Infoma

NAV ON DOCKER

Practical insights and examples

A NAV-Skills webinar

NAV ON DOCKER
SPEAKER INTRODUCTION

- **Tobias Fenster** @TobiasFenster
  CTO at Axians Infoma
  Microsoft MVP

- **Jakub Vanak** @vanakjakub
  Software Developer at Marques Olivia

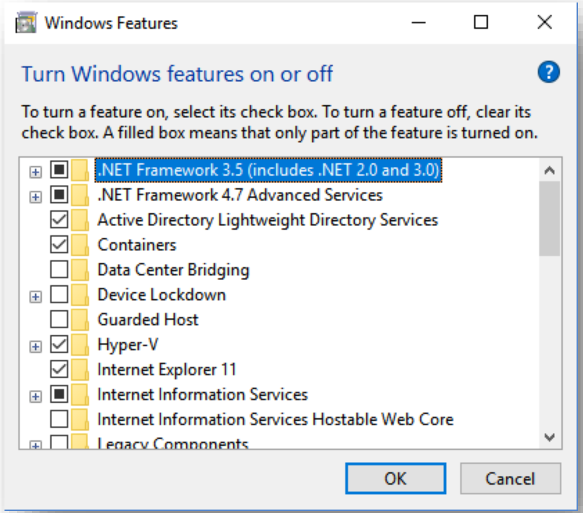
AGENDA

- How to get up and running
- Database options and handling
- Customize the Docker image

NAV ON DOCKER - HOW TO GET UP AND RUNNING
OS AND PREREQUISITES

Infoma

- Windows Server 2016
  - Activate Feature: Containers
- Windows 10
  - Professional / Enterprise edition
  - Activate Features: Hyper-V, Containers
- Windows Server 1709
  - Activate Features: Hyper-V, Containers
  - No benefits over WS2016 right now as the official NAV images are based on the WS2016 core image.


Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ■ .NET Framework 4.7 Advanced Services
- ☑ Active Directory Lightweight Directory Services
- ☑ Containers
- ☐ Data Center Bridging
- ☐ Device Lockdown
- ☐ Guarded Host
- ☑ Hyper-V
- ☑ Internet Explorer 11
- ■ Internet Information Services
- ☐ Internet Information Services Hostable Web Core
- ☐ Legacy Components

NAV ON DOCKER - HOW TO GET UP AND RUNNING
INSTALL DOCKER

- Official Docker pages:
  - https://docs.docker.com/engine/installation/windows/docker-ee/#install-docker-ee

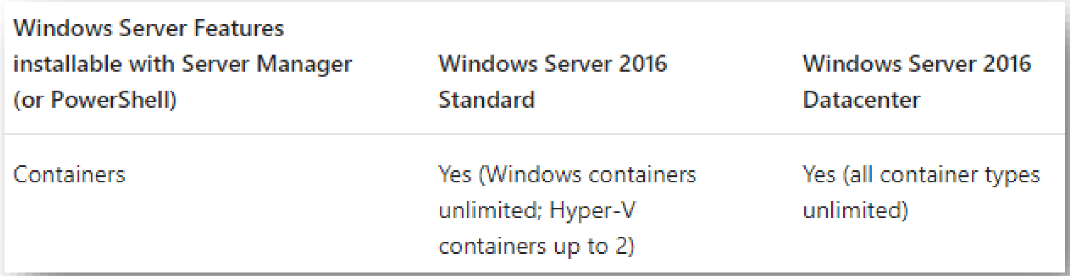
```
Install-Module DockerProvider -Force

Install-Package Docker -ProviderName DockerProvider -Force
```

- Jakub's script:
  - https://gist.github.com/Koubek/1831c2aba7f558de4b1461476105ba85
  - Install/Upgrade Docker EE (Windows Server).
  - Test and install prerequisites, show available version and eventually start the installation
  - Also included in Waldo's script library: /NAV Docker/01_InstallDockerEE.ps1
- Recommended: navcontainerhelper as explained on Freddy's blog

NAV ON DOCKER - HOW TO GET UP AND RUNNING
OS IMPACT ON DOCKER EDITIONS

- Windows Server
  - Docker EE (licensed within Windows Server license)
  - Process Isolation (by default)
  - Hyper-V Isolation (can be used when needed)
- Windows 10
  - Docker CE
  - Hyper-V Isolation only
  - Memory “weirdness”



Windows Server Features installable with Server Manager (or PowerShell)	Windows Server 2016 Standard	Windows Server 2016 Datacenter
Containers	Yes (Windows containers unlimited; Hyper-V containers up to 2)	Yes (all container types unlimited)

Source: https://docs.microsoft.com/en-us/windows-server/get-started/2016-edition-comparison

NAV ON DOCKER - HOW TO GET UP AND RUNNING
GET THE RIGHT IMAGE

Infoma

- Command to obtain a docker image: docker pull microsoft/dynamics-nav:[tag]
- [tag] specifies a version, CU and language as [version[-cu][-country]] where
  - version = 2016, 2017 or 2018 (omit to get the latest)
  - cu = rtm, cu1, cu2, cu3, … (omit to get the lates)
  - country = dk, dek, fr, gb, na, … (omit to get w1)
- Examples:

  docker pull microsoft/dynamics-nav:2017 => latest CU of 2017 in W1 localization

  docker pull microsoft/dynamics-nav:2016-cu18 => 2016 CU18 in W1 localization

  docker pull microsoft/dynamics-nav:2018-dk => latest cu of 2018 in dk localization

NAV ON DOCKER - HOW TO GET UP AND RUNNING
START A CONTAINER

- Standard docker command:

```
docker run -e accept_eula=Y microsoft/dynamics-nav:2018-rtm
```

- navcontainerhelper command:

```
New-NavContainer -containerName test -accept_eula
```

- Windows 10 / Hyper-V isolation specific:

```
docker run –m 3G -e accept_eula=Y microsoft/dynamics-nav
```

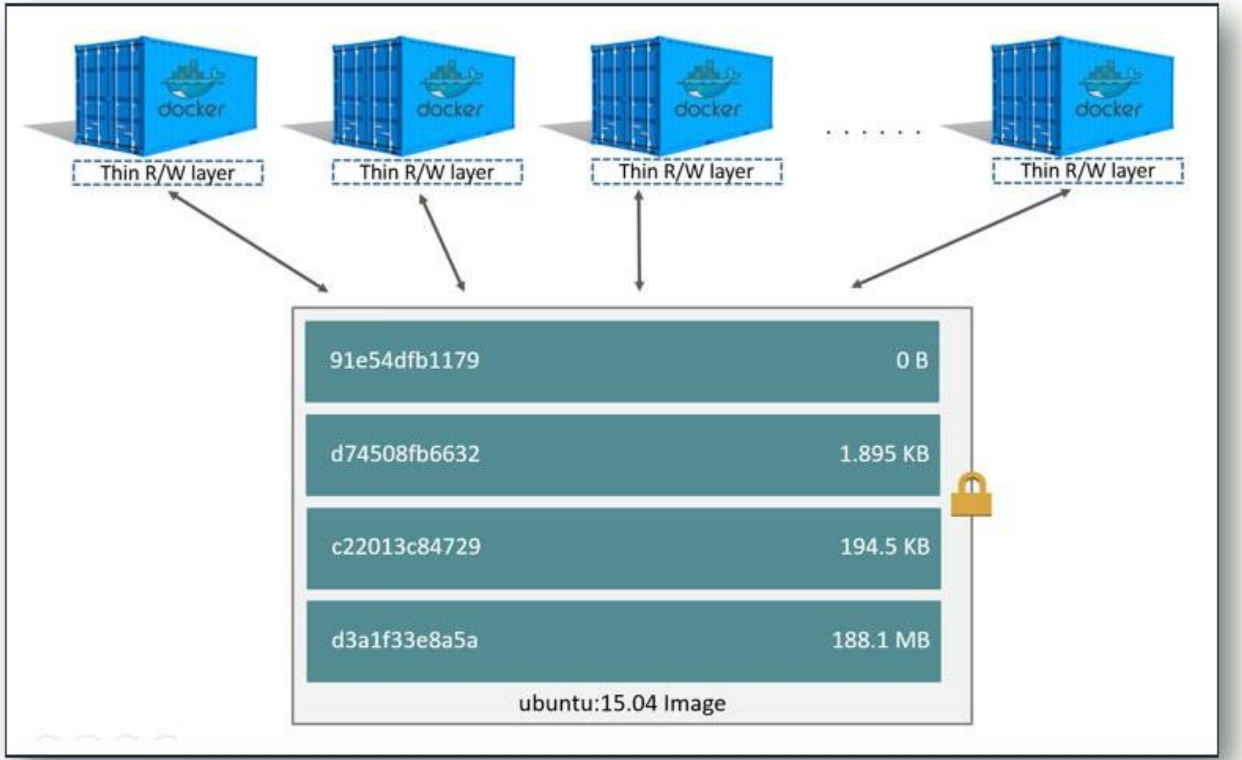
  (this actually reserves 3G of memory even if NAV only needs 1G)

DEMO: START YOUR FIRST NAV CONTAINER

NAV ON DOCKER – DATABASE HANDLING
DOCKER CONTAINERS AND DATA

- Layering in Docker containers:
  - Image layers (readonly) + one container layer (read/write)
  - All write operations are being made within the top layer (the container layer).
  - When you modify an existing file, Docker checks if it exists in the container layer
    - Yes → modify this file.
    - No → drill down through all layers from the top one until the first version of the file is reached. Copy this file into the container layer and use it.
- Volumes
  - Map a host folder to a folder inside the container which leads to the same performance as directly on the host and results in persistency even if the container is removed (and the container layer with it)

NAV ON DOCKER – DATABASE HANDLING
DOCKER CONTAINERS AND DATA



Source: https://docs.docker.com/engine/userguide/storagedriver/imagesandcontainers

NAV ON DOCKER – DATABASE HANDLING BASIC CONSIDERATIONS AND SCENARIOS

- Is the database tightly coupled with exclusively the one NAV server instance inside the container and can easily be recreated, e.g. in a demo / sales environment and doesn‘t rely too much on performance → Scenario 1) db in container without a volume
- Same as before, but you need better performance and the database needs to survive a re-creation of the container, e.g. for a CU update → Scenario 2) db in container with a volume
- Will you have multiple NAV server instances connected to the database or you need to fully tweak performance → Scenario 3) db in a central SQL server

NAV ON DOCKER – DATABASE HANDLING SCENARIO 1) DB IN CONTAINER WITHOUT A VOLUME

- Data is stored within the container layer.
- Data is being removed when the container is being removed, but not on restart
- Performance is lower, this especially hurts where disk I/O performance is crucial.
- PRO:
  - Easiest way how to spin a container (no volume dependency).
  - Ideal for demos, simple testing or workshops/trainings.
- CON:
  - The worst performance compared with the following use-cases.
  - NAV images are using SQL Express (DB size = max. 10 GB!).
  - 1 SQL instance per container.

NAV ON DOCKER – DATABASE HANDLING SCENARIO 2) DB IN CONTAINER WITH A VOLUME

- Data is mapped between the container and the host system and will never be removed
- Performance should be practically the same as running the database on the host HDD.
- PRO:
    - Still a very easy way to spin a container (slightly more complicated than the previous one).
    - Performs better compared to the previous one.
    - Ideal for demos, simple testing or workshops/trainings if you want / need more performance and persistent data even if you remove the container
- CONS:
    - NAV images are using SQL Express (DB size = max. 10 GB!).
    - 1 SQL instance per container.

DEMO: USE A VOLUME FOR THE DATABASE

NAV ON DOCKER – DATABASE HANDLING SCENARIO 3) DB OUTSIDE THE CONTAINER

- Very likely the best performance, definitely the most flexibility:
  - SQL as a traditional service on the Docker host / on a remote system
  - SQL running as a container on the Docker host / on a remote system
  - We can choose edition (Express / Developer / Standard / Enterprise)
- PRO:
  - Flexibility: Do whatever you want, no limitation by NAV on Docker (even SQL on Linux ;))
  - The way to go for performance optimization
  - Minimal overhead → 1 SQL instance for all containers.
- CONS:
  - Harder to configure (although it got a lot easier).

NAV ON DOCKER – AUTHORIZATION DATABASE SERVER

- Using SQL Server inside the NAV container → nothing to do for NAV, use sa for C/SIDE
- Using SQL Server outside the NAV container
  - Use SQL authentication by adding parameters to docker run / navcontainerhelper
  - Use Windows authentication by using group Managed Service Accounts
  - C/SIDE works the same

NAV ON DOCKER – AUTHORIZATION
NAV SERVER (WINDOWS / WEB CLIENT / VS CODE)

Infoma

- Just start it and the scripts inside the container will create a user admin with a generated password for NavUserPassword auth (if it doesn‘t exist)
- Give it a specific username and password without configuring Windows auth works the same
- Give it your Windows user and password and configure Windows auth and it will set everything up so that your user works with Windows SSO
- Use gMSA to make Windows auth work for every Windows login in the database

Infoma

DEMO: USE GMSA WITH AN EXTERNAL DATABASE

NAV ON DOCKER – CUSTOMIZE THE IMAGE SCENARIOS

- Warning: The scenarios on this slide work with the scripts on Github but will take some days until they work in the images (will have to be rebuilt)
- Download a .zip and put it anywhere in the container e.g. to add dlls or override the scripts by adding e.g.

  ```
  -e folders='c:\run\my=https://myserver/myscripts.zip\scripts;
  c:\program files\microsoft dynamics nav\110\service\add-
  ins=https://myserver/myscripts.zip\add-ins’
  ```
- Change settings with a simple parameter like

  ```
  -e customNavSettings="EnableDebbuging=true,
  ReportPDFFontEmbedding=false"
  ```

DEMO: CONFIG CHANGES AND COPIED FILES

NAV ON DOCKER – CUSTOMIZE THE IMAGE SCENARIOS

- Use docker build to persist those changes into your own image by building it on top of another
- Use docker commit to persist changes of a stopped container into a new image

Infoma

DEMO: BUILD AN IMAGE

axians

Infoma

THANK YOU FOR YOUR ATTENTION!

Watch https://github.com/Koubek/nav-docker-examples
and https://navblog.axians-infoma.de for updates

For questions, please contact

@vanakjakub or @TobiasFenster

VINCI ENERGIES