Introduction and agenda

Tobias Fenster
CTO at Axians Infoma
Microsoft MVP for Business Applications
@tobiasfenster and
https://techblog.axians-infoma.com

Vote at ve.link/td18vote

Agenda

- Quick intro to Docker and the overall scenario
- Self-service container environments
- Multi-container environments
- Automated extension 2.0 builds with multi-stage images
- Using Azure Container Instances

Quick intro to Docker and the overall scenario

Photo by chuttersnap on Unsplash

Quick intro to docker

NAV TECH DAYS 2018
mibuso.com

What is Docker? Leading cross platform software container environment

What is a Docker container and a Docker image?

- An image is a template with the minimum amount of os, libraries and application binaries needed
- A container is an instance of an image with an immutable base and it's changes on top
- A container is NOT a VM, you especially don't have a GUI and nothing you can connect to with RDP!

Quick intro to docker

NAV TECH DAYS 2018
mibuso.com

What is a Docker host? The (physical or virtual) machine where the containers are running

What is a Docker registry? A place where you and others can upload (push) and download (pull) images

Why Docker?

- Easy way to create deployments / configuration in a very stable and reliable way (no "works here", helps a lot to avoid gaps between dev and ops)
- Better resource usage than in VMs, especially because there is no guest os as the host kernel is directly used
- Big ecosystem of readily available images, primarily on Docker Hub

Introduction to the Infoma scenario

Axians Infoma is an ISV for > 1.200 customers with > 100 employees directly working on the product newsystem (program managers, developers, back office etc.)

Central team provides infrastructure:

- Standard images for laptops, central VMs for development, central SQL Servers / NST / IIS for dev and test
- Local NST installs for some cases but more because of how NAV currently works than because we like it: debugging, need to restart, cmdlets that work only locally, development of server-side dlls

Technical infrastructure must be useable as quick and easy as possible: standardize, minimize friction, don't expect infrastructure knowledge and don't create a need for it

- A lot of employees with infrastructure knowledge, but main strength and therefore focus is working on the product
- Time spent on local dev infrastructure is very likely time spent on something that won't improve the product or customer satisfaction

Introduction to the Infoma scenario

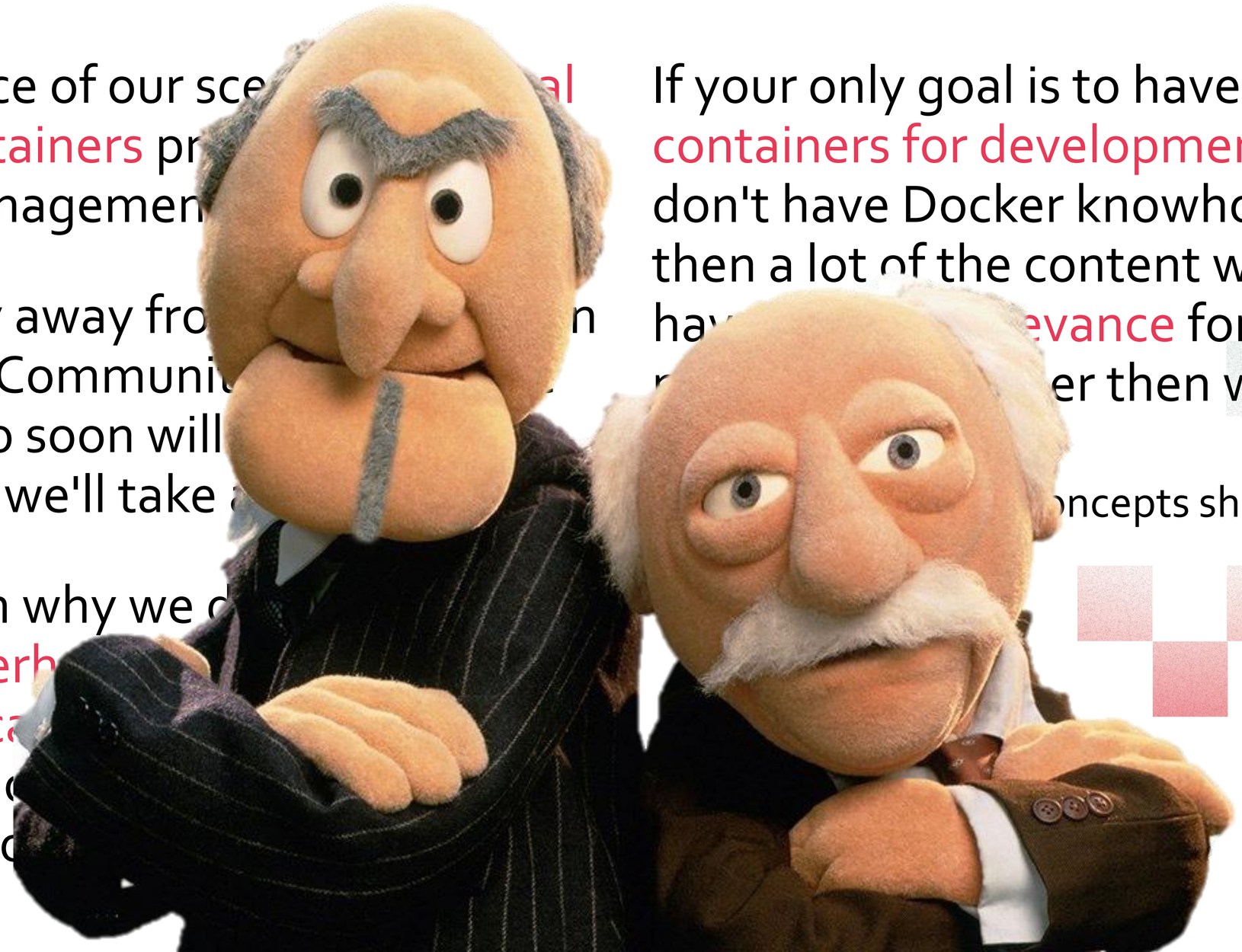
Consequence of our sce al
Docker containers pr
Release Managemen

We can stay away fro n
10 / Docker Communi
note: Win 10 soon will
isolation → we'll take

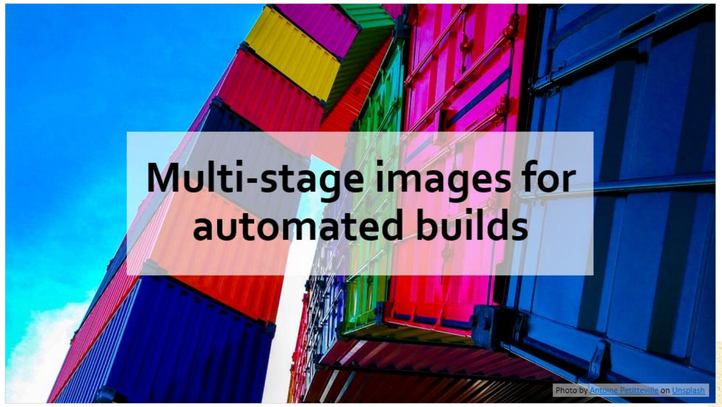
Main reason why we d
navcontainerh
assumes loca
ops professi
advanced Do

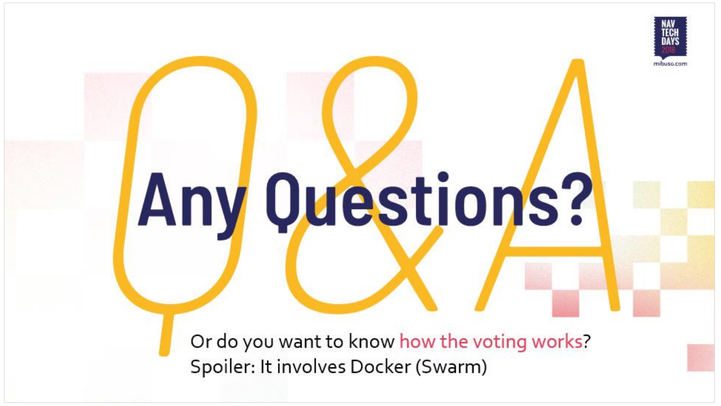
If your only goal is to have local containers for development and you don't have Docker knowhow already, then a lot of the content will only hav evance for you as
r er then would be

oncepts should still be



Your choice: ve.link/td18vote





Self-service container environments

Photo by Rye Jessen on Unsplash

Self-service container environments

Why? Easy access to releases

1-3 major releases, 4-6 bugfix releases for each country solution per year → up to 20 Infoma newsytem releases per year

Business central / NAV cumulative updates, releases and previews

All of those should be readily available for quick tests

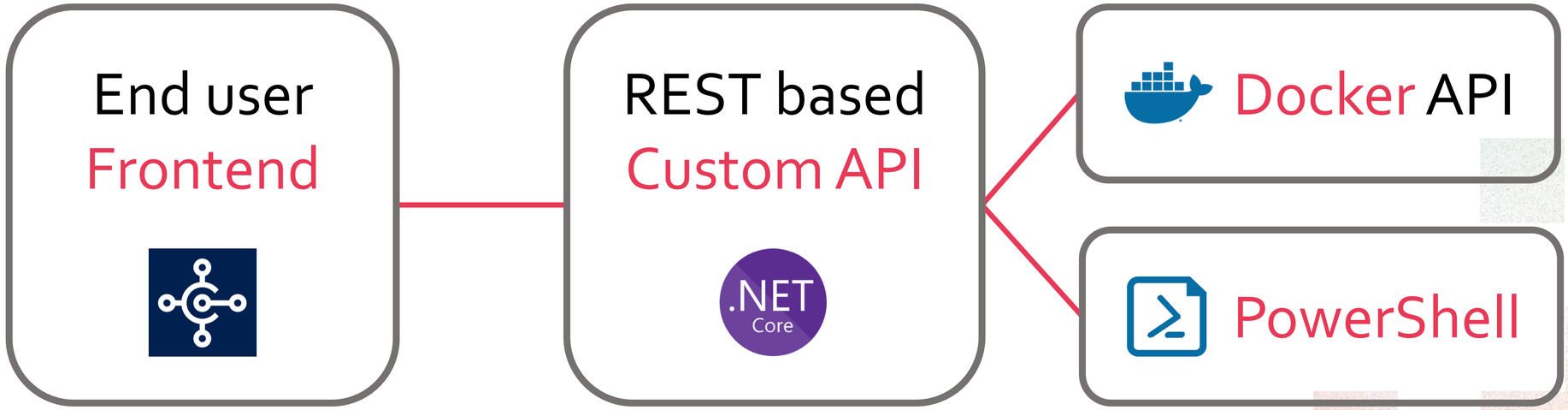
- And in the extremely unlikely case that we need to create an individual hotfix for a customer on an old release

Let's create one!

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

How?



Tools Container Host

Release Container Host

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

Frontend: BC Extension v2

- Available images maintained in a table, pulled nightly
- Containers valid for max. 3 days
- Calls proxy API through a REST interface to create or delete containers, get status and logs

Custom API: .Net Core App

- Creates gMSA (for win auth); if newsystem container then downloads DLLs from TFS and gets backup
- Constructs and executes docker run command
- Gets running containers and logs from the Docker API

Container: Standard NAV/BC image from Microsoft with a couple of additional scripts and specific settings

- Script 1: Grant an AD user group access to the database
- Script 2: Automatically convert the database on startup (in case the .bak is from an older CU than the container)
- Use Windows authentication
- Use our dev license
- Enable ClickOnce (always get the matching clients)
- Use transparent networking
- Custom NST and WebClient settings
- Set labels to identify releases

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments


```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments


```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

NAV TECH DAYS 2018
mibuso.com

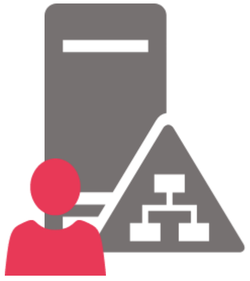
```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

Self-service container environments

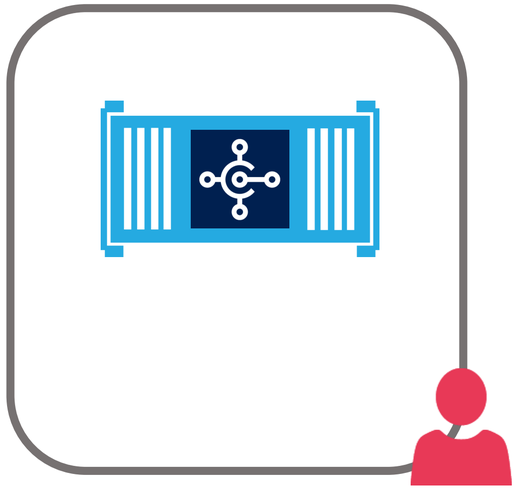
NAV TECH DAYS 2018
mibuso.com

```
docker run --name testtfe --hostname testtfe -e accept_eula=y
--network MyTransparentNetwork -e usessl=n -e clickonce=y
--security-opt "credentialspec=file://testtfe.json"
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-user-access.zip\gua,
c:\run\my=https://tools.axians-infoma.de/invoke-conversion.zip\conversion"
-v c:\nsys-freeze:c:\bkp -e bakfile=c:\bkp\newsystem_180100200.bak -v
"c:\temp\testtfe.180100200\kumulativ:C:\Program Files\Microsoft Dynamics
NAV\100\Service\Add-ins\Infoma"
-e customWebSettings="Productname=Infoma newsystem"
-e customNavSettings="SqlLongRunningThreshold=10000"
--label Owner=FUM-GLOBAL\TFENSTER --label InfomaApiGenerated=true
--label NsysRelease=180100200 --label NavRelease=100
-d microsoft/dynamics-nav:2017-cu16-de
```

What about those gMSAs?



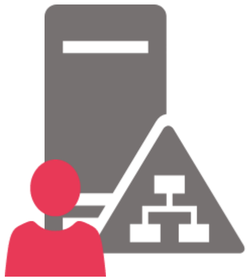
AD with a gMSA



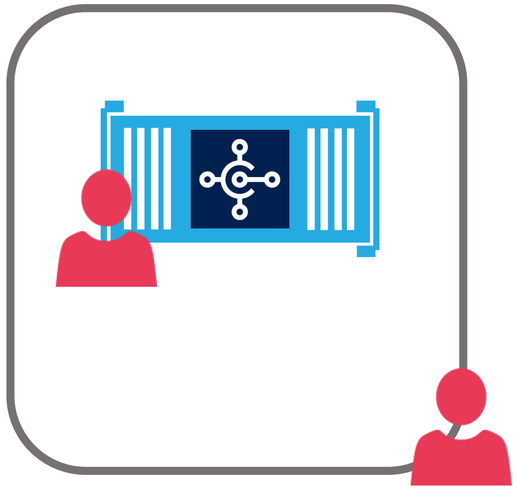
Container Host
with installed gMSA

SQL with a DB
where the gMSA
has access

What about those gMSAs?



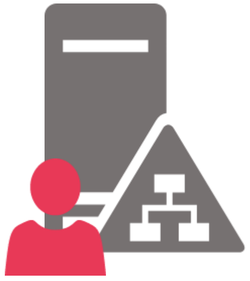
AD with a gMSA



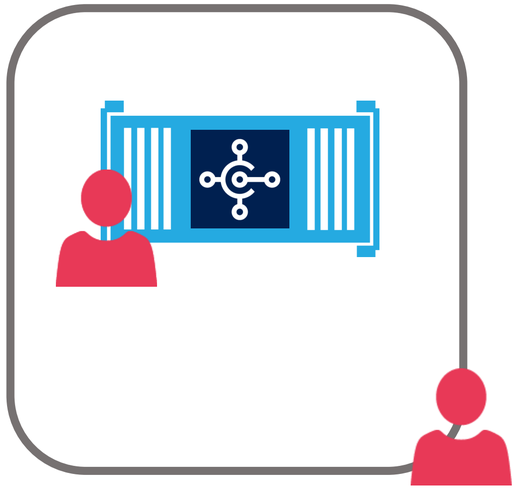
Container Host
with installed gMSA

SQL with a DB
where the gMSA
has access

What about those gMSAs?



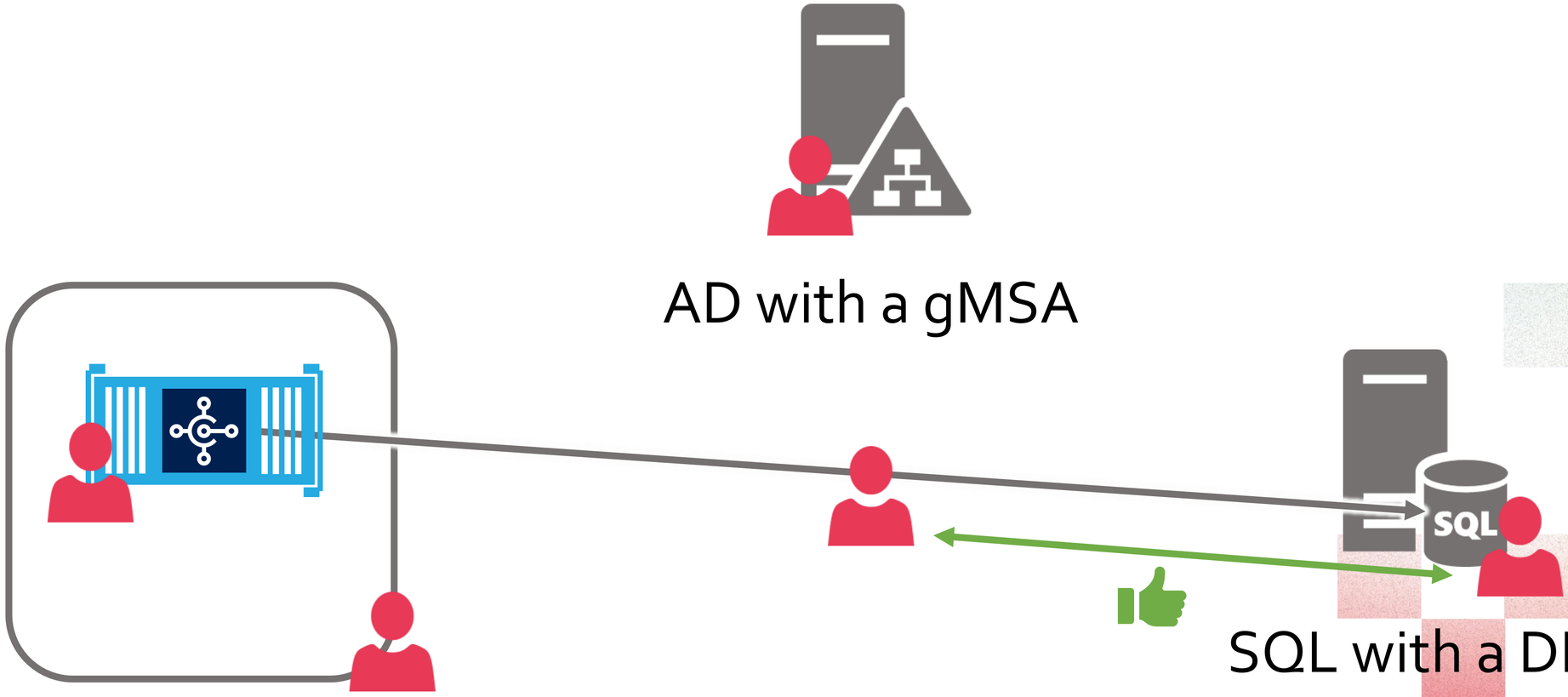
AD with a gMSA



Container Host
with installed gMSA

SQL with a DB
where the gMSA
has access

What about those gMSAs?



What about those gMSAs?

NAV TECH DAYS 2018
mibuso.com

Windows Server 2016: gMSA = container name = host name → 1 gMSA for every container, no dynamic scaling with e.g. container name_1, name_2 etc. generated on demand

Windows Server 2019: name doesn't matter

See the container up and running!

gMSA is used for outgoing connection if process in the container uses accounts Local System or Network Service

gMSAs don't have a password, can be only used on allowed machines

Container usage: Download credentialspec JSON file, add it as security opt

```
docker run --security-opt "credentialspec=file://testtfe.json" …
```

https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/manage-serviceaccounts

Multi-container environments

Multi-container environments

NAV TECH DAYS 2018
mibuso.com

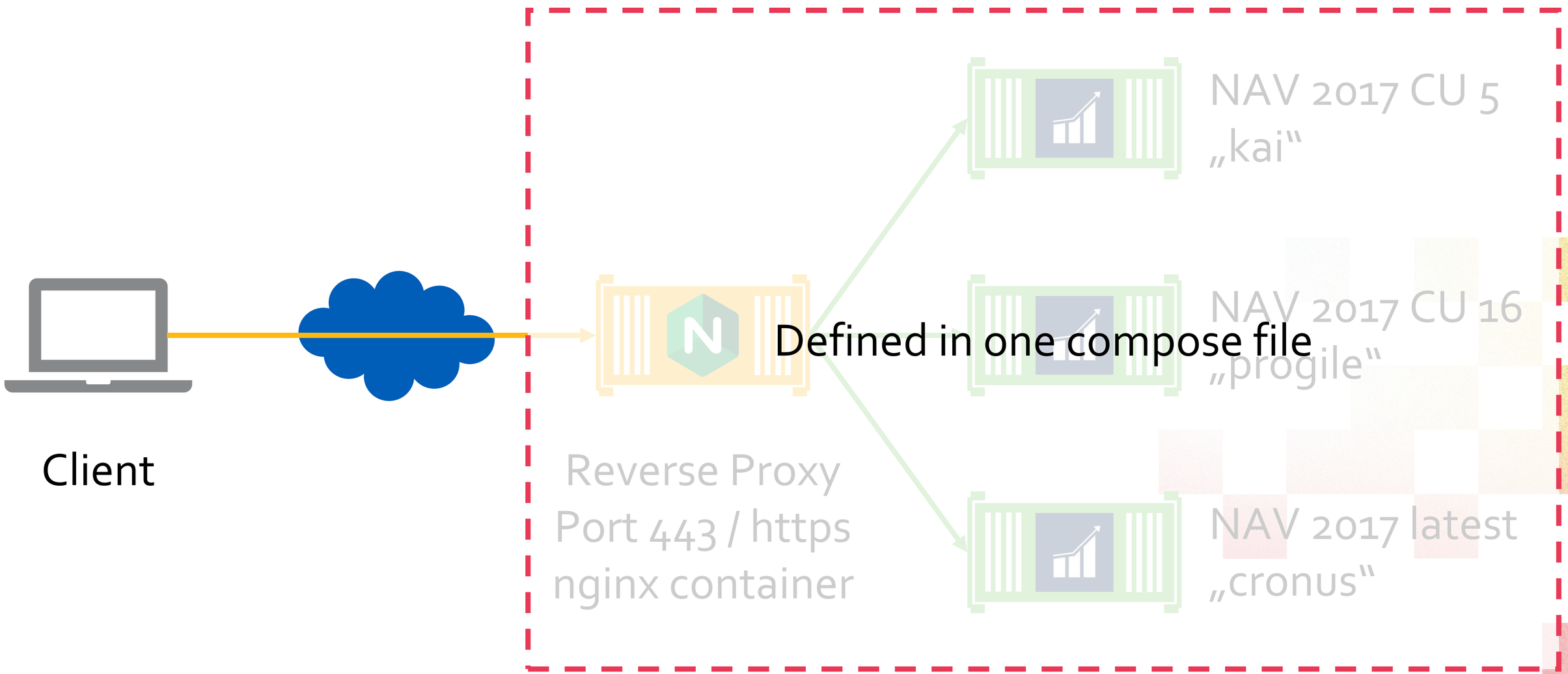
Why? Multiple very similar containers or more complex scenarios:

Release tests with 10 databases and all need the "same" container with NST / IIS → stable, reproduceable, easily adjustable config

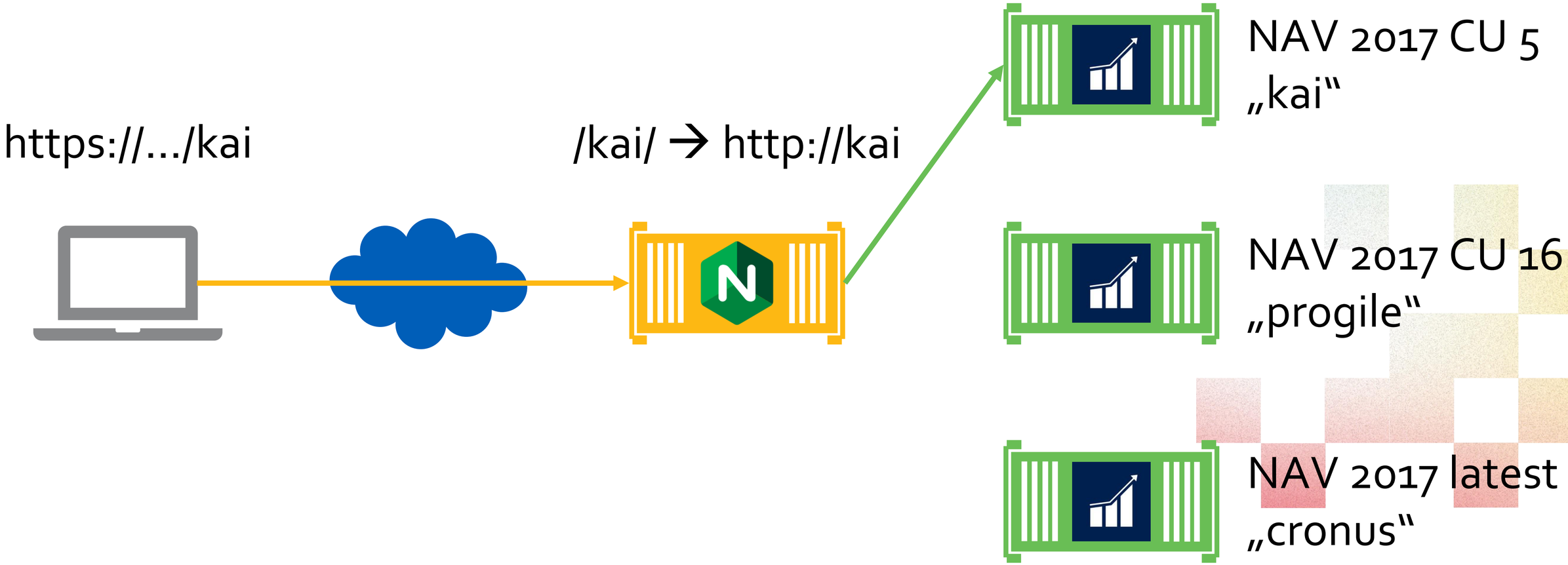
Tools host with different images → infrastructure as code

Externally available environment with multiple containers and endpoints but only a reverse proxy is exposed → multiple containers working together

Reverse proxy



Reverse proxy



Multi-container environments

How? Docker compose

Descriptive YAML for containers, their configuration and the networking setup

For the scenario with 10 identical containers: PowerShell script to generate compose file from templates

Compose is created by Docker as well, needs to be downloaded separately

Let's see the code

Multi-container environments

NAV TECH DAYS 2018
mibuso.com

YAML can be changed and Docker will only update the changed parts

Allows easily updating or even changing the host

Dynamically scalable if needed: number of replicas

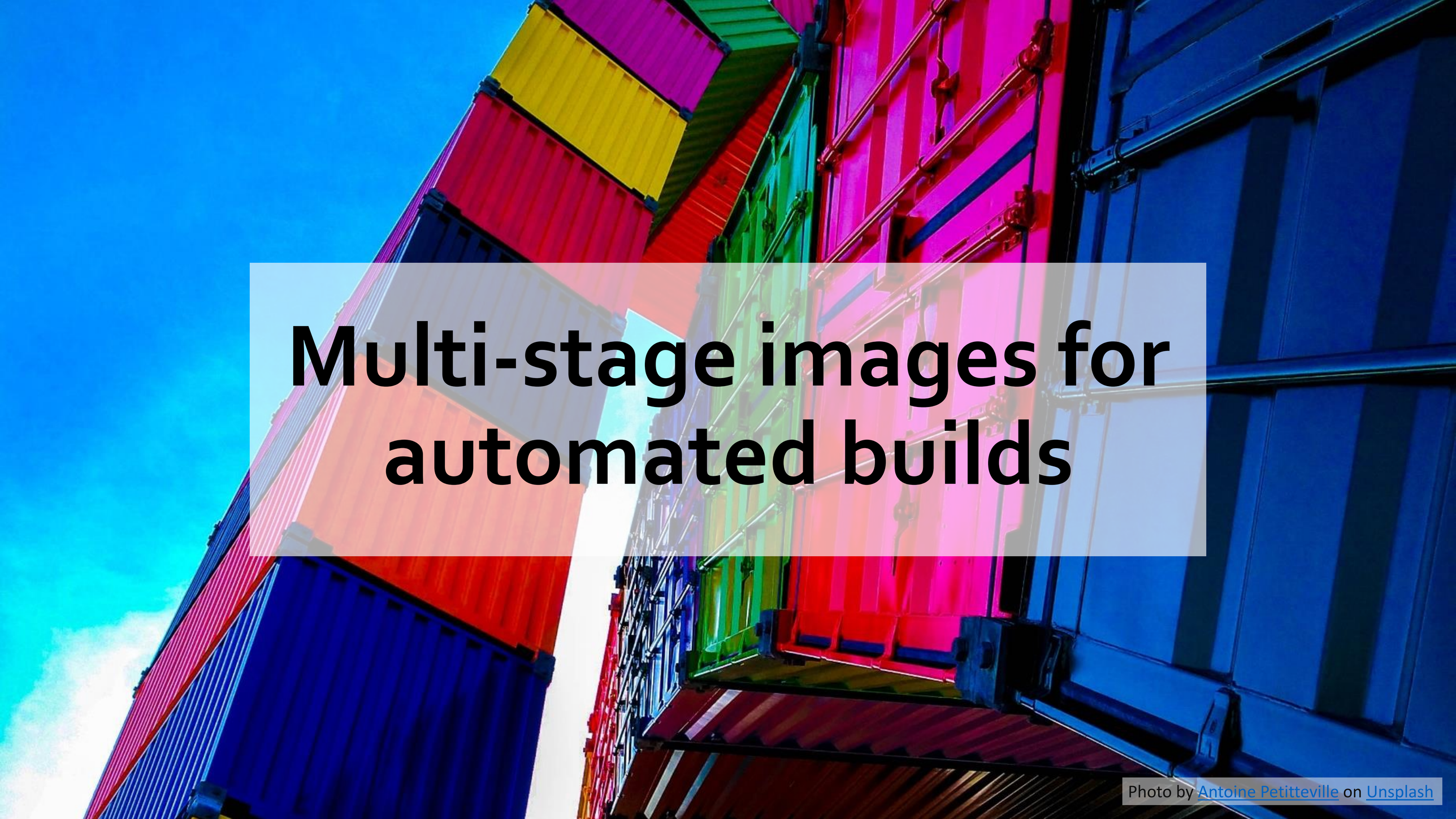
- Windows auth works fine but needs correctly named containers (gMSA)

Even more flexible alternative: Docker Swarm

- Spans multiple hosts (nodes) and places containers on the nodes on demand
- Very flexible networking from Server 2019 onwards
- Dynamic reverse proxy setup with Traefik or others (almost no setup)
- Can run mixed OS: some nodes Windows, some nodes Linux

Widely used alternative to Docker Swarm in the Linux world: Kubernetes

- Windows GA expected in the next months
- Probably Windows authentication soon after



Multi-stage images for automated builds

Photo by Antoine Petitteville on Unsplash

Automated build with multi-stage images

NAV TECH DAYS 2018
mibuso.com

Why automated builds?

- Manually building is very time consuming and error prone
- Automation allows immediate / periodic builds → fail fast and early!

Why in containers?

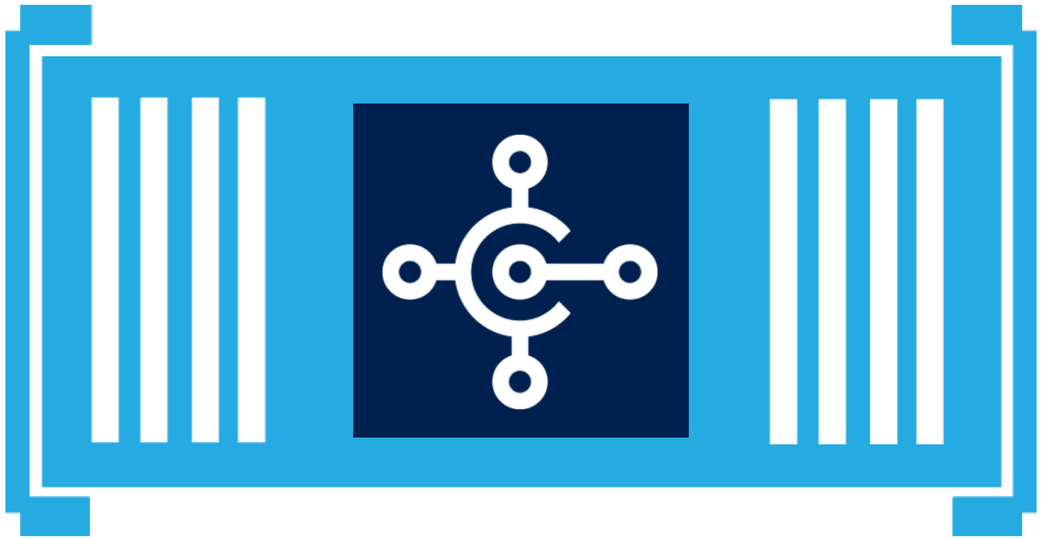
- Keeping build environments clean is difficult → often periodic re-installs if you use traditional VMs
- Creating a container for every build and every step is not much overhead but guarantees a clean, controlled and reproduceable environment (same is true for automated testing)

Why multi-stage?

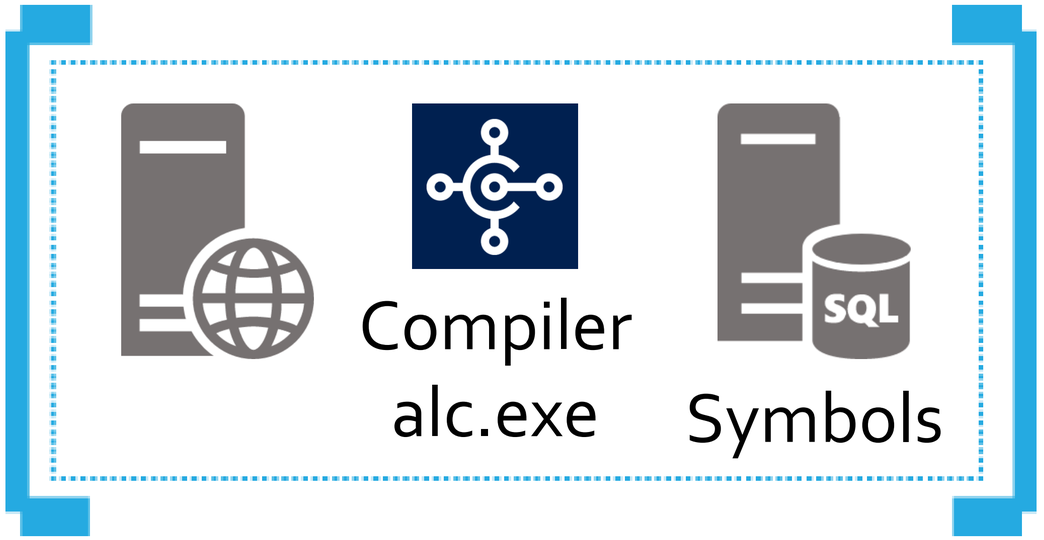
- Standard bcsandbox / bconprem is quite big and somewhat slow to start, multi-stage image reduces that → gated check-ins with quicker results

Automated build with multi-stage images

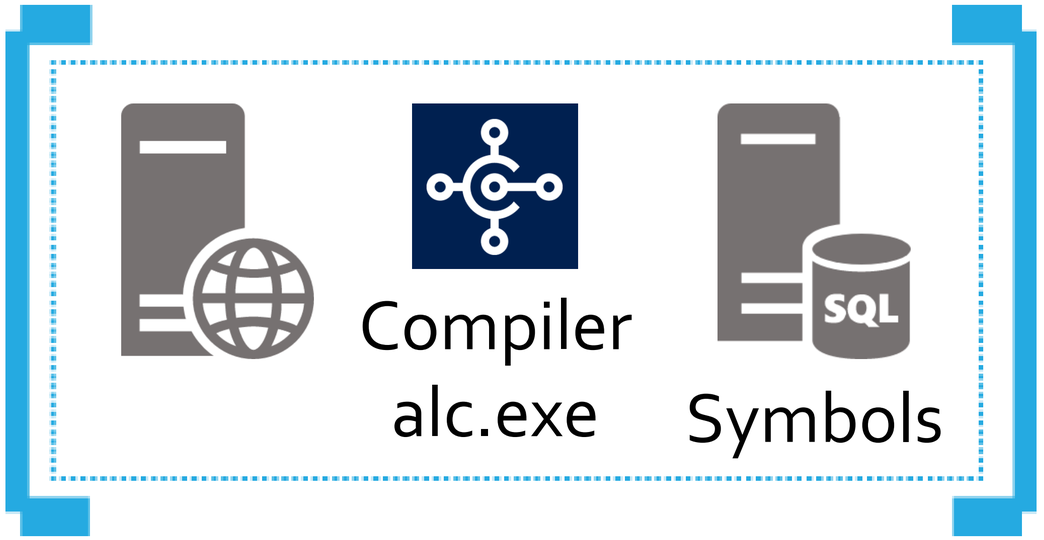
NAV TECH DAYS 2018
mibuso.com



Automated build with multi-stage images



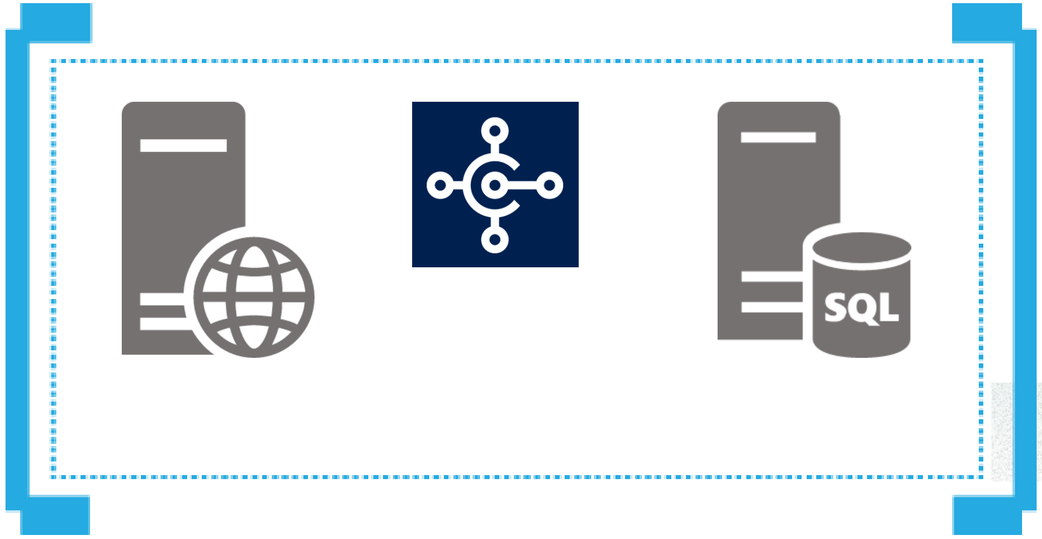
Automated build with multi-stage images



Automated build with multi-stage images

NAV TECH DAYS 2018
mibuso.com

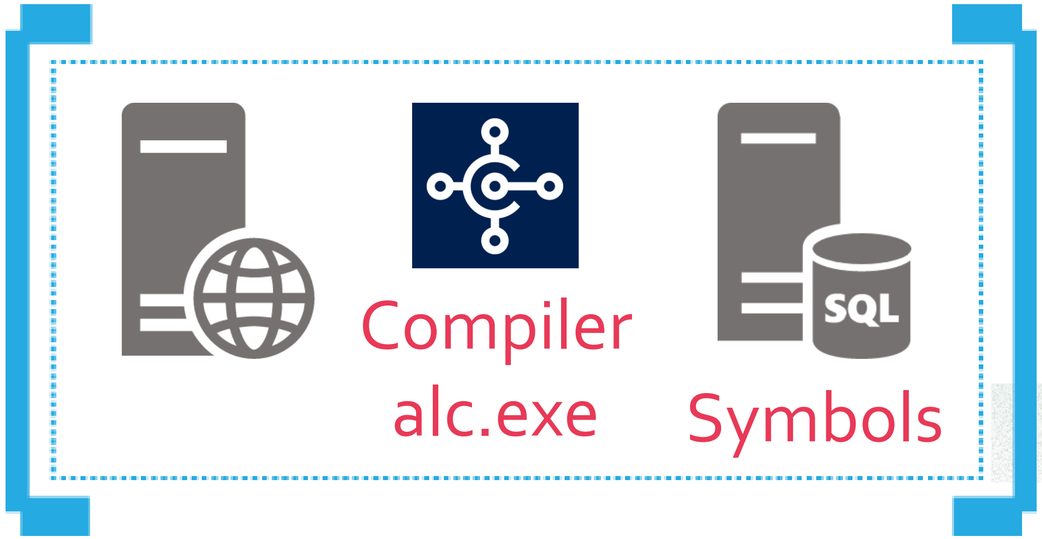
FROM bcsandbox
(windowsservercore +6G)



Automated build with multi-stage images

FROM bcsandbox
(windowsservercore +6G)

First stage: full

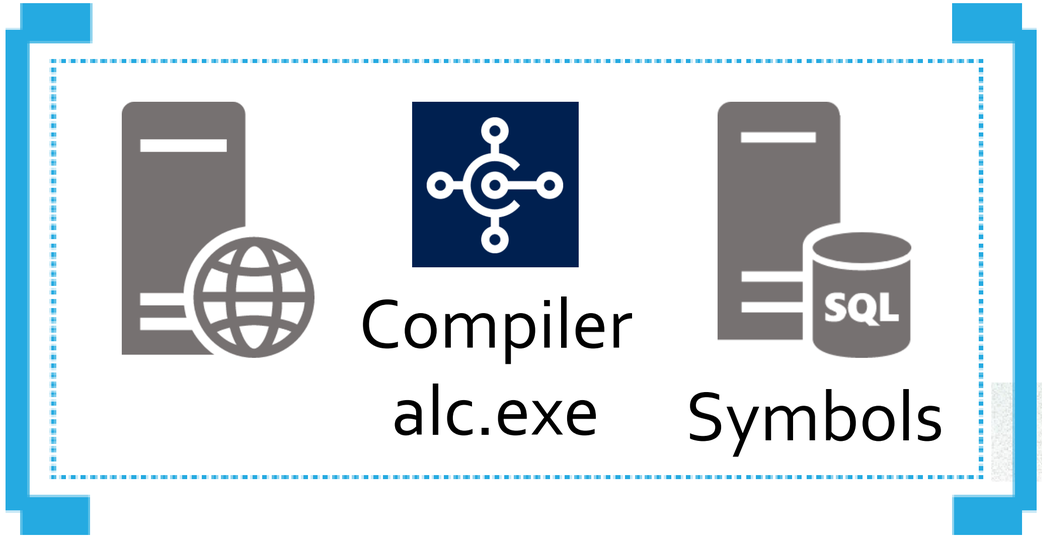


Automated build with multi-stage images

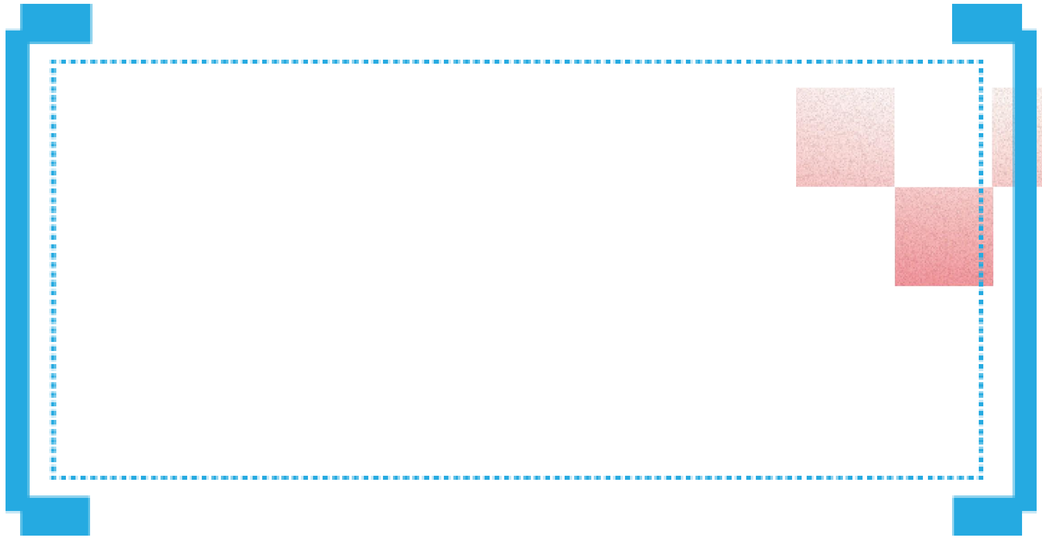
NAV TECH DAYS 2018
mibuso.com

FROM bcsandbox
(windowsservercore +6G)

First stage: full



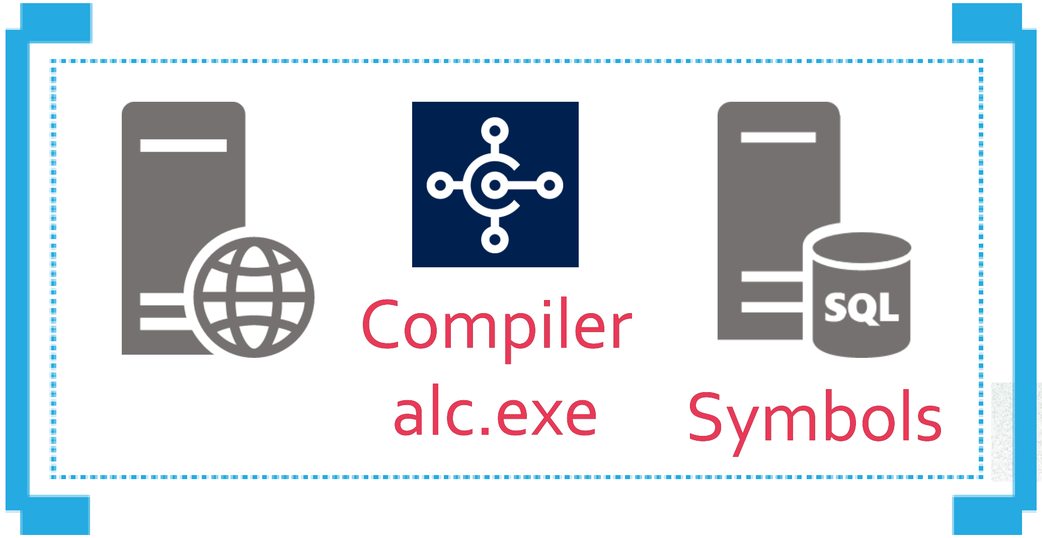
FROM windowsservercore



Automated build with multi-stage images

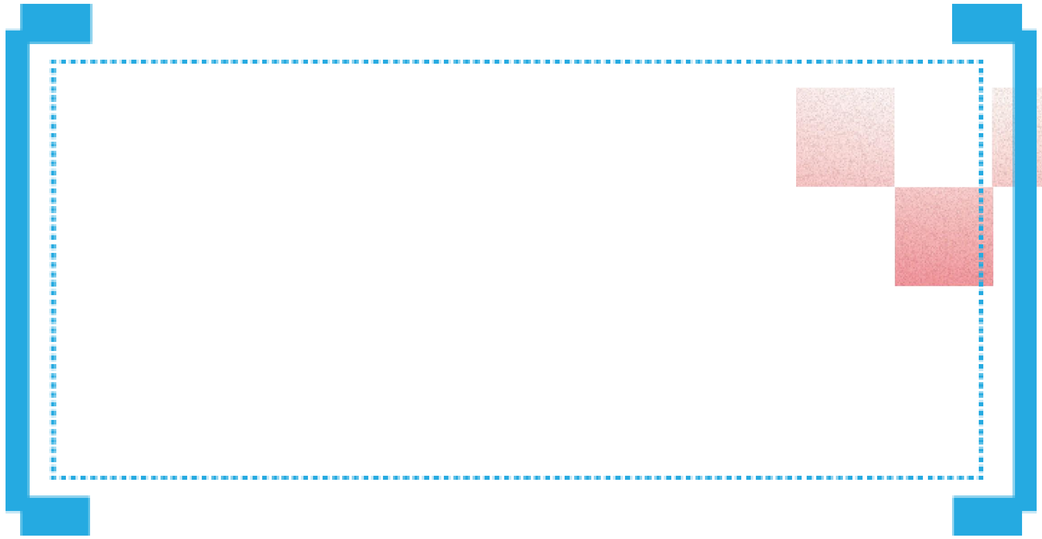
FROM bcsandbox
(windowsservercore +6G)

First stage: full



FROM windowsservercore

Second stage: build
(+300M)

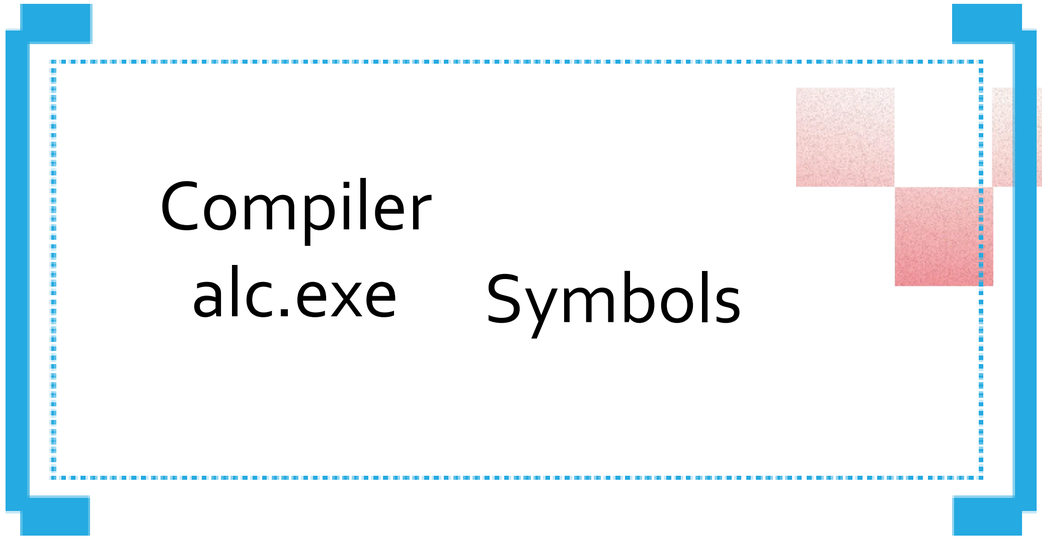


Automated build with multi-stage images

Let’s see the code

FROM windowsservercore

Second stage: build
(+300M)



Automated build with multi-stage images

Automated builds for continuous integration can be achieved in other ways as well

- AJ Kauffmann, Gunnar Gestsson and Kamil Sacek showed a way at TechDays
- Even the grumpy old guys talked about it
- Freddy Kristiansen and Stan Stempin showed a way at Directions and blogged about it
- …

Multi-stage images often used in connection with but not limited to building

- Usually it works the other way: Build image is bigger than production image
- Could also be something like
  - First stage build dlls
  - Second stage build .app
  - Third stage BC + dlls + app

Azure Container Instances

Photo by Vidar Nordli-Mathisen on Unsplash

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

Why? You quickly need 1-n business central "installations" to test or demo something or for e.g. a workshop

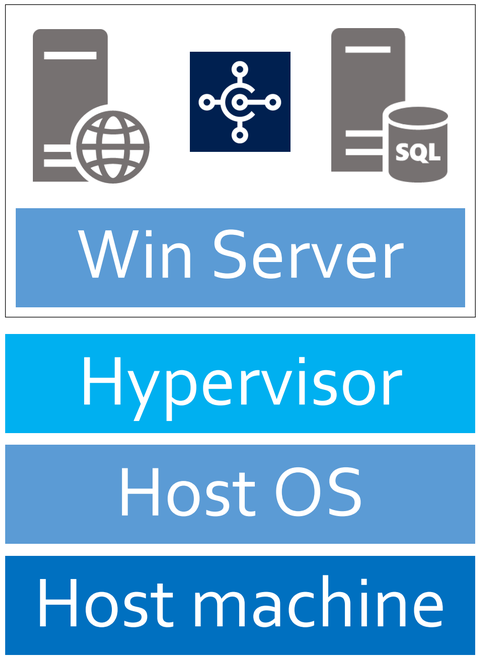
Azure Container Instances (ACIs) just run 1-n containers without the need to worry about the base infrastructure

- Start / stop / scale up / scale down as needed
- Example: 115 ACIs with 4 cores / 12 GB RAM each, started at 8:30, stopped at 13:30
  → 460 cores and 1.4TB of RAM for 5 hours with a ramp-up time of < 30 minutes

Paid on demand by seconds of CPU / RAM / Windows license (see Azure pricing calculator)

- Example above cost around 300€

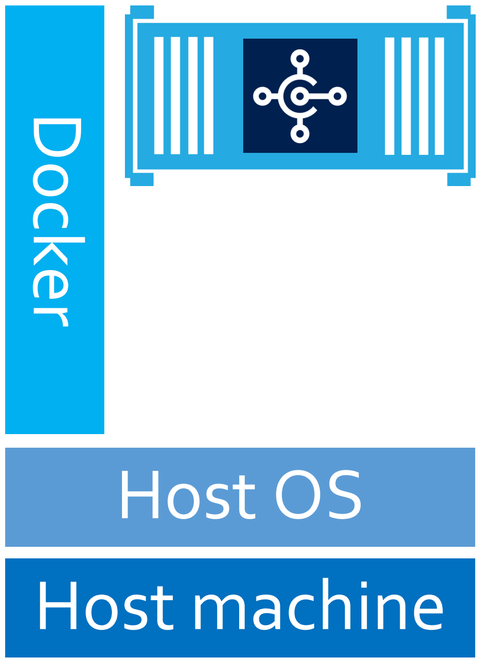
Azure Virtual Machines



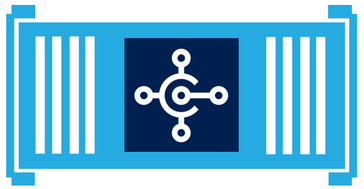
Local VMs

VMs on Azure

Azure Container Instances



Local Containers



ACIs

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeplomentent -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeploment -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeploment -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeploment -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeplomentent -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeplomentent -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

NAV TECH DAYS 2018
mibuso.com

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my little VS Code extension)

```
New-AzureRmResourceGroupDeployment
    -Name mydeploment -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

Azure Container Instances

How? Multiple ways

- Azure Portal GUI
- Azure command line or PowerShell cmdlets for single containers
- ARM template deployed manually → custom or pre-defined like quickstart template
- ARM template deployed through Azure command line or PowerShell cmdlets
- Probably more… (including my Bartev VSCode extension)

See it in the Portal GUI

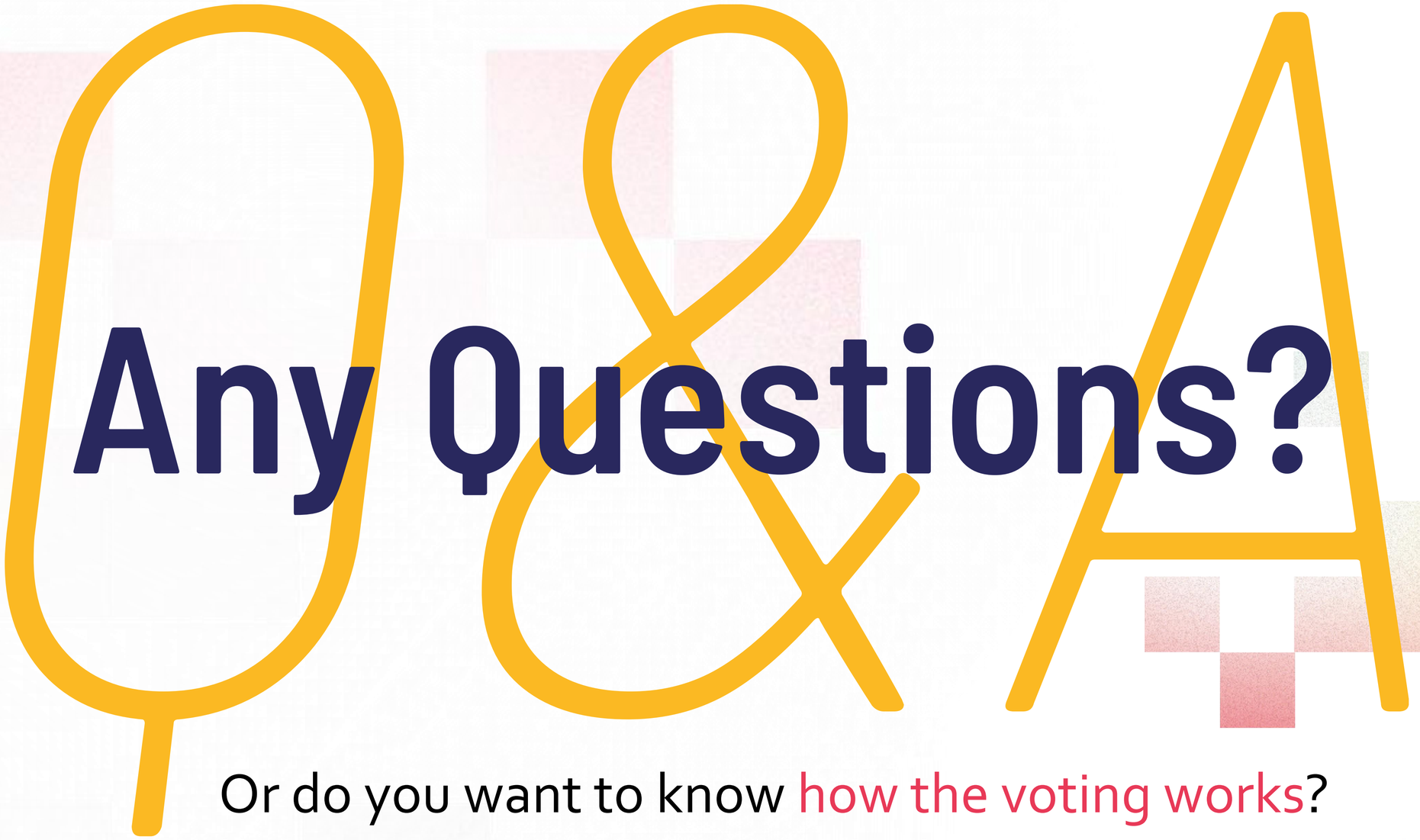
```
New-AzureRmResourceGroupDeployment
    -Name mydeploment -ResourceGroupName fk18mati -contGroupName matibasc
    -TemplateFile azuredeploy.json
    -cpuCores 4 -memoryInGb 12
    -dnsPrefix matibasc -azurecontainerSuffix .azurecontainer.io
    -navRelease bcinsider.azurecr.io/bcsandbox:de
    -username basc -password ngph2614 -acceptEula Y
    -asJob
```

NAV TECH DAYS 2018
mibuso.com

Q & A

Any Questions?

Or do you want to know how the voting works?
Spoiler: It involves Docker (Swarm)



Any Questions?

Or do you want to know how the voting works?
https://training.play-with-docker.com/swarm-stack-intro/


NAV
TECH
DAYS
2018
mibuso.com
Thank you
THANK YOU