ISE to modify files in the container

If you are new to Docker you might not yet be annoyed over how cumbersome it is to modify files in the container. You can connect using the PowerShell prompt or the command prompt, but since the file system is remote and you don't have a UI, you cannot edit files using Notepad.

But…

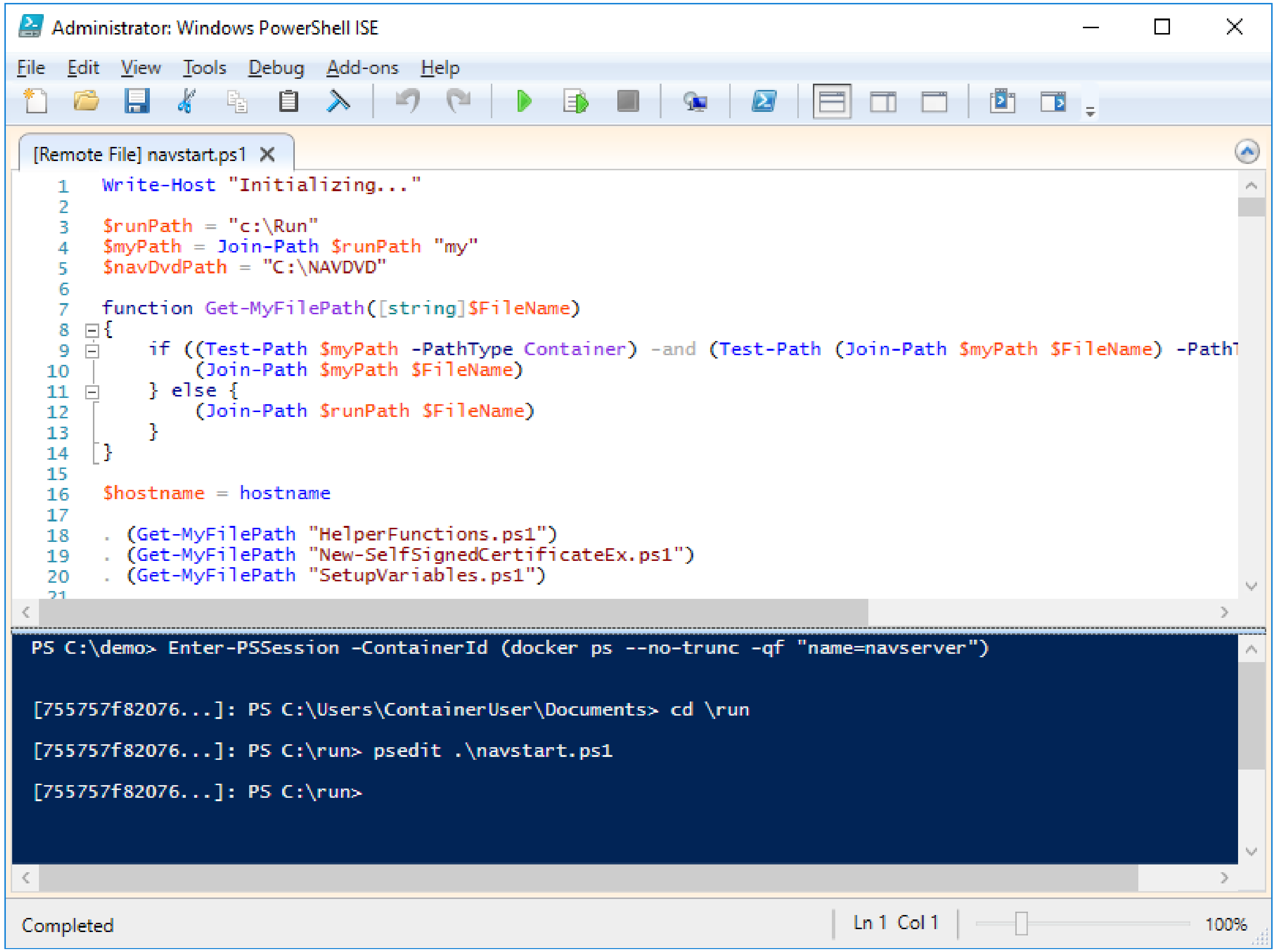
You can use ISE – it just requires a small trick.
Open ISE and run

Enter-PSSession -ContainerId (docker ps --no-trunc -qf "name=navserver")

Now you will enter a remote session in PowerShell (much like the navserver PowerShell Prompt) and inside of this you can use psEdit to edit files remotely without having to share folders and copy back and forth.

PS. The navcontainerhelper introduces a function which is called Enter-NavContainer <containername> which does exactly this.

Note that psEdit is an ISE specific function and does NOT work inside the navserver PowerShell Prompt.

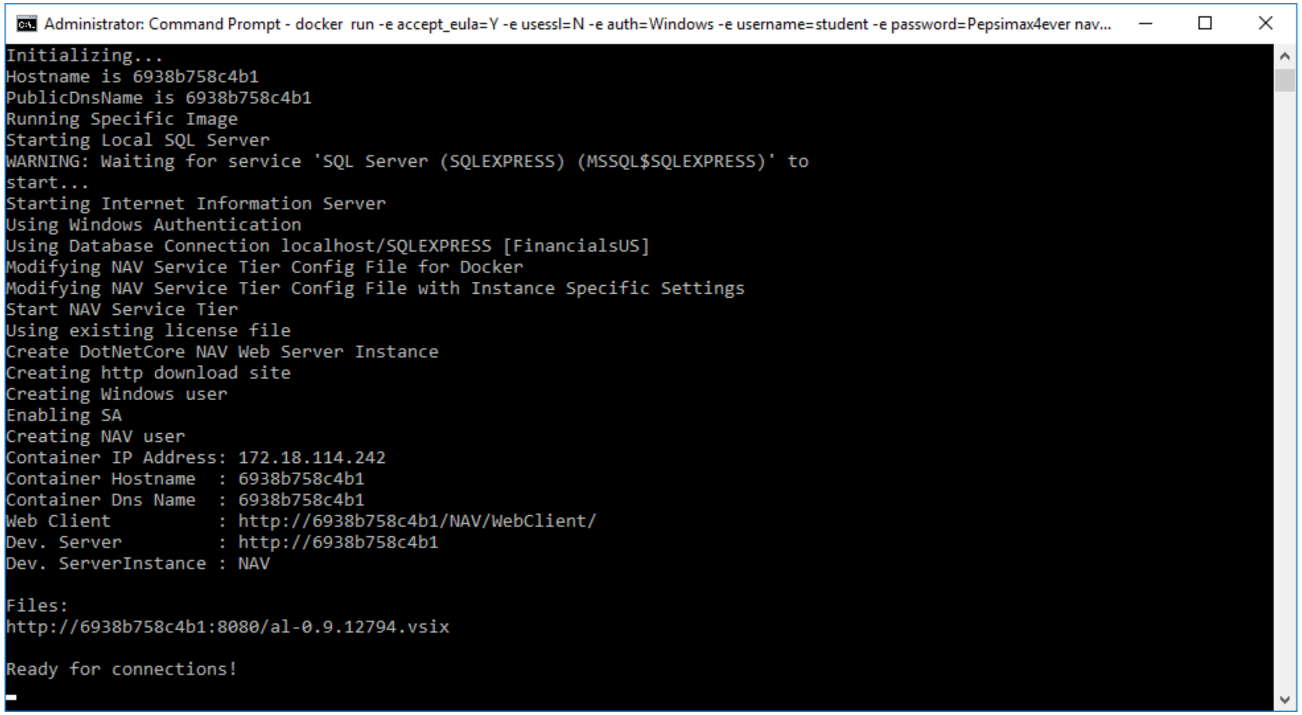

Administrator: Windows PowerShell ISE
File
Edit
View
Tools
Debug
Add-ons
Help
[Remote File] navstart.ps1
Completed
Ln 1 Col 1
100%

```
Write-Host "Initializing..."

$runPath = "c:\Run"
$myPath = Join-Path $runPath "my"
$navDvdPath = "C:\NAVDVD"

function Get-MyFilePath([string]$FileName)
{
    if ((Test-Path $myPath -PathType Container) -and (Test-Path (Join-Path $myPath $FileName) -PathT
        (Join-Path $myPath $FileName)
    } else {
        (Join-Path $runPath $FileName)
    }
}

$hostname = hostname

. (Get-MyFilePath "HelperFunctions.ps1")
. (Get-MyFilePath "New-SelfSignedCertificateEx.ps1")
. (Get-MyFilePath "SetupVariables.ps1")
```

```
PS C:\demo> Enter-PSSession -ContainerId (docker ps --no-trunc -qf "name=navserver")

[755757f82076...]: PS C:\Users\ContainerUser\Documents> cd \run

[755757f82076...]: PS C:\run> psedit .\navstart.ps1

[755757f82076...]: PS C:\run>
```

Advanced parameters

When using Docker run with the NAV image, there are a lot of different parameters you can use. All NAV image specific parameters are specified as environment variables (-e or –env).

There are a number of different parameters you can set when running the NAV Container. This command uses some of them: **docker run -e accept_eula=Y -e usessl=N -e auth=Windows -e username=student -e password=<password> --name test microsoft/dynamics-nav:devpreview-finus** If you specify the password of your student user, then this command will start NAV in a Container without SSL and using Windows Authentication. Note that this is a known hack, that you can use Windows Authentication between two machines if they share the same username and password. **Note** that the Web client is now without SSL and if you open it in a browser, you will find that you are logged directly into NAV. Use: **docker rm test -f** to remove the container named test.	

Administrator: Command Prompt - docker run -e accept_eula=Y -e usessl=N -e auth=Windows -e username=student -e password=Pepsimax4ever nav...

```
Initializing...
Hostname is 6938b758c4b1
PublicDnsName is 6938b758c4b1
Running Specific Image
Starting Local SQL Server
WARNING: Waiting for service 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' to
start...
Starting Internet Information Server
Using Windows Authentication
Using Database Connection localhost/SQLEXPRESS [FinancialsUS]
Modifying NAV Service Tier Config File for Docker
Modifying NAV Service Tier Config File with Instance Specific Settings
Start NAV Service Tier
Using existing license file
Create DotNetCore NAV Web Server Instance
Creating http download site
Creating Windows user
Enabling SA
Creating NAV user
Container IP Address: 172.18.114.242
Container Hostname  : 6938b758c4b1
Container Dns Name  : 6938b758c4b1
Web Client          : http://6938b758c4b1/NAV/WebClient/
Dev. Server         : http://6938b758c4b1
Dev. ServerInstance : NAV

Files:
http://6938b758c4b1:8080/al-0.9.12794.vsix

Ready for connections!
```

In the above example, test is the container name.
Most Docker commands take container ID or container name as parameter.
The container name however is not added to the DNS resolver and you cannot ping the container name.
In order to access the container using TCP or HTTP you need to use the hostname.
The default hostname is the first 10 characters of the container ID.
You can specify your own hostname using:

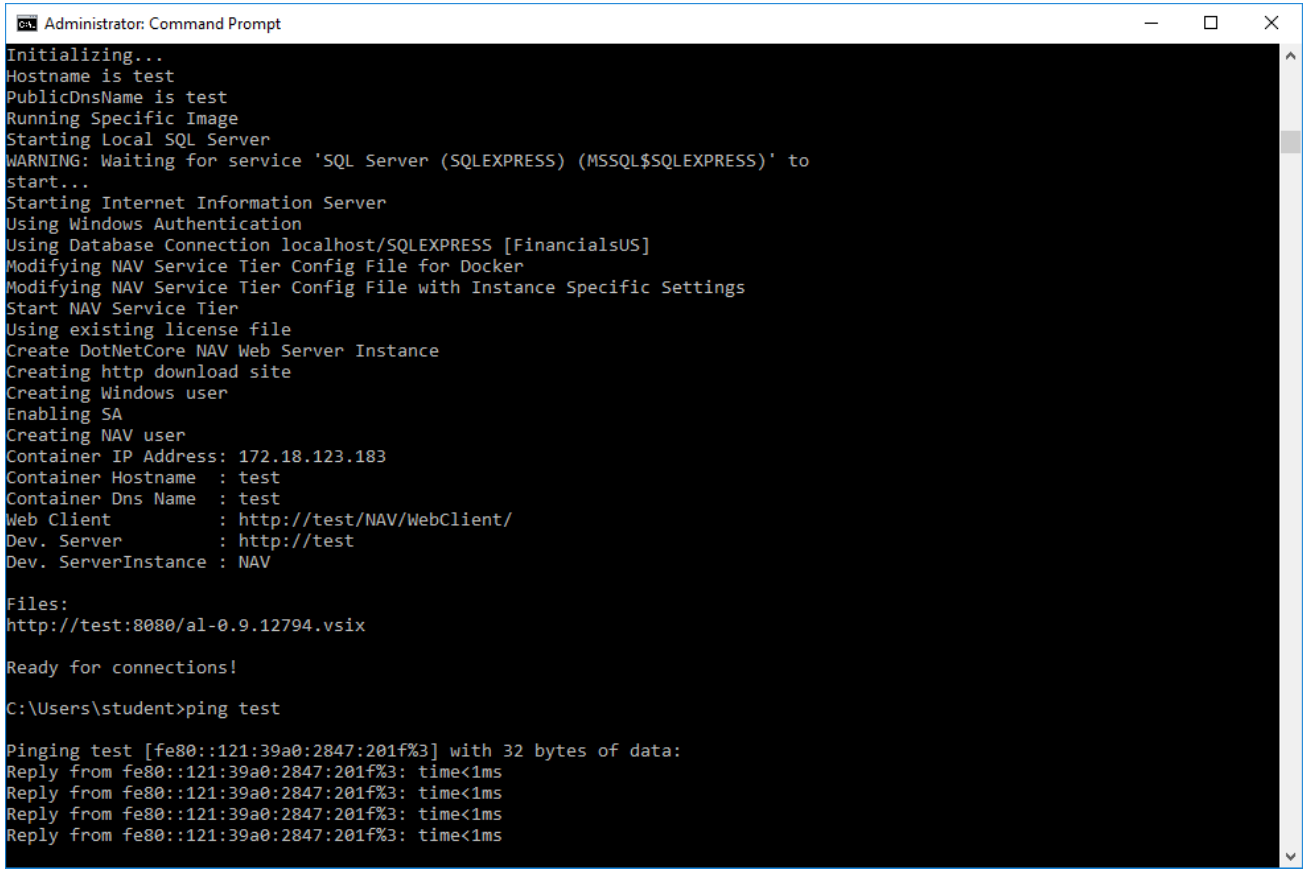
--hostname test

Docker will automatically maintain the IP address in the DNS resolution for the hostname, locally on the host.

You can also specify a public DNS name, which is the CNAME record, which points to your host if you are exposing the container to the world using a trusted certificate. PublicDnsName will default to the hostname.

-e publicDnsName=s11.navdemo.net

If you do not use SSL, the publicDnsName is only used for calculating properties like PublicWebBaseUrl, PublicSoapBaseUrl etc. in the config file.



```
Initializing...
Hostname is test
PublicDnsName is test
Running Specific Image
Starting Local SQL Server
WARNING: Waiting for service 'SQL Server (SQLEXPRESS) (MSSQL$SQLEXPRESS)' to
start...
Starting Internet Information Server
Using Windows Authentication
Using Database Connection localhost/SQLEXPRESS [FinancialsUS]
Modifying NAV Service Tier Config File for Docker
Modifying NAV Service Tier Config File with Instance Specific Settings
Start NAV Service Tier
Using existing license file
Create DotNetCore NAV Web Server Instance
Creating http download site
Creating Windows user
Enabling SA
Creating NAV user
Container IP Address: 172.18.123.183
Container Hostname  : test
Container Dns Name  : test
Web Client          : http://test/NAV/WebClient/
Dev. Server         : http://test
Dev. ServerInstance : NAV

Files:
http://test:8080/al-0.9.12794.vsix

Ready for connections!

C:\Users\student>ping test

Pinging test [fe80::121:39a0:2847:201f%3] with 32 bytes of data:
Reply from fe80::121:39a0:2847:201f%3: time<1ms
Reply from fe80::121:39a0:2847:201f%3: time<1ms
Reply from fe80::121:39a0:2847:201f%3: time<1ms
Reply from fe80::121:39a0:2847:201f%3: time<1ms
```

Pulling new NAV on Docker Images

Up until now, you have only used the images that already exist on the VM. If you want to use non-existing images or if you want to get an update to an existing image, you will have to pull these images. If you pull dynamics-nav:2017 without any specific cu you will always get the latest cu. Pulling dynamics-nav:2017-cu8 will only be updated if there are updates to the Windows Server Core or the NAV Generic Image.

All shipped versions of NAV since NAV 2016RTM are available on the public docker hub, where you also can find the EULA and the supported tags.

https://hub.docker.com/r/microsoft/dynamics-nav/

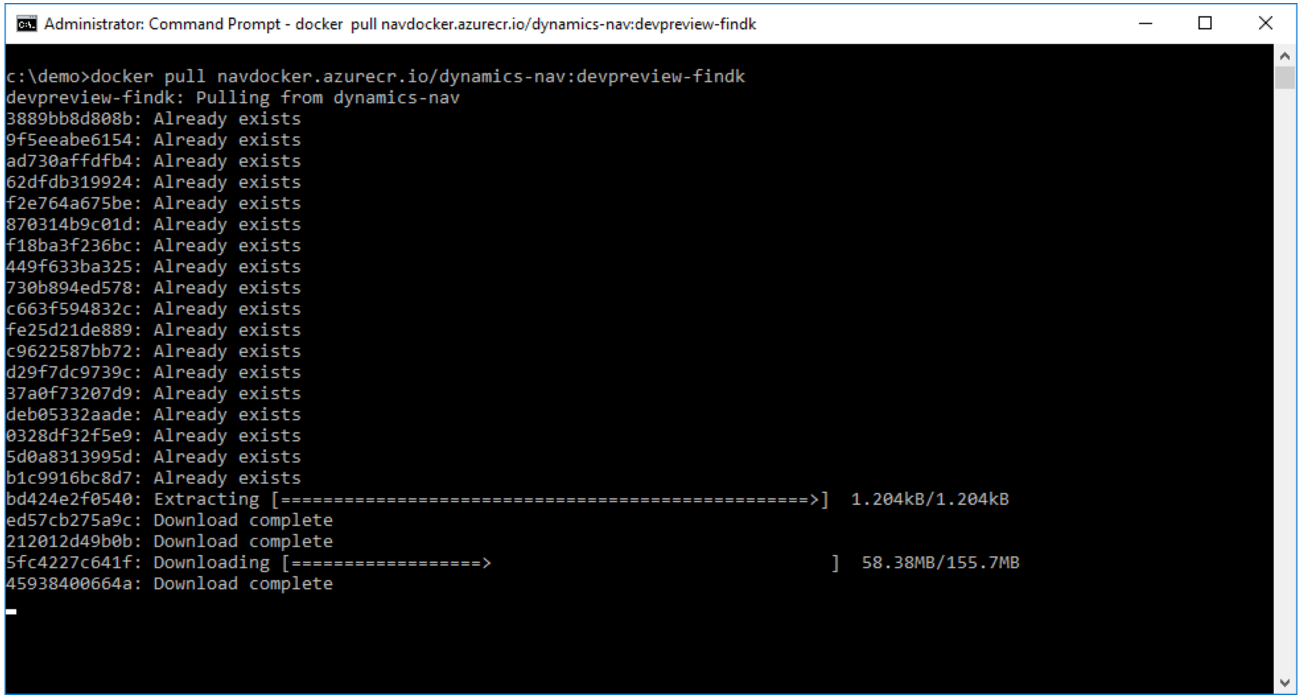
under Tags, you will find a list of all the tags in the public repository.

Docker images are constructed in layers. That means a Docker pull will only need to download those layers that are different from already downloaded layers. Try:

docker pull microsoft/dynamics-nav:devpreview-findk

You will see, that the first ~18 layers already exists and only a few layers will need to be downloaded.

The ~5 layers, which are downloaded is the difference between the US localization and the DK localization, so only this difference will have to be downloaded.



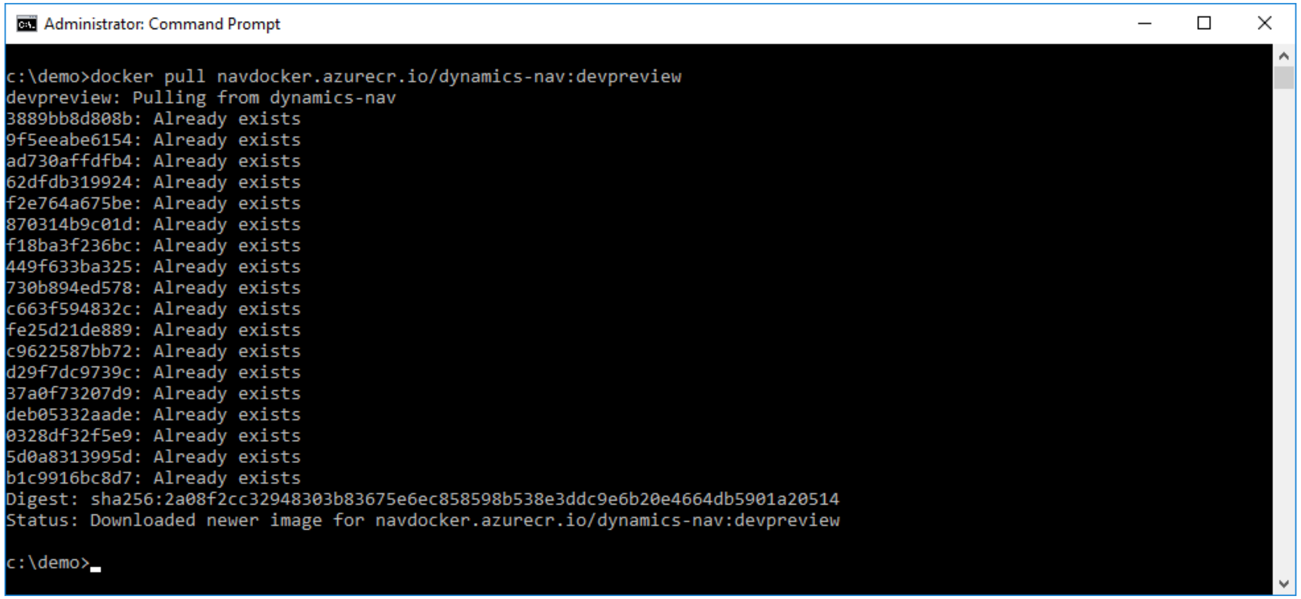
Administrator: Command Prompt - docker pull navdocker.azurecr.io/dynamics-nav:devpreview-findk

```
c:\demo>docker pull navdocker.azurecr.io/dynamics-nav:devpreview-findk
devpreview-findk: Pulling from dynamics-nav
3889bb8d808b: Already exists
9f5eeabe6154: Already exists
ad730affdfb4: Already exists
62dfdb319924: Already exists
f2e764a675be: Already exists
870314b9c01d: Already exists
f18ba3f236bc: Already exists
449f633ba325: Already exists
730b894ed578: Already exists
c663f594832c: Already exists
fe25d21de889: Already exists
c9622587bb72: Already exists
d29f7dc9739c: Already exists
37a0f73207d9: Already exists
deb05332aade: Already exists
0328df32f5e9: Already exists
5d0a8313995d: Already exists
b1c9916bc8d7: Already exists
bd424e2f0540: Extracting [=================================================>]  1.204kB/1.204kB
ed57cb275a9c: Download complete
212012d49b0b: Download complete
5fc4227c641f: Downloading [=================>                                ]  58.38MB/155.7MB
45938400664a: Download complete
```

As you might have guessed by now – if US and DK includes W1 (are built on top of W1), pulling W1 should not cause any downloads. Try:

docker pull microsoft/dynamics-nav:devpreview

Indeed – nothing to download, all layers already exist.



```
c:\demo>docker pull navdocker.azurecr.io/dynamics-nav:devpreview
devpreview: Pulling from dynamics-nav
3889bb8d808b: Already exists
9f5eeabe6154: Already exists
ad730affdfb4: Already exists
62dfdb319924: Already exists
f2e764a675be: Already exists
870314b9c01d: Already exists
f18ba3f236bc: Already exists
449f633ba325: Already exists
730b894ed578: Already exists
c663f594832c: Already exists
fe25d21de889: Already exists
c9622587bb72: Already exists
d29f7dc9739c: Already exists
37a0f73207d9: Already exists
deb05332aade: Already exists
0328df32f5e9: Already exists
5d0a8313995d: Already exists
b1c9916bc8d7: Already exists
Digest: sha256:2a08f2cc32948303b83675e6ec858598b538e3ddc9e6b20e4664db5901a20514
Status: Downloaded newer image for navdocker.azurecr.io/dynamics-nav:devpreview

c:\demo>
```

Using the navcontainerhelper

The navcontainerhelper is already installed on the workshop VM, but you can easily install it on your local box from the PowerShell Gallery using:

Install-module navcontainerhelper -force

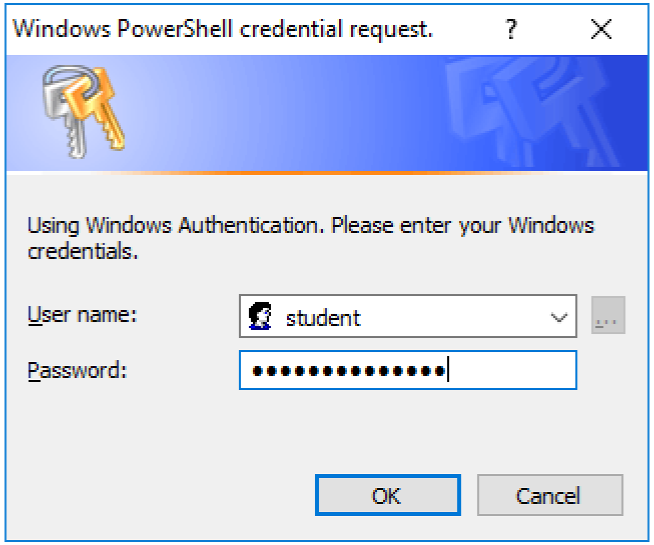
Even though Docker is a command line executable, you can use it in PowerShell like other executables and it does have some advantages. For that, we have created the navcontainerhelper, an open source project which is supposed to make it easier to work with containers.

Create a folder called C:\TEST. Start PowerShell ISE, create a new script and p aste in this line:

New-NavContainer -accept_eula -containerName myserver -includeCSide

save it as C:\TEST\start.ps1 and run it.

By default, the New-NavContainer will create a container running Windows Authentication, using the same image as the navserver container. Please supply the Windows Credentials of your workshop VM.



Using Windows Authentication. Please enter your Windows credentials.

User name: student

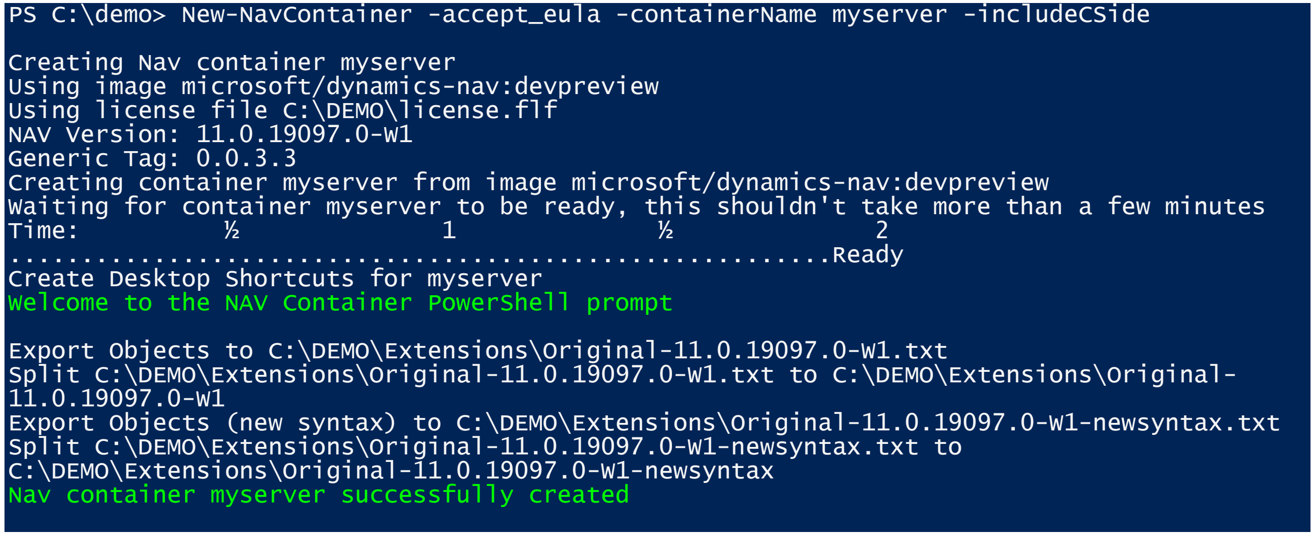
Password: ••••••••••••••

You should see an output, which is like the output on the right here.

Note that first time you run a specific version and include CSide, the container will automatically export all objects as text (baseline for object handling functions). You can avoid this by adding -doNotExportObjectsToText

You can press F5 again and again and the script will automatically remove the old container and start a fresh, this time without exporting objects.

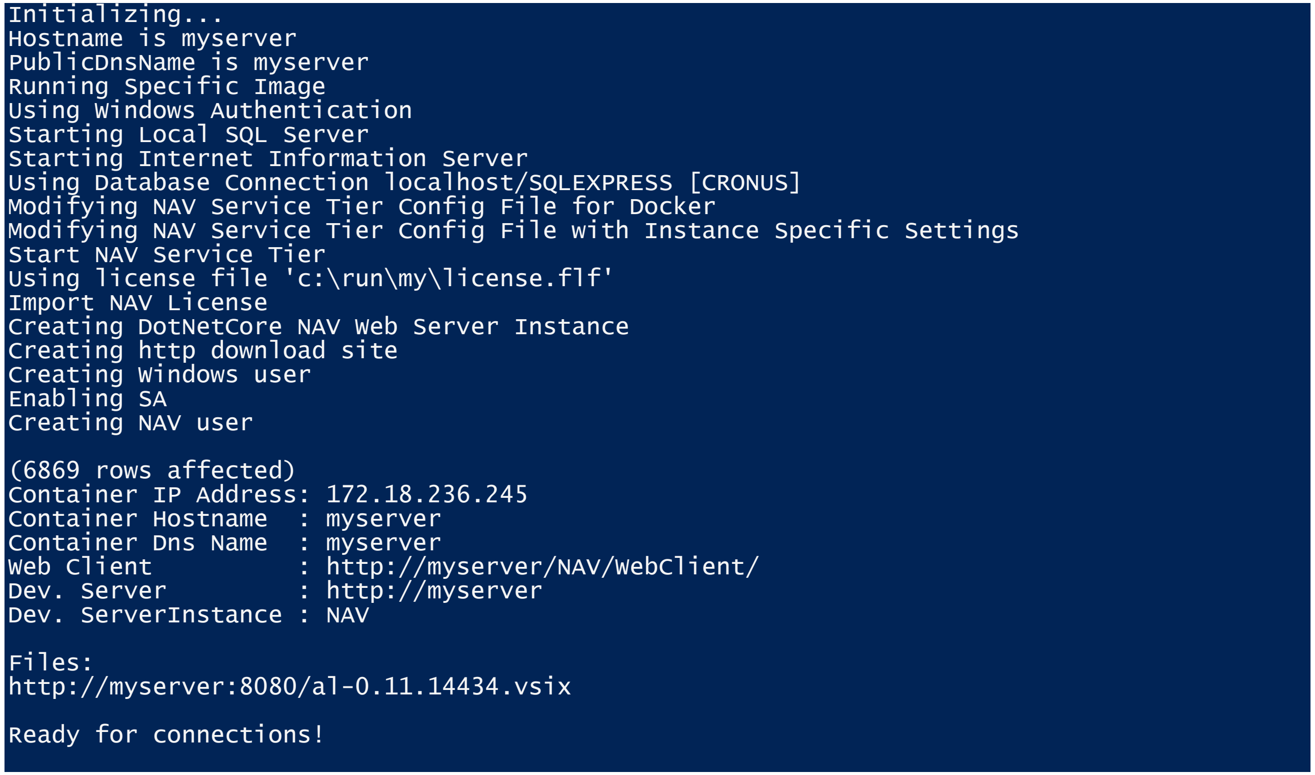
Note that you will have a new set of shortcuts on the desktop to connect to your myserver container.



```
PS C:\demo> New-NavContainer -accept_eula -containerName myserver -includeCSide

Creating Nav container myserver
Using image microsoft/dynamics-nav:devpreview
Using license file C:\DEMO\license.flf
NAV Version: 11.0.19097.0-W1
Generic Tag: 0.0.3.3
Creating container myserver from image microsoft/dynamics-nav:devpreview
Waiting for container myserver to be ready, this shouldn't take more than a few minutes
Time:          ½              1              ½              2
..............................................................Ready
Create Desktop Shortcuts for myserver
Welcome to the NAV Container PowerShell prompt

Export Objects to C:\DEMO\Extensions\Original-11.0.19097.0-W1.txt
Split C:\DEMO\Extensions\Original-11.0.19097.0-W1.txt to C:\DEMO\Extensions\Original-
11.0.19097.0-W1
Export Objects (new syntax) to C:\DEMO\Extensions\Original-11.0.19097.0-W1-newsyntax.txt
Split C:\DEMO\Extensions\Original-11.0.19097.0-W1-newsyntax.txt to
C:\DEMO\Extensions\Original-11.0.19097.0-W1-newsyntax
Nav container myserver successfully created
```

If you want to see the actual output of the container, write:

docker logs myserver

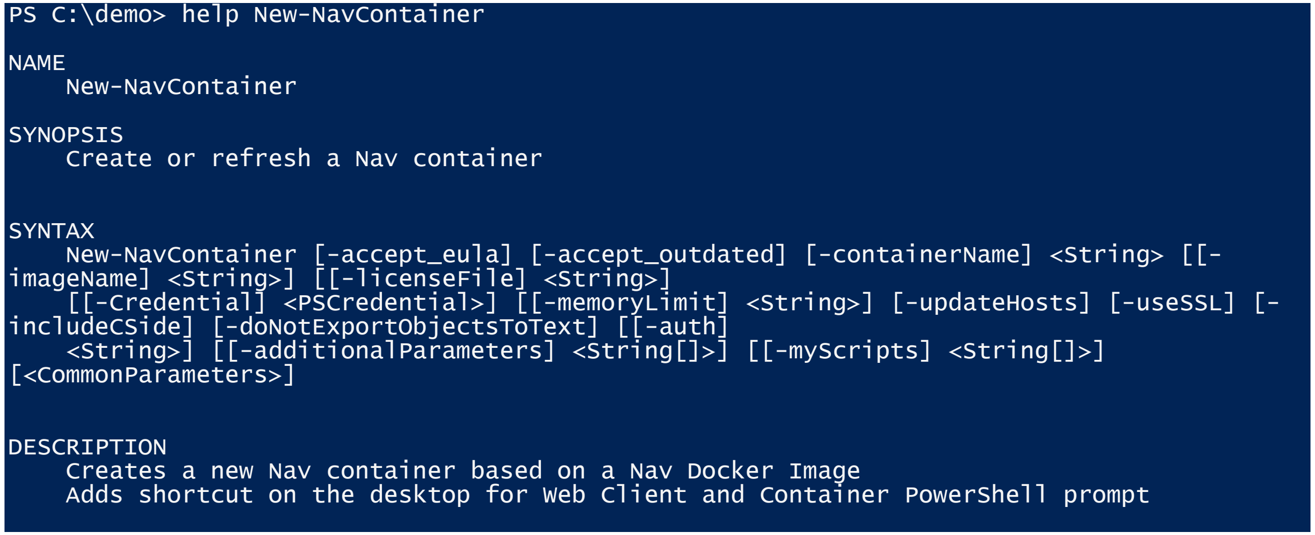


```
Initializing...
Hostname is myserver
PublicDnsName is myserver
Running Specific Image
Using Windows Authentication
Starting Local SQL Server
Starting Internet Information Server
Using Database Connection localhost/SQLEXPRESS [CRONUS]
Modifying NAV Service Tier Config File for Docker
Modifying NAV Service Tier Config File with Instance Specific Settings
Start NAV Service Tier
Using license file 'c:\run\my\license.flf'
Import NAV License
Creating DotNetCore NAV Web Server Instance
Creating http download site
Creating Windows user
Enabling SA
Creating NAV user

(6869 rows affected)
Container IP Address: 172.18.236.245
Container Hostname  : myserver
Container Dns Name  : myserver
Web Client          : http://myserver/NAV/WebClient/
Dev. Server         : http://myserver
Dev. ServerInstance : NAV

Files:
http://myserver:8080/al-0.11.14434.vsix

Ready for connections!
```

You can use

help new-navcontainer

To list all parameters available in the new-navcontainer function.



```
PS C:\demo> help New-NavContainer

NAME
    New-NavContainer

SYNOPSIS
    Create or refresh a Nav container


SYNTAX
    New-NavContainer [-accept_eula] [-accept_outdated] [-containerName] <String> [[-
imageName] <String>] [[-licenseFile] <String>]
    [[-Credential] <PSCredential>] [[-memoryLimit] <String>] [-updateHosts] [-useSSL] [-
includeCSide] [-doNotExportObjectsToText] [[-auth]
    <String>] [[-additionalParameters] <String[]>] [[-myScripts] <String[]>]
[<CommonParameters>]


DESCRIPTION
    Creates a new Nav container based on a Nav Docker Image
    Adds shortcut on the desktop for Web Client and Container PowerShell prompt
```

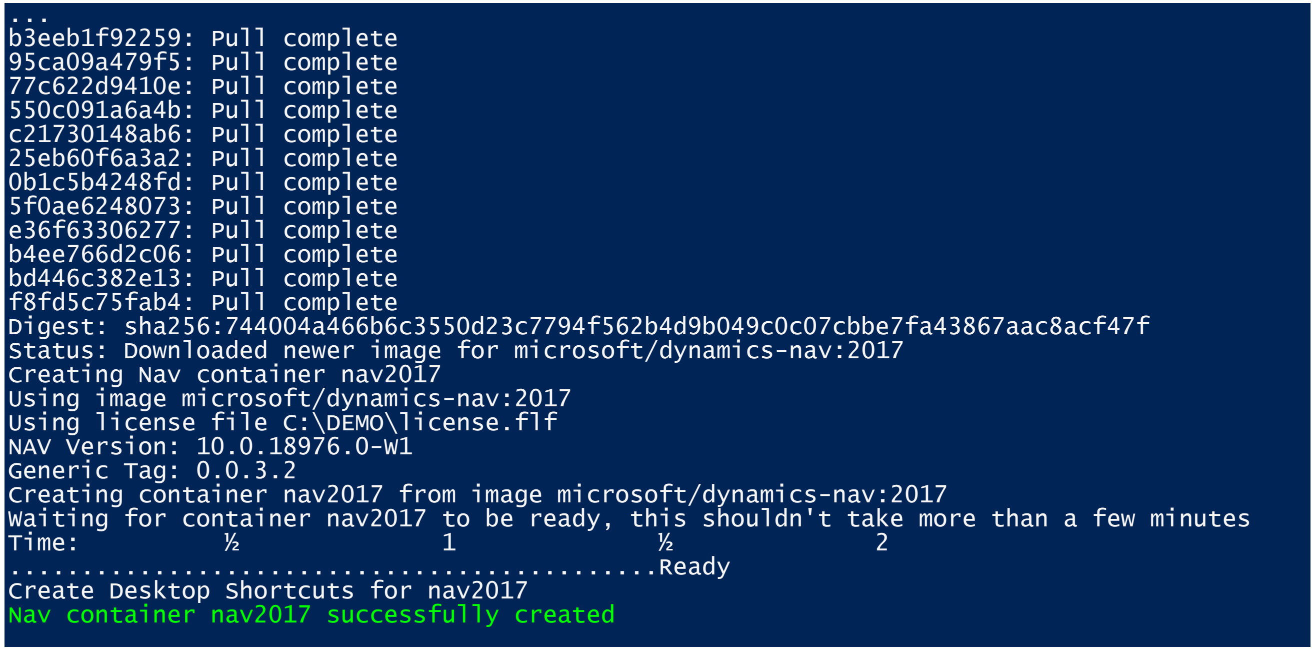
If you want to spin up a new container with NAV 2017, you can write:

New-NavContainer -accept_eula -containerName nav2017 -imageName microsoft/dynamics-nav:2017 -includeCSide -doNotExportObjectsToText

You will see, that the function automatically pulls the NAV 2017 CU12 W1 image and it will take some time to complete the pull as most of the layers are changed.

The Generic Tag here is 0.0.3.2 (previous was 0.0.3.3)

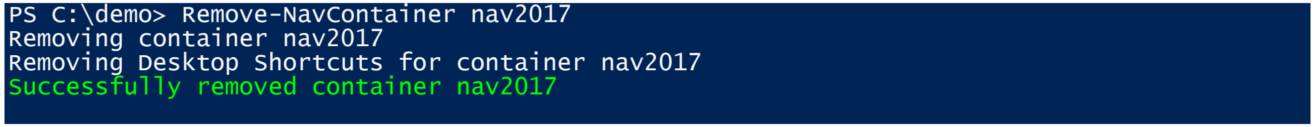
Again you will find shortcuts on the desktop to connect to your nav 2017 container.



```
...
b3eeb1f92259: Pull complete
95ca09a479f5: Pull complete
77c622d9410e: Pull complete
550c091a6a4b: Pull complete
c21730148ab6: Pull complete
25eb60f6a3a2: Pull complete
0b1c5b4248fd: Pull complete
5f0ae6248073: Pull complete
e36f63306277: Pull complete
b4ee766d2c06: Pull complete
bd446c382e13: Pull complete
f8fd5c75fab4: Pull complete
Digest: sha256:744004a466b6c3550d23c7794f562b4d9b049c0c07cbbe7fa43867aac8acf47f
Status: Downloaded newer image for microsoft/dynamics-nav:2017
Creating Nav container nav2017
Using image microsoft/dynamics-nav:2017
Using license file C:\DEMO\license.flf
NAV Version: 10.0.18976.0-W1
Generic Tag: 0.0.3.2
Creating container nav2017 from image microsoft/dynamics-nav:2017
Waiting for container nav2017 to be ready, this shouldn't take more than a few minutes
Time:          ½              1              ½              2
.............................................Ready
Create Desktop Shortcuts for nav2017
Nav container nav2017 successfully created
```

Remove-NavContainer nav2017

Will clean up after your Nav 2017 container



```
PS C:\demo> Remove-NavContainer nav2017
Removing container nav2017
Removing Desktop Shortcuts for container nav2017
Successfully removed container nav2017
```

Using a different database server

Up until now, we have been using NAV in a Container with the database living inside the same container. That is convenient when doing demos, but frequently you probably want to run the database on a different SQL Server or maybe even on Azure SQL. It is very easy to point out a different database server, instance, and name on the command line and if your containers are set up with gMSA (Group Managed Service Accounts) and Windows Authentication this should be sufficient to connect.

BUT…

The Workshop VMs do NOT use gMSA (as this would require the setup of AD servers), meaning that connecting to a different database server requires a little more work. What you will do here is to override the SetupDatabase script. Overriding scripts is an important feature of the NAV on Docker Images, and it really provides the flexibility needed for setting up a system as complex as NAV.

In the end we'll have our new container ”myserver” running NAV and we'll just reuse the first container ”navserver” as a database server. We could also use a new container containing only SQL Server, but we'll be faster this way.

Copy this script, paste it into PowerShell ISE, modify the "<Password>" with the password for your Workshop VM, and save the script as c:\test\SetupDatabase.ps1.

Inside the container, the primary setup runner (**c:\run\navstart.ps1**), will call several scripts in the c:\run folder to setup variables, database, license etc.

For each of these scripts, it will check whether there is a script in **c:\run\my** which overrides the behavior. The overriding script can determine if it is necessary to call the default behavior, but it needs to take over the responsibility of the script.

Specifying scripts in the myscripts parameter for new-navcontainer will automatically copy these scripts to the c:\run\my folder inside the container.

The NAV on Docker documentation talks about all the different scripts you can override and typical scenarios for why you would want to do it.

```
if (!$RestartingInstance) {

    Write-Host "Change Database Connection"
    $DatabaseServer = "navserver"
    $DatabaseInstance = "SQLEXPRESS"
    $DatabaseName = "CRONUS"
    $DatabaseUserName = "sa"
    $DatabasePassword = "<Password>"
    $EncryptionPassword = "1234abcd!1234abcd"
    $TrustSQLServerCertificate = $true

    $DatabaseSecurePassword = ConvertTo-SecureString -String $DatabasePassword -AsPlainText -Force
    $DatabaseCredentials = New-Object PSCredential -ArgumentList $DatabaseUserName, $DatabaseSecurePassword

    $EncryptionKeyPath = Join-Path $ServiceTierFolder 'DynamicsNAV.key'
    $EncryptionSecurePassword = ConvertTo-SecureString -String $EncryptionPassword -AsPlainText -Force
    New-NAVEncryptionKey -KeyPath $EncryptionKeyPath -Password $EncryptionSecurePassword -Force | Out-Null

    Write-Host "Import Encryption Key"
    Import-NAVEncryptionKey -ServerInstance NAV `
                            -ApplicationDatabaseServer $DatabaseServer `
                            -ApplicationDatabaseCredentials $DatabaseCredentials `
                            -ApplicationDatabaseName $DatabaseName `
                            -KeyPath $EncryptionKeyPath `
                            -Password $EncryptionSecurePassword `
                            -WarningAction SilentlyContinue `
                            -Force

    Set-NAVServerConfiguration -ServerInstance "NAV" -KeyName "EnableSqlConnectionEncryption" -KeyValue "true" -WarningAction SilentlyContinue
    Set-NAVServerConfiguration -ServerInstance "NAV" -KeyName "TrustSQLServerCertificate" -KeyValue $TrustSQLServerCertificate.Tostring().ToLowerInvariant() -WarningAction SilentlyContinue
    Set-NavServerConfiguration -serverinstance "NAV" -databaseCredentials $DatabaseCredentials
}
```

Replace the content of start.ps1 with this:

```
New-NavContainer -accept_eula `
                 -containerName myserver `
                 -includeCSide `
                 -auth NavUserPassword `
                 -additionalParameters @("-e databaseServer=navserver","-e databaseInstance=SQLEXPRESS","-e databaseName=CRONUS") `
                 -myScripts @("c:\test\setupdatabase.ps1")
```

Press F5 and wait for the container to complete.

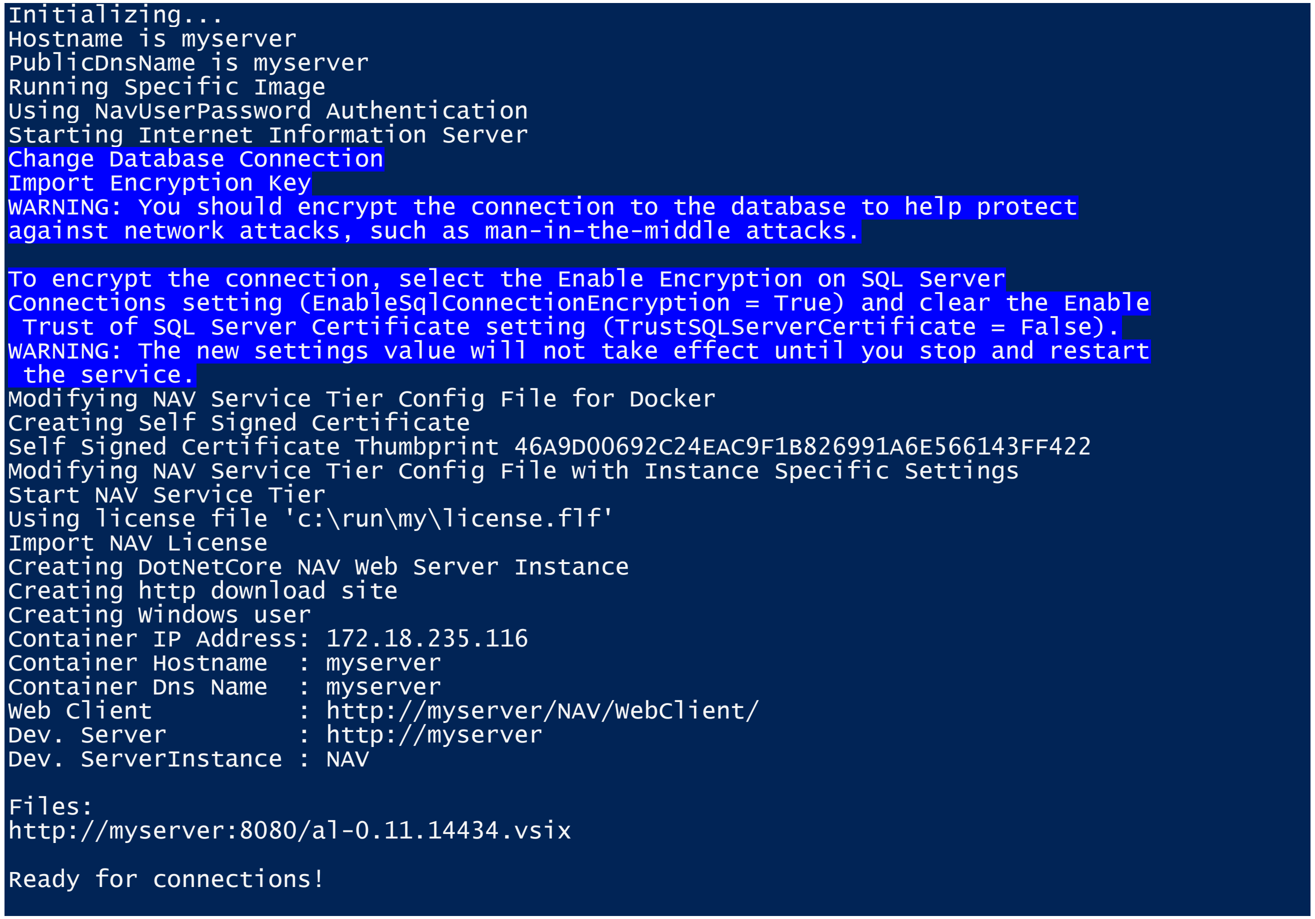
To inspect the log, write:

docker logs myserver

and you will see that the myserver container never starts the local SQL Server, instead it changes the database connection.

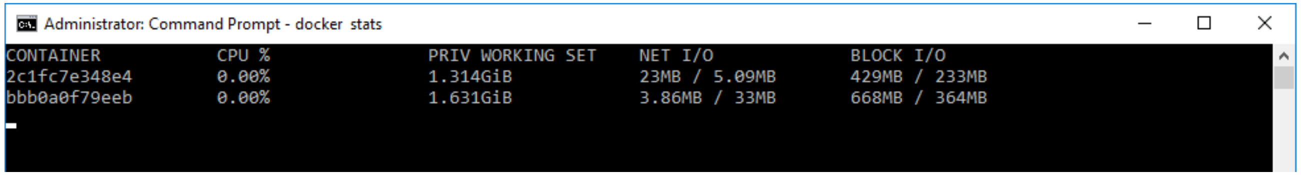
Try to connect to the navserver Web client and the myserver Web Client (on the Desktop) at the same time and you will see, that they are using the same database.

The highlighted section on the right are all outputs, which is a result of the script override.



```
Initializing...
Hostname is myserver
PublicDnsName is myserver
Running Specific Image
Using NavUserPassword Authentication
Starting Internet Information Server
Change Database Connection
Import Encryption Key
WARNING: You should encrypt the connection to the database to help protect
against network attacks, such as man-in-the-middle attacks.

To encrypt the connection, select the Enable Encryption on SQL Server
Connections setting (EnableSqlConnectionEncryption = True) and clear the Enable
 Trust of SQL Server Certificate setting (TrustSQLServerCertificate = False).
WARNING: The new settings value will not take effect until you stop and restart
 the service.
Modifying NAV Service Tier Config File for Docker
Creating Self Signed Certificate
Self Signed Certificate Thumbprint 46A9D00692C24EAC9F1B826991A6E566143FF422
Modifying NAV Service Tier Config File with Instance Specific Settings
Start NAV Service Tier
Using license file 'c:\run\my\license.flf'
Import NAV License
Creating DotNetCore NAV Web Server Instance
Creating http download site
Creating Windows user
Container IP Address: 172.18.235.116
Container Hostname  : myserver
Container Dns Name  : myserver
Web Client          : http://myserver/NAV/WebClient/
Dev. Server         : http://myserver
Dev. ServerInstance : NAV

Files:
http://myserver:8080/al-0.11.14434.vsix

Ready for connections!
```

Running **docker stats** now reveals two containers and the one running SQL Server and NAV uses more memory than the one running NAV only.



```
CONTAINER           CPU %               PRIV WORKING SET    NET I/O             BLOCK I/O
2c1fc7e348e4        0.00%               1.314GiB            23MB / 5.09MB       429MB / 233MB
bbb0a0f79eeb        0.00%               1.631GiB            3.86MB / 33MB       668MB / 364MB
```

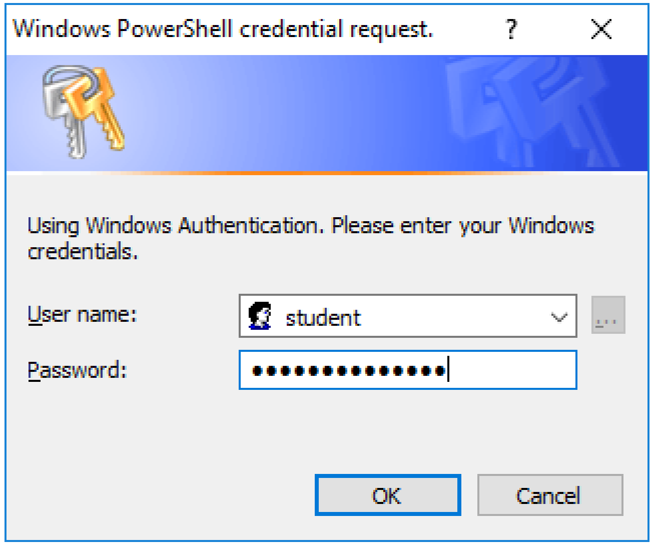
Using the Object Handling Functions in navcontainerhelper

Open the Nav Container Helper prompt and run

New-NavContainer -accept_eula -containerName myserver -includeCSide

To create a CSide development environment next to the navserver container.

Use your Workshop VM credentials when asked for credentials.



Windows PowerShell credential request.

Using Windows Authentication. Please enter your Windows credentials.

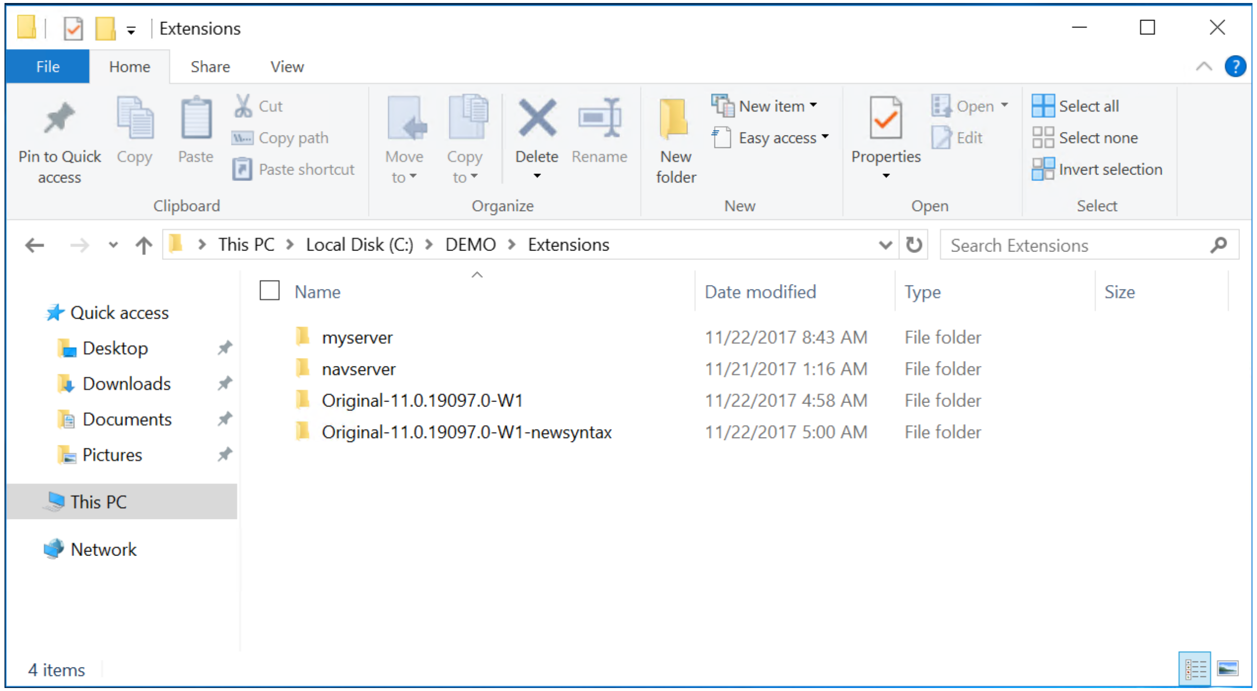
User name: student

Password: ••••••••••••••

Navigate to the C:\DEMO\Extensions folder and examine the folders:

myserver is a folder with files specific for the myserver container
navserver is a folder with files specific for the navserver container
Original-11.0.19097.0-W1 contains all the base objects for build 11.0.19097.0 (w1 version)
Original-11.0.19097.0-W1-newsyntax contains all the base objects for build 11.0.19097.0 (w1 version) in new syntax format (for txt2al)

The reason for these base object folders are for being able to create deltas from changes in a container.



Extensions

This PC › Local Disk (C:) › DEMO › Extensions

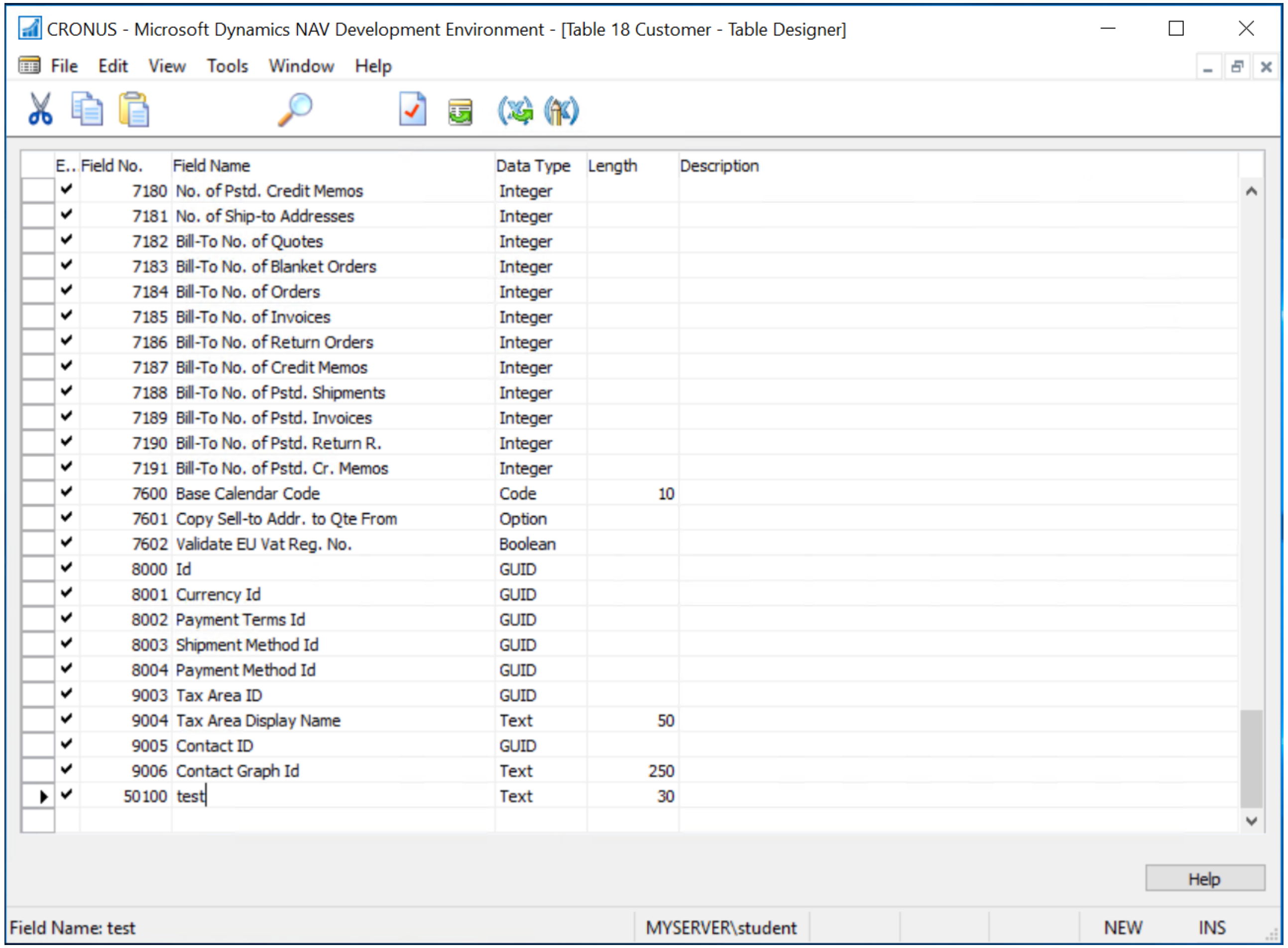
Name	Date modified	Type	Size
myserver	11/22/2017 8:43 AM	File folder	
navserver	11/21/2017 1:16 AM	File folder	
Original-11.0.19097.0-W1	11/22/2017 4:58 AM	File folder	
Original-11.0.19097.0-W1-newsyntax	11/22/2017 5:00 AM	File folder	

4 items

Start the myserver CSIDE client and modify a few objects.

Please only create modifications which are allowed in extensions v1.

Save your modifications and close the classic development environment.

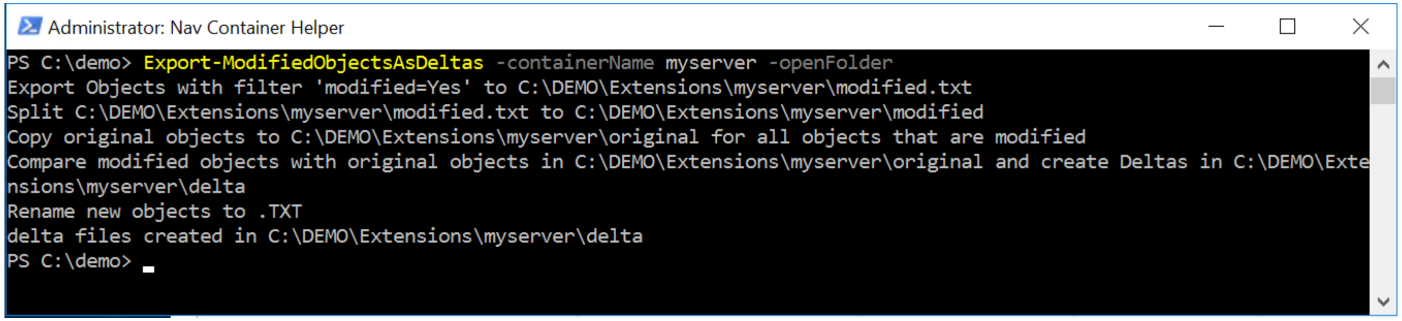

CRONUS - Microsoft Dynamics NAV Development Environment - [Table 18 Customer - Table Designer]
File
Edit
View
Tools
Window
Help
Help

E..	Field No.	Field Name	Data Type	Length	Description
✓	7180	No. of Pstd. Credit Memos	Integer		
✓	7181	No. of Ship-to Addresses	Integer		
✓	7182	Bill-To No. of Quotes	Integer		
✓	7183	Bill-To No. of Blanket Orders	Integer		
✓	7184	Bill-To No. of Orders	Integer		
✓	7185	Bill-To No. of Invoices	Integer		
✓	7186	Bill-To No. of Return Orders	Integer		
✓	7187	Bill-To No. of Credit Memos	Integer		
✓	7188	Bill-To No. of Pstd. Shipments	Integer		
✓	7189	Bill-To No. of Pstd. Invoices	Integer		
✓	7190	Bill-To No. of Pstd. Return R.	Integer		
✓	7191	Bill-To No. of Pstd. Cr. Memos	Integer		
✓	7600	Base Calendar Code	Code	10	
✓	7601	Copy Sell-to Addr. to Qte From	Option		
✓	7602	Validate EU Vat Reg. No.	Boolean		
✓	8000	Id	GUID		
✓	8001	Currency Id	GUID		
✓	8002	Payment Terms Id	GUID		
✓	8003	Shipment Method Id	GUID		
✓	8004	Payment Method Id	GUID		
✓	9003	Tax Area ID	GUID		
✓	9004	Tax Area Display Name	Text	50	
✓	9005	Contact ID	GUID		
✓	9006	Contact Graph Id	Text	250	
✓	50100	test	Text	30	

Field Name: test | MYSERVER\student | NEW | INS

Run

Export-ModifiedObjectsAsDeltas -containerName myserver -openfolder

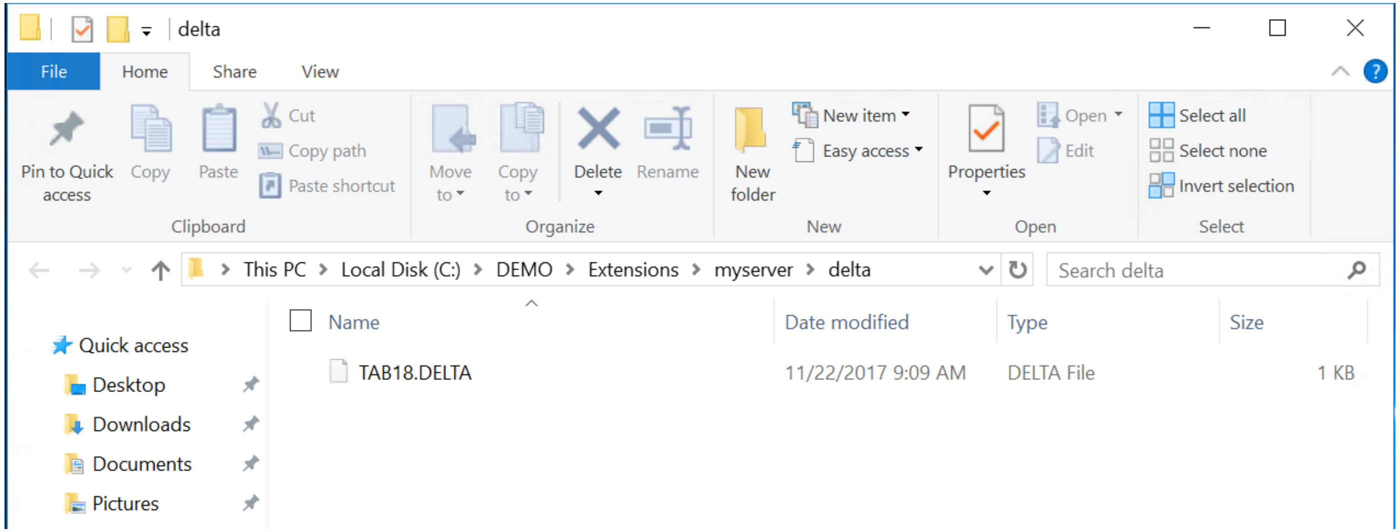


```
PS C:\demo> Export-ModifiedObjectsAsDeltas -containerName myserver -openFolder
Export Objects with filter 'modified=Yes' to C:\DEMO\Extensions\myserver\modified.txt
Split C:\DEMO\Extensions\myserver\modified.txt to C:\DEMO\Extensions\myserver\modified
Copy original objects to C:\DEMO\Extensions\myserver\original for all objects that are modified
Compare modified objects with original objects in C:\DEMO\Extensions\myserver\original and create Deltas in C:\DEMO\Exte
nsions\myserver\delta
Rename new objects to .TXT
delta files created in C:\DEMO\Extensions\myserver\delta
PS C:\demo> _
```

You should see a folder being opened with TXT files for new objects and DELTA files for changed.

If you navigate to the parent folder, you will find work folders for:

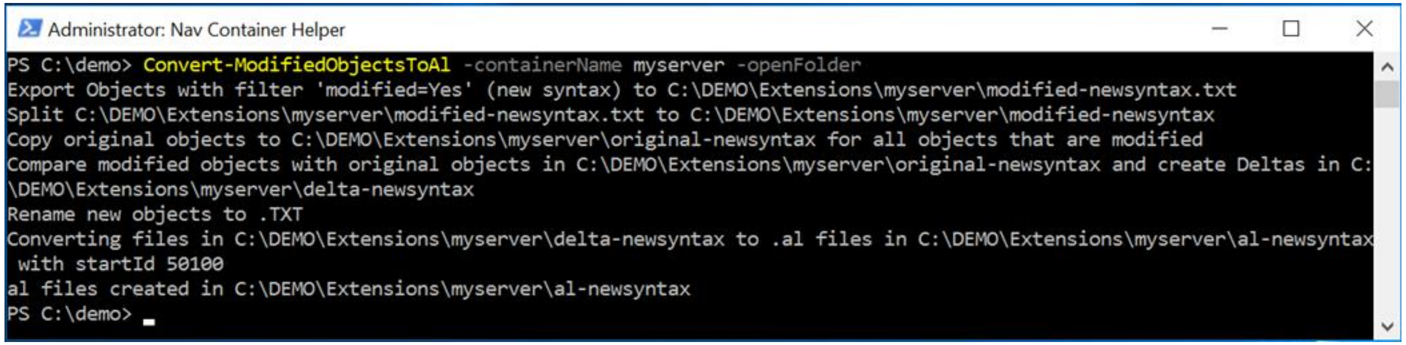
- original
- modified
- delta



Name	Date modified	Type	Size
TAB18.DELTA	11/22/2017 9:09 AM	DELTA File	1 KB

Try also

Convert-ModifiedObjectsToAl -containerName myserver -openFolder



```
PS C:\demo> Convert-ModifiedObjectsToAl -containerName myserver -openFolder
Export Objects with filter 'modified=Yes' (new syntax) to C:\DEMO\Extensions\myserver\modified-newsyntax.txt
Split C:\DEMO\Extensions\myserver\modified-newsyntax.txt to C:\DEMO\Extensions\myserver\modified-newsyntax
Copy original objects to C:\DEMO\Extensions\myserver\original-newsyntax for all objects that are modified
Compare modified objects with original objects in C:\DEMO\Extensions\myserver\original-newsyntax and create Deltas in C:
\DEMO\Extensions\myserver\delta-newsyntax
Rename new objects to .TXT
Converting files in C:\DEMO\Extensions\myserver\delta-newsyntax to .al files in C:\DEMO\Extensions\myserver\al-newsyntax
 with startId 50100
al files created in C:\DEMO\Extensions\myserver\al-newsyntax
PS C:\demo> _
```

Inspect other object handling functions, especially import and compile functions.

Portainer.io

Portainer is a free GUI for maintaining your Docker environment.

Portainer doesn’t work with IE and Edge doesn’t run on Windows Server 2016, so we need to download and install Chrome on the Workshop VM from:

https://www.google.com/intl/en/chrome/browser/

Copy the PowerShell script, paste it into PowerShell ISE and run it. The script will:

1. Reconfigure Docker deamon
2. Open port 2375 in the firewall
3. Create a Portainer directory
4. Get the IP address
5. Download and run the Portainer Docker image

```
'{
    "hosts": ["tcp://0.0.0.0:2375", "npipe://"]
}' | Set-Content "C:\ProgramData\docker\config\daemon.json"
restart-service docker

netsh advfirewall firewall add rule name="Docker" dir=in action=allow protocol=TCP localport=2375

new-item -Path "C:\Portainer" -ItemType Directory
$ipAddress = (get-netadapter | Select-Object -First 1 | get-netipaddress | ? addressfamily -eq
'IPv4').ipaddress

docker run -d -v C:\Portainer:C:\data --name portainer --hostname portainer portainer/portainer -H
tcp://${ipAddress}:2375
```

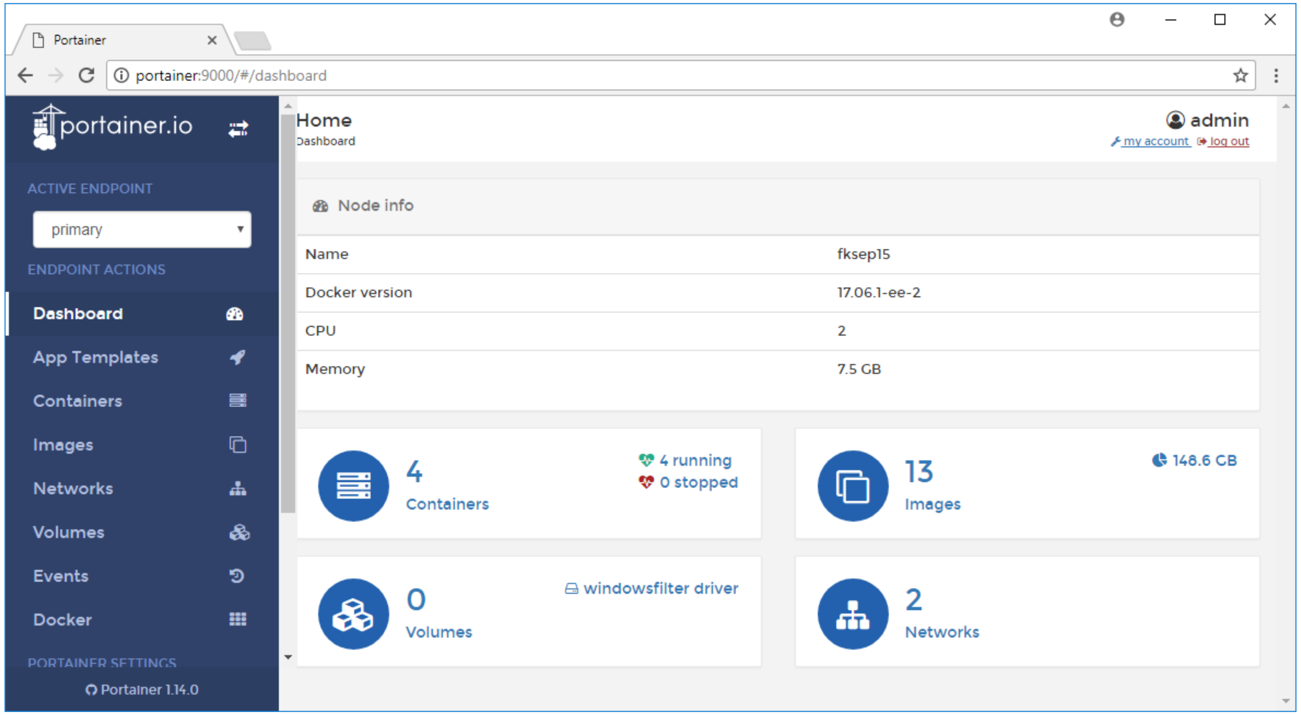
Open Google Chrome and navigate to

http://portainer:9000

On your first connection, you will have to create an admin password for Portainer.

After that…

Welcome to a free tool for maintaining your Docker environment



Node info

Name	fksep15
Docker version	17.06.1-ee-2
CPU	2
Memory	7.5 GB

4 Containers — 4 running, 0 stopped

13 Images — 148.6 GB

0 Volumes — windowsfilter driver

2 Networks