// DAYS OF KNOWLEDGE

YOUR ANNUAL DYNAMICS 365 BUSINESS CENTRAL CONFERENCE

2019

Practical usage of Business Central Containers

Tobias Fenster & Freddy Kristiansen

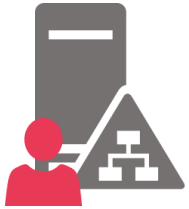
Agenda

- Introduction
  - Run a Business Central Container as your sandbox
- OS Compatibility
  - Selecting the right image platform
  - Building a Generic image that matches the host
- CI/CD
  - Utilizing Containers for Continuous Integration
- Authentication
  - Windows, AAD or Username/Password
- Multi-BC-Container Azure VMs
  - Using Traefik as Reverse Proxy
- Q&A

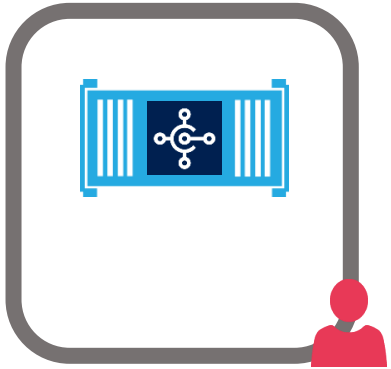

Windows authentication with group managed service accounts (gMSAs)

Infoma

How do gMSAs work with containers?



AD with a gMSA

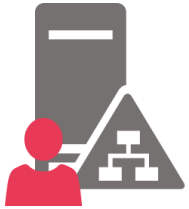


Container Host
with installed gMSA

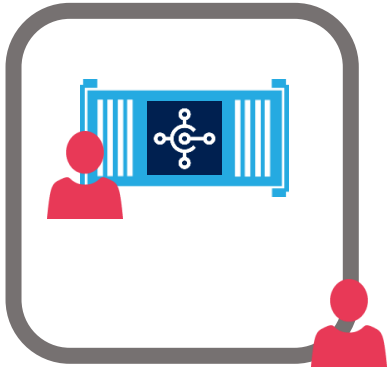
SQL with a DB
where the gMSA
has access

Infoma

How do gMSAs work with containers?



AD with a gMSA

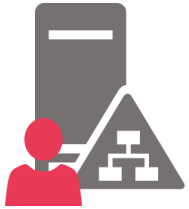


Container Host
with installed gMSA

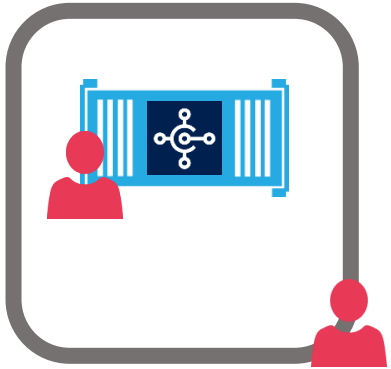
SQL with a DB
where the gMSA
has access

Infoma

How do gMSAs work with containers?



AD with a gMSA

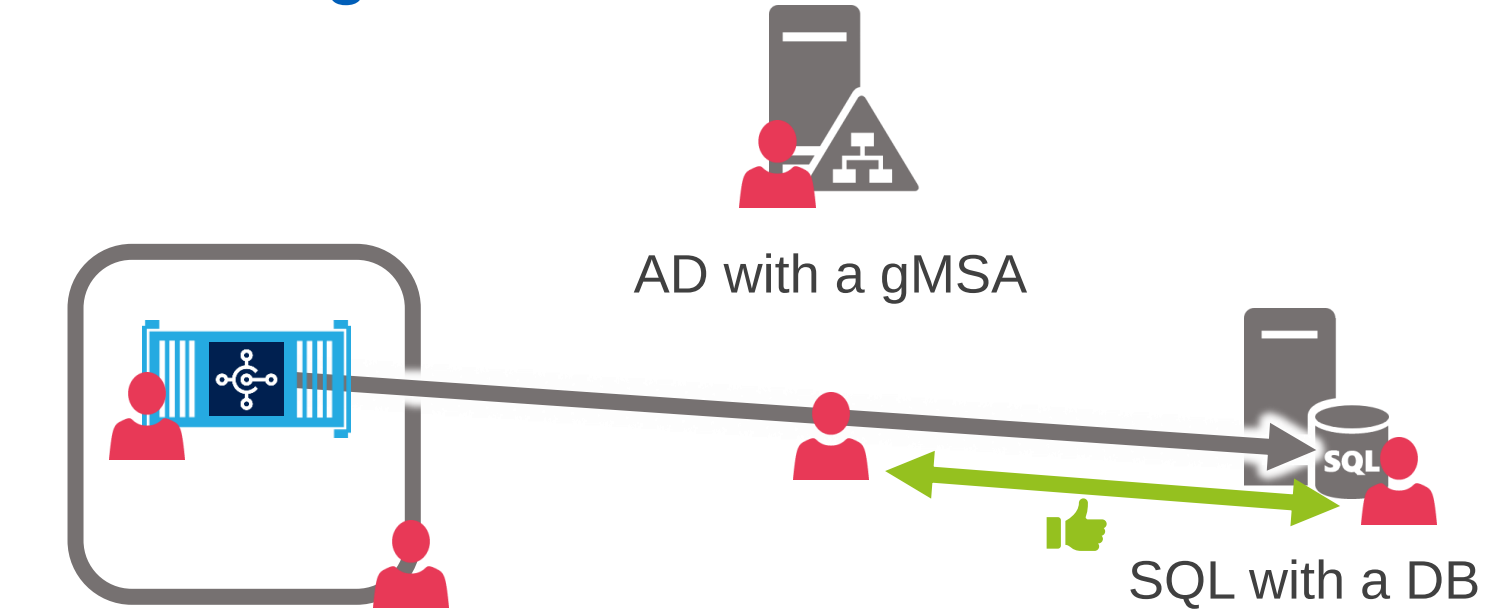


Container Host
with installed gMSA

SQL with a DB
where the gMSA
has access

Infoma

How do gMSAs work with containers?



Infoma

How do gMSAs work with containers?

Things to note

- 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 – but the host name is ignored and always the gMSA name is used → still 1 gMSA for every container
- 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 machine

Infoma

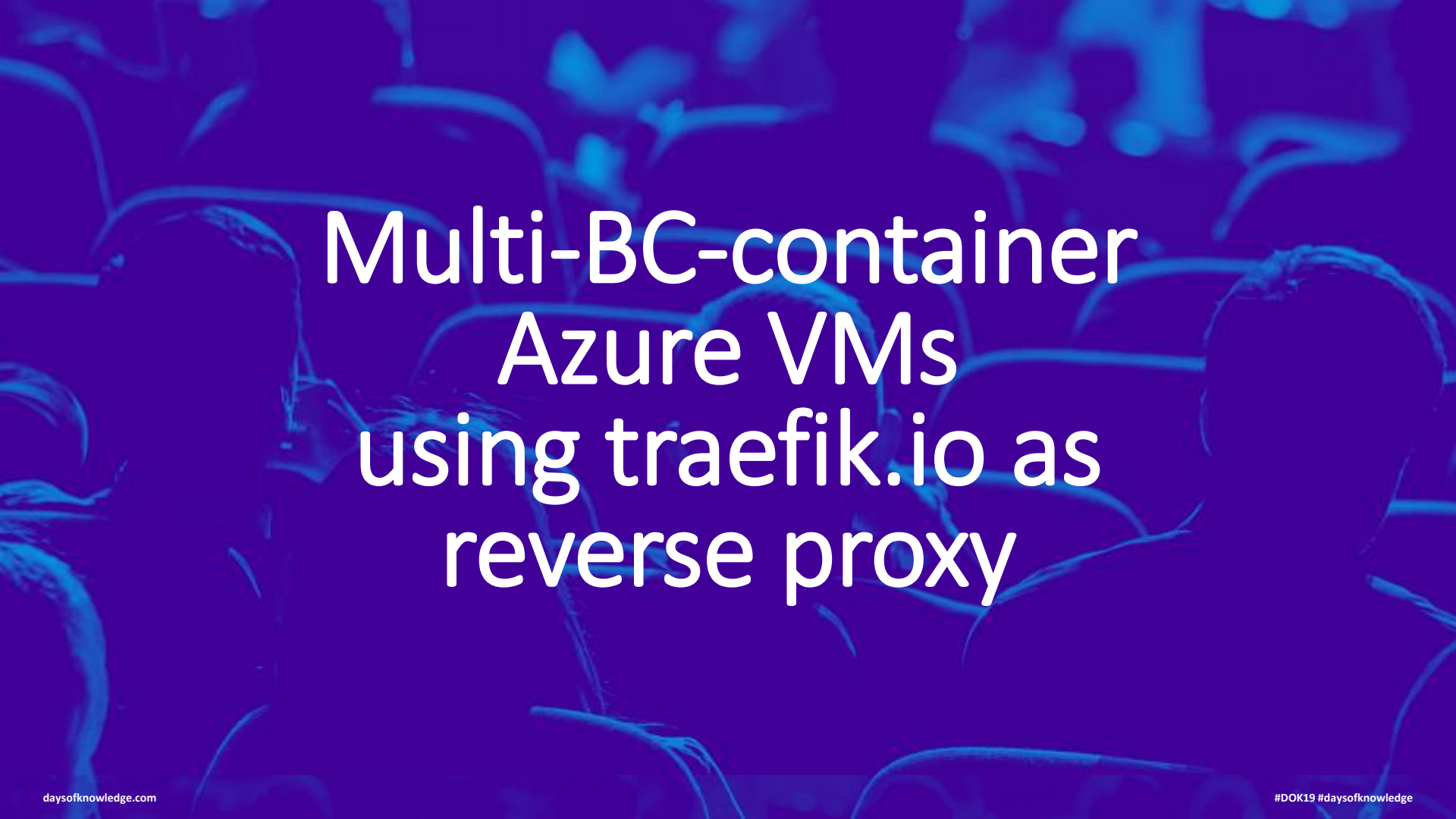
How do gMSAs work with containers?
Things to note

- 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


Demo


Multi-BC-container
Azure VMs
using traefik.io as
reverse proxy

Infoma

What problem are we solving?

Introduction to the scenario

- Docker containers allow running multiple versions / CUs of Business Central on the same VM
- Docker containers have a much lower resource overhead than full VMs
- Creating / starting and stopping / deleting containers is a lot quicker than full VMs

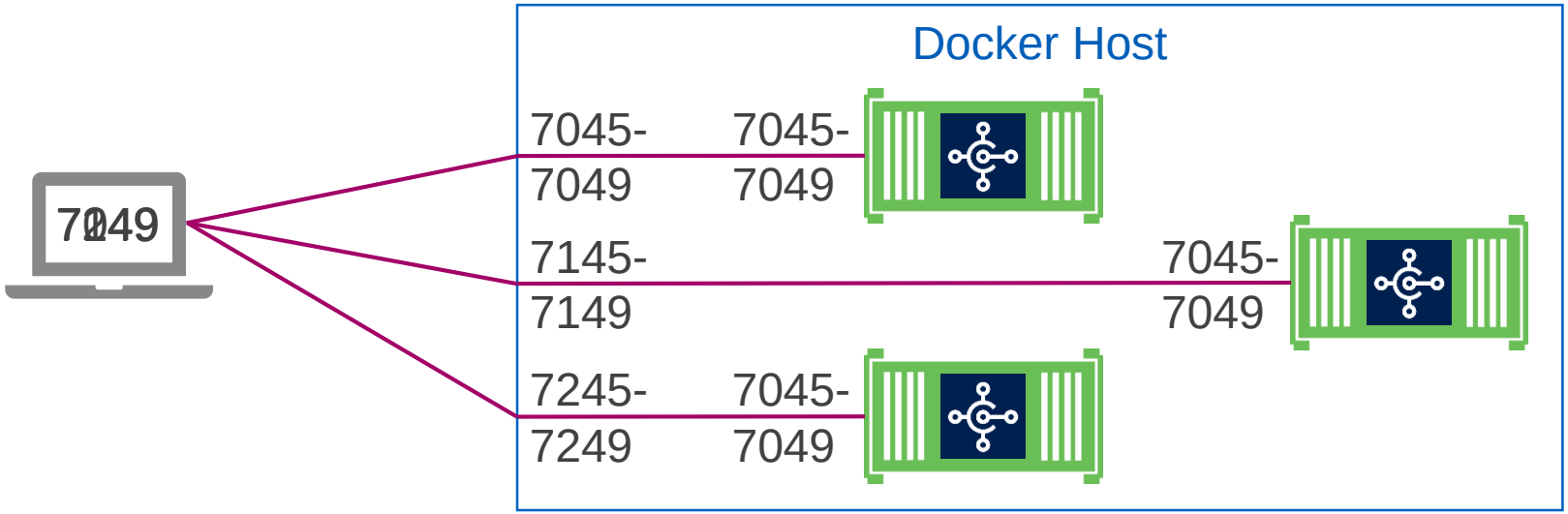
➔ You want to run multiple containers on the same VM

➔ But how can you connect your development / test / etc. machines to those containers?

How can we solve that problem?

Mapping ports

- Run your containers and map their ports to different host ports



How can we solve that problem?
Mapping ports – The good and the bad

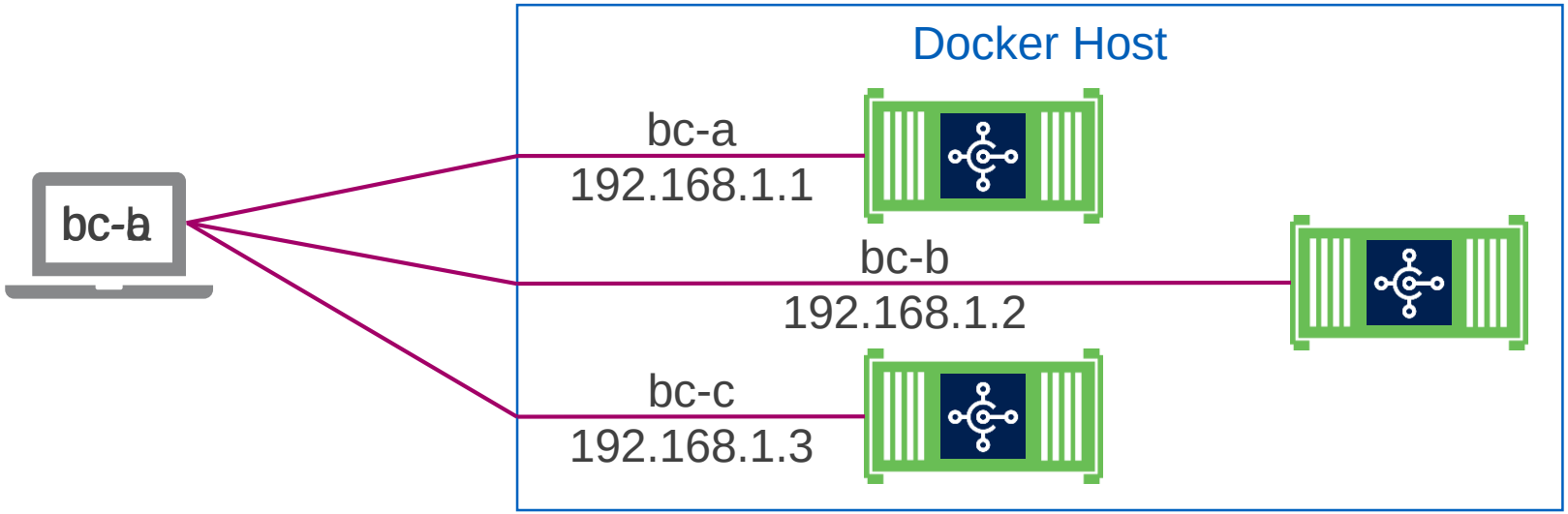
- Good:
  - Easy to connect to from the client (if you know the right port)
- Bad:
  - Always need to determine which ports are free for the next container
  - Don’t forget 80, 443, 8080
  - Need to open ports on the firewall of the VM
  - On Azure that becomes two firewalls (VM and Azure networking)

➔ Possible but somewhat complicated and error prone

How can we solve that problem?

Transparent networking

- Run every container with its own IP (and name)



How can we solve that problem?

Transparent networking – The good and the bad

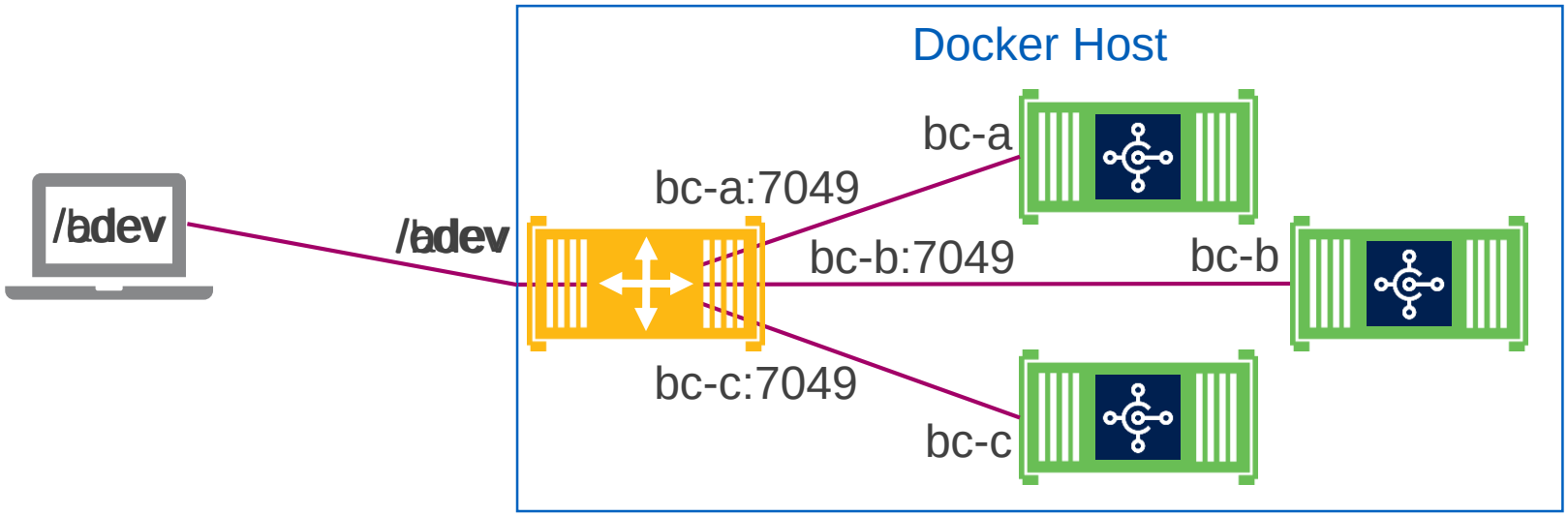
- Good:
  - Easy to connect to from the client (you only need the name)
  - Creating a new container is easy
- Bad:
  - Needs to be allowed on your network
  - Needs a specific setting on your hypervisor (MAC address spoofing)
  - Not possible on Azure

➔ Good solution for on prem if allowed but not for Azure

How can we solve that problem?

Reverse proxy

- Run your containers behind a reverse proxy



How can we solve that problem?

Reverse proxy – The good and the bad

- Good:
  - Easy to connect to from the client (you only need the name)
  - Creating a new container is easy and it works on Azure
  - You only need one entry point per service in the firewalls
- Bad:
  - One more component to set up and maintain
  - Non-TCP-traffic needs more work (RTC and SQL/finsql)
  - URLs returned from SOAP and REST endpoints not correct

➔ Good solution for both worlds, especially with automated setup

How can we solve that problem?

Reverse proxy – The details

- Implemented using traefik (https://traefik.io/)
  - Cloud-native, container-native reverse proxy
  - Easy to set up and run, e.g. integrated LetsEncrypt support
  - Picks new containers up by checking their labels
- Regex-based rules for the mapping, e.g.
  - https://myvm.westeurope.cloudapp.azure.com/bc-arest/* maps to http://bc-a:7048/NAV/OData/*
  - https://myvm.westeurope.cloudapp.azure.com/bc-a/* maps to http://bc-a:80/bc-a/*

How can we solve that problem?

Reverse proxy – The details

- Additional config for the Business Central container:
  - Set PublicODataBaseUrl, PublicSOAPBaseUrl, PublicWebBaseUrl and PublicDnsName so that Business Central knows what it is called from the outside
  - Set WebServerInstance to a different name as it otherwise insists on redirecting to /NAV
  - Health check needs to be different: Traefik only picks up healthy containers but for the regular health check to work, traefik routing needs to be in place…
- Traefik needs a setup file called traefik.toml

How can we solve that problem?

Reverse proxy – The details

- Integrated into
  - aka.ms/getbc and related Azure ARM templates with a „Use Traefik“ toggle
  - navcontainerhelper with -useTraefik
- Base setup needed
  - New navcontainerhelper cmdlet Setup-TraefikContainerForNavContainers
- Check techblog.axians-infoma.com in the next couple of days


Demo


Q&A

Learn more?

Blogs

- https://www.axians-infoma.com/techblog/
- https://freddysblog.com

Twitter

- @tobiasfenster
- @freddydk

Github

- https://github.com/microsoft/nav-docker
- https://github.com/microsoft/navcontainerhelper
- https://github.com/microsoft/nav-arm-templates


Thank you
for listening