DIRECTIONS ASIA

The SMB Focused Dynamics Partner Community

www.directionsasia.com

Real life scenarios for BC on Docker

Tobias Fenster, 01-04-2019, Kuala Lumpur

Introduction

Tobias Fenster

CTO at Axians Infoma

Microsoft MVP for Business Applications

@tobiasfenster and https://techblog.axians-infoma.com

Agenda

- Quick intro to Docker and the overall scenario
- Self-service container environments
- Multi-container environments
- Using Azure Container Instances

- Not part of the agenda: CI/CD, as we have a somewhat unusual approach but containers make a ton of sense there → check https://freddysblog.com/category/ci-cd/

Quick intro to Docker and the overall scenario

Photo by chuttersnap on Unsplash

Introduction to Docker

If you are not sure yet what this is

- What is Docker? Leading cross platform software container env
- 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!
- What is a Docker host? The (physical or virtual) machine where the containers are running

Introduction to Docker

If you are not sure yet what this is

- 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, also on Microsoft’s registry (mcr.microsoft.com)

Introduction to the scenario

Background to explain why we are doing what we are doing

- Axians Infoma is an ISV for > 1.200 customers with > 100 employees directly working on the product newsystem (program managers, developers, back office etc.)
- 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 itself
  - Time spent on local dev infrastructure is very likely time spent on something that won't improve theproduct or customer satisfaction

Introduction to the scenario

Background to explain why we are doing what we are doing

- Central team provides all 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
- Consequence of our scenario: Central Docker containers provided by our Release Management & Tooling team

Introduction to the scenario

A word on navcontainerhelper

- Main reason why we are not using navcontainerhelper a lot is that it mostly assumes local Docker installs and our ops professionals already have advanced Docker knowledge
- 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 won't have direct relevance for you as navcontainerhelper then would be your easiest route
  - Technology and concepts of this session should still be interesting

Self-service container environments

Photo by Rye Jessen on Unsplash

Self-service container environments

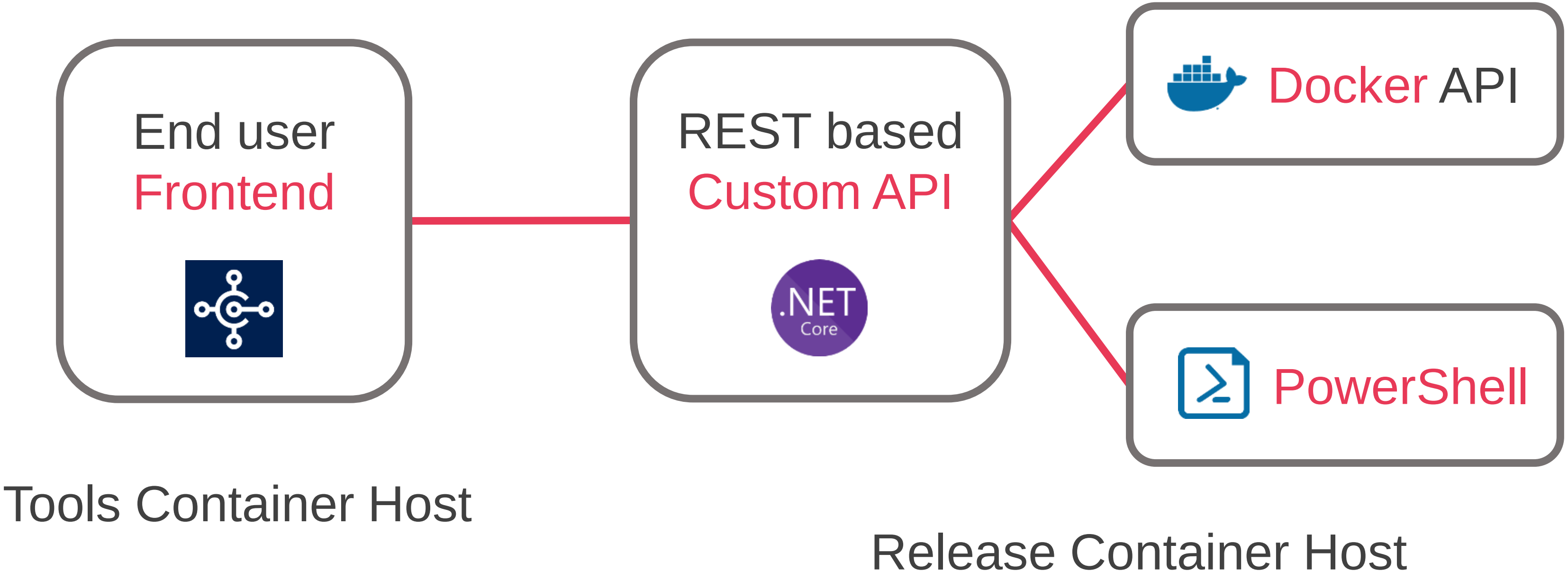
Overview

- Why? Easy access to releases
- 1-3 major releases, 4-6 bugfix releases for each country solution per year → up to 20 Infoma newsystem releases per year
- Business central / NAV cumulative updates, releases and previews
- All of those should be readily available for quick tests

Let’s create one

Self-service container environments

Overview



Self-service container environments

Details

- Client: Extension v2 in a Business Central Container
  - Available images maintained in a table, pulled nightly
  - Containers valid for max. 3 days, deleted nightly
  - Calls proxy API through a REST interface to create or delete containers, get status and logs
- Proxy API: Custom .NET Core application
  - 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

Self-service container environments

Details

- Container: Standard NAV / BC image from Microsoft with a couple of additional scripts and specific settings
  - Script 1: Grant an AD user or 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)
  - Parameters: Use Windows authentication and our dev license

Self-service container environments

Example docker run command

```
docker run --security-opt "credentialspec=file://testtfe.json"
--name testtfe --hostname testtfe -e accept_eula=y -e accept_outdated=y
--network MyTransparentNetwork -e clickonce=y -e usessl=y
-e auth=Windows -e username=admin -e password=Passw0rd*123
-e folders="c:\run\my=https://tools.axians-infoma.de/grant-useraccess.zip\nav-docker-
samples-grant-useraccess,c:\run\my=https://tools.axians-
infoma.de/invokeconversion.zip\nav-docker-samples-invoke-conversion"
-e DevDomain=FUM-GLOBAL -e DevGroup=GRP_INFOMA_DEV_ALL
-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
```

Demo: See the running container

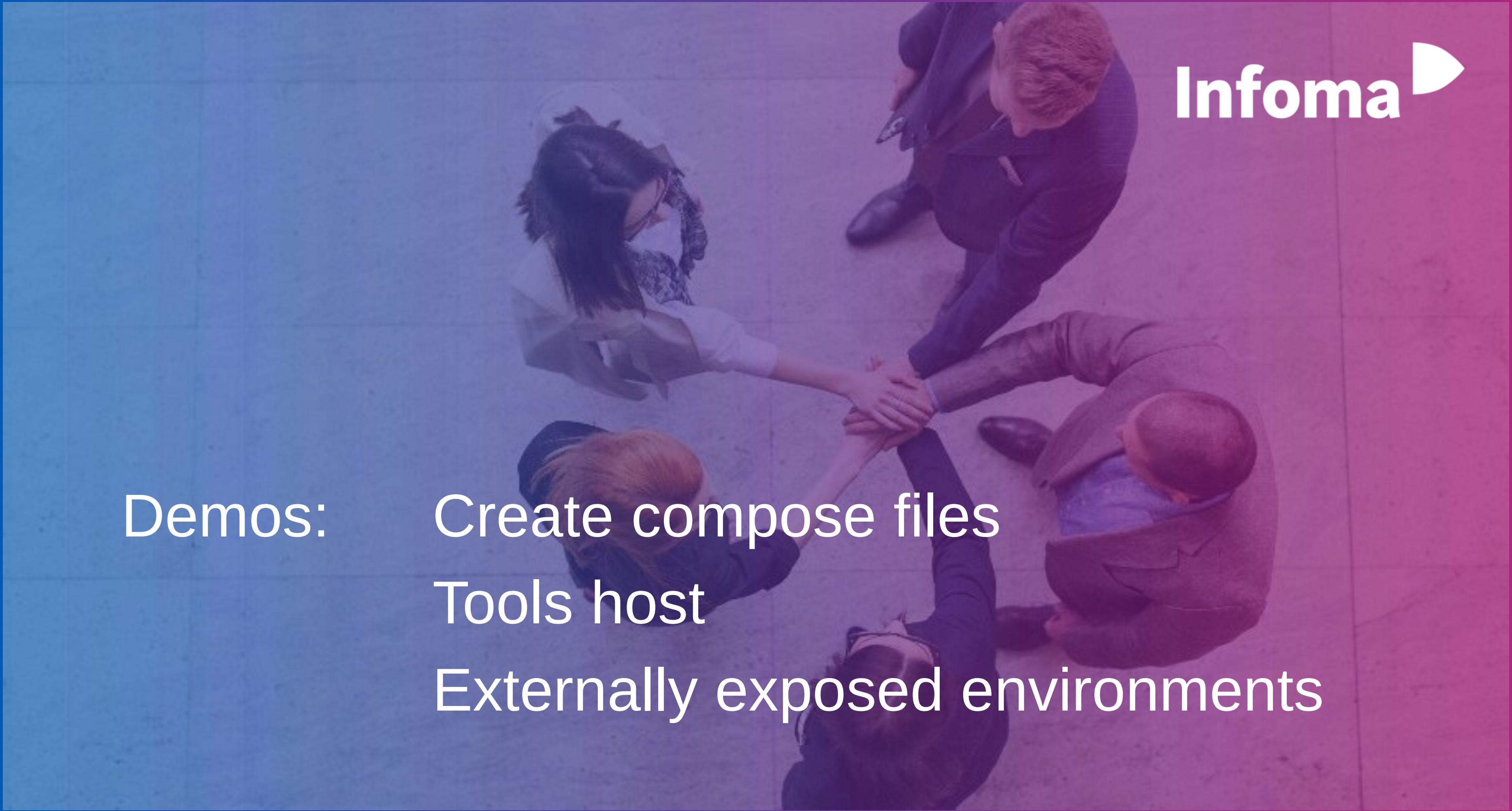
Multi-container environments

Photo by Hans-Jorg Van Broekhoven on Unsplash

Multi-container environments

Overview

- Why? Multiple very similar containers or more complex scenarios
  - Release tests with 10 databases and all need the same container
  - Tools host with different images
  - Externally available environment with multiple endpoints but only a reverse proxy is exposed
- How? Docker compose
  - Describes in YAML the containers, their configuration and the networking setup
  - For the scenario with 10 identical containers: PowerShell script to generate compose file from templates



Infoma

Demos: Create compose files

Tools host

Externally exposed environments

Multi-container environments

Details

- YAML definitions 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

Multi-container environments

Additions

- 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 happened very recently including Windows authentication

Azure Container Instances

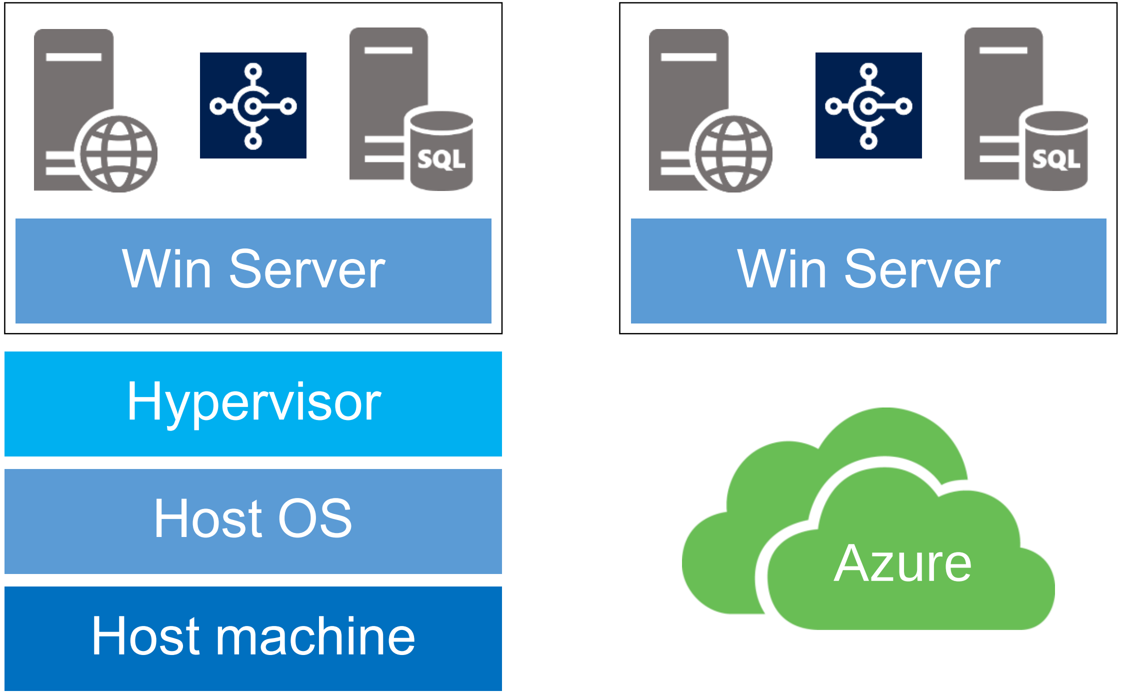
Photo by Vidar Nordli-Mathisen on Unsplash

Using Azure Container Instances

Overview

- 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
  - Paid on demand by seconds of CPU / RAM / Windows license (see)
- How? Multiple ways
  - Azure Portal GUI, Azure command line or PowerShell cmdlets
  - ARM template deployed manually, through Azure command line or PowerShell cmdlets
  - Probably more... (including my little VS Code extension)

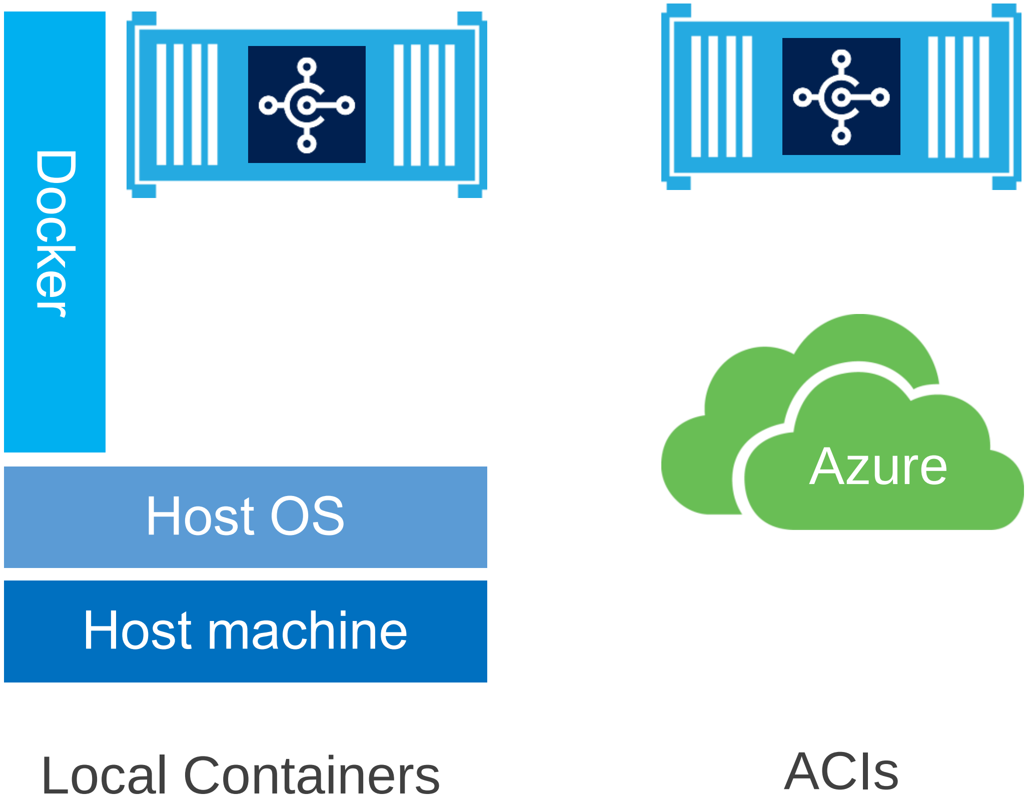
Azure VMs



Local VMs

VMs on Azure

ACIs



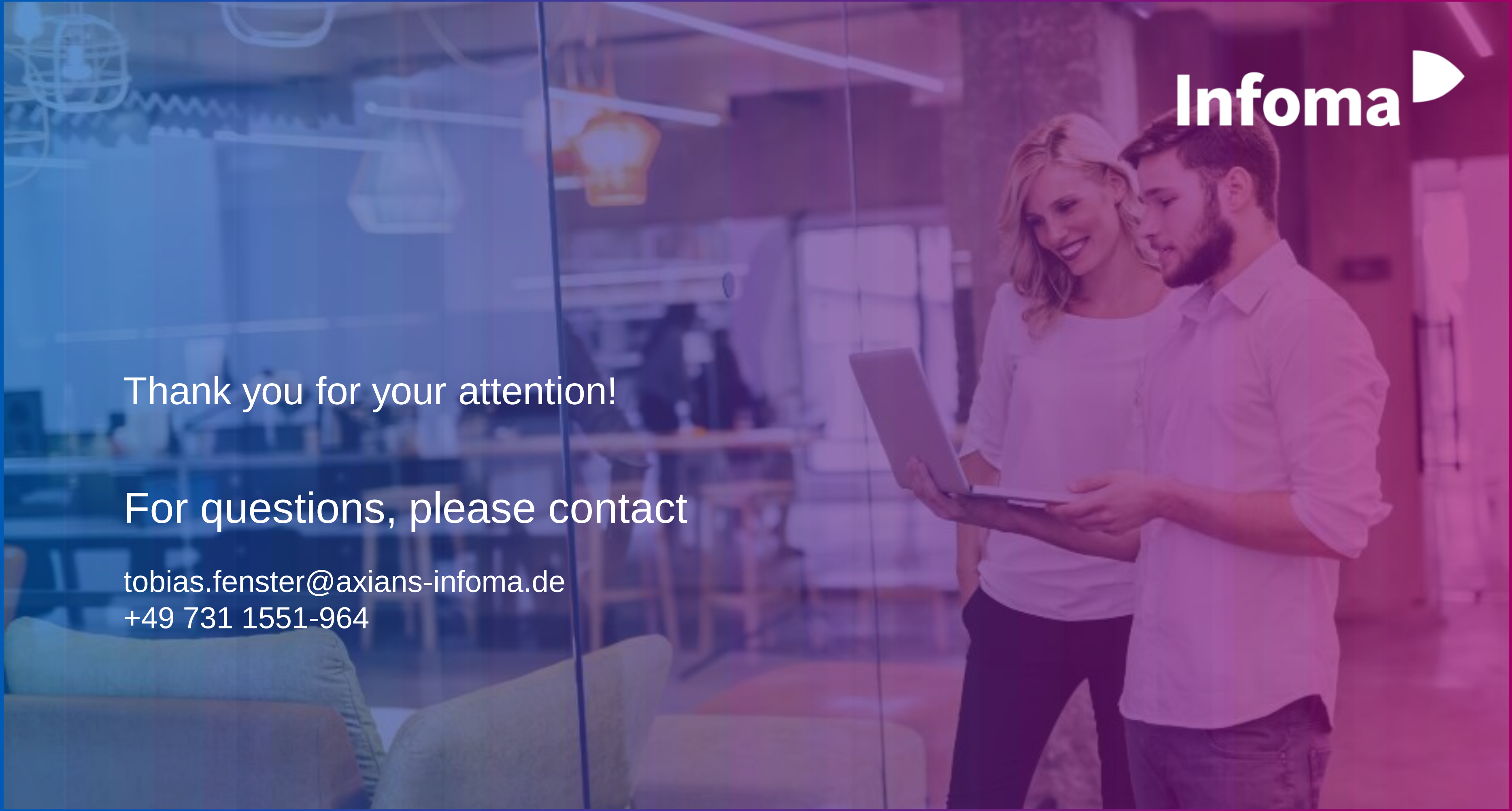
Infoma

Demo: Create an ACI

Using Azure Container Instances

Additions

- Azure Container Registry (ACR) offers "serverless" build infrastructure
  - Create your own image without installing Docker
  - Especially useful if you want to have custom images based on multiple standard images (like NAV 2017, NAV 2018, BC OnPrem, BC Sandbox, nightly build, etc.)



Thank you for your attention!

For questions, please contact

tobias.fenster@axians-infoma.de
+49 731 1551-964

DIRECTIONS
ASIA

The SMB Focused Dynamics Partner Community

www.directionsasia.com