



Running

Spring Boot Apps

on

Docker Windows Containers

with

Ansible

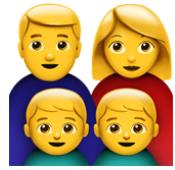
Jonas Hecht | Senior IT-Nerd | @jonashackt



Integrating

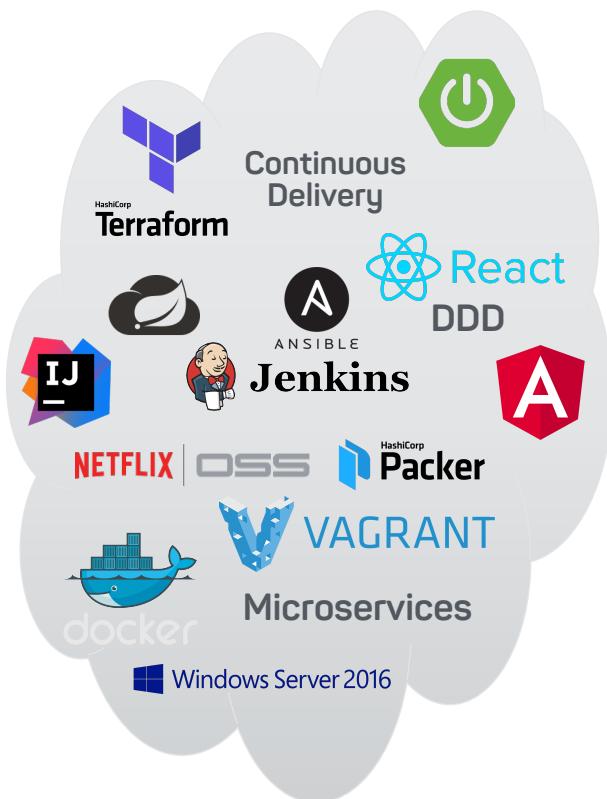
Legacy Windows Apps with Microservices

Jonas Hecht | Senior IT-Nerd | [@jonashackt](https://twitter.com/jonashackt)



meetup.com/jugthde

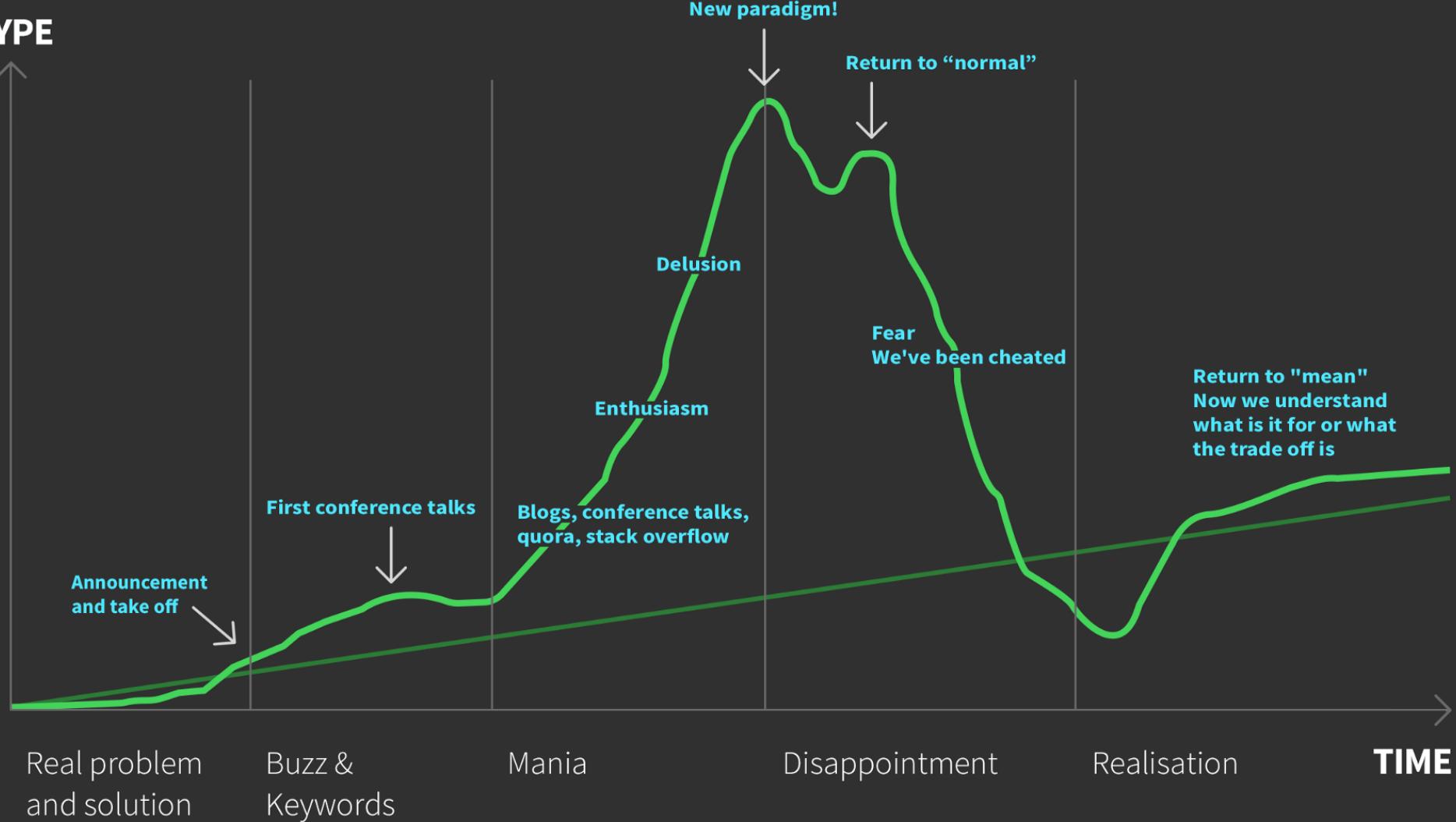
tools & methodologies



real world



HYPE



Real problem
and solution

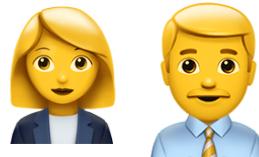
Buzz &
Keywords

Mania

Disappointment

Realisation

TIME



Build (and scale) a Windows C/C++ backed Spring Boot App!



Make no compromises -
automate it!

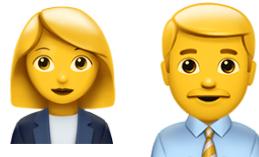
100% comprehensible



Open Source

build it from the ground up

(bit.ly/2q4sD3x)



Build (and scale) a Windows C/C++ backed Spring Boot App!



Build (and scale) a Windows C/C++ backed Spring Boot App!

1. Windows box
2. Ansible provisions Windows
3. Prepare Docker on Windows
4. Run Spring Boot App on Docker Windows Container
5. Scale Spring Boot Apps

1. Windows box



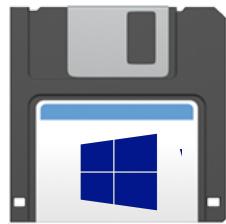
developer.microsoft.com/en-us/microsoft-edge/tools/vms/
atlas.hashicorp.com/boxes/search?q=windows+10

ansible-windows-docker-springboot_default_1487243623466_51562 [Running]

```
Administrator: Windows PowerShell
PS C:\Users\vagrant> (Get-ItemProperty -Path c:\windows\system32\hal.dll).VersionInfo.FileVersion
10.0.14393.206 (rs1_release.160915-0644)
PS C:\Users\vagrant>
```



microsoft.com/evalcenter/evaluate-windows-server-2016



Evaluation ISO





Autounattend.xml



windows_server_2016_docker.json



vagrantfile-windows_2016.template



Evaluation ISO



HashiCorp
Packer





Demo!

github.com/jonashackt/ansible-windows-talk#1-windows-box

Build (and scale) a Windows C/C++ backed Spring Boot App!



- Windows box
- 2. Ansible provisions Windows
- 3. Prepare Docker on Windows
- 4. Run Spring Boot App on Docker Windows Container
- 5. Scale Spring Boot Apps

2. Ansible provisions Windows



FOREMAN



CFEngine



HashiCorp



docs.ansible.com



blog.codecentric.de/en/2017/01/ansible-windows-spring-boot/

Powershell 3.0+

github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1

Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser



FOLDERS

- ansible-windows-simple
 - group_vars
 - all.yml
- .gitignore
- hostsfile
- LICENSE
- README.md
- windows-playbook.yml

all.yml

```
1 ansible_user: vagrant
2 ansible_password: vagrant
3 ansible_port: 55986 # not 5986, as we would use for non-virtualized environments
4 ansible_connection: winrm
5
6 # The following is necessary for Python 2.7.9+ when using default WinRM self-signed certificates:
7 ansible_winrm_server_cert_validation: ignore
```

```
ansible stageName -i hostsfile -m win_ping
```

```
ansible-playbook -i hostsfile playbookName.yml --extra-vars "host=stageName"
```

--syntax-check



--check



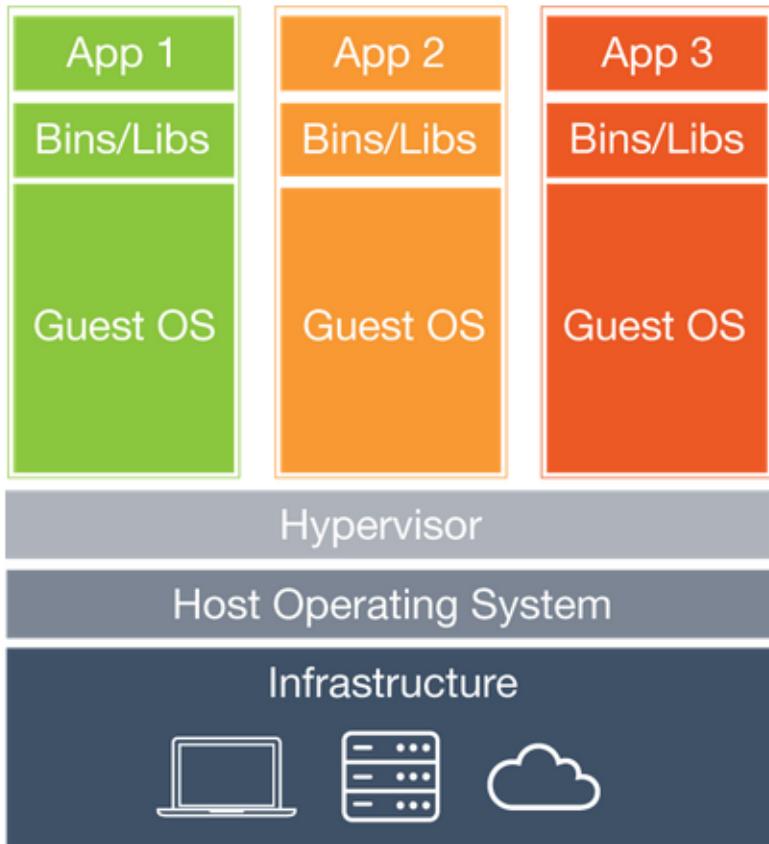
Demo!

github.com/jonashackt/ansible-windows-talk#2-ansible-provisions-windows

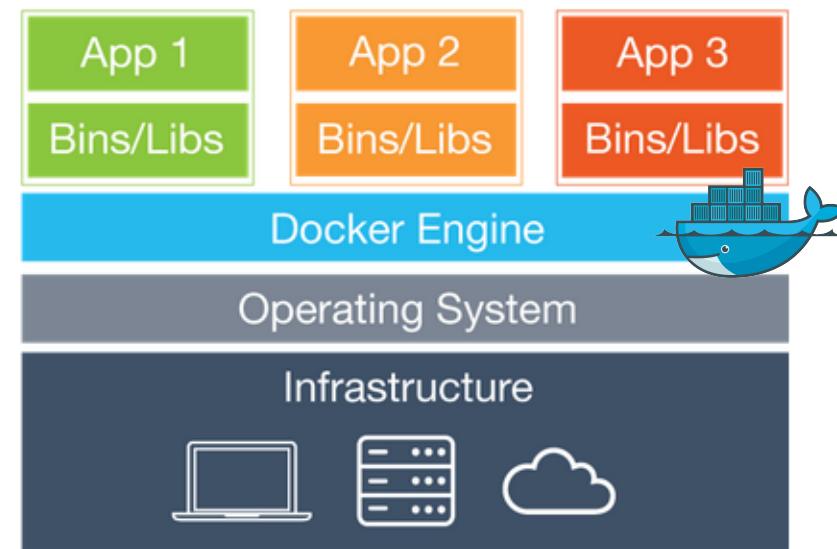
Build (and scale) a Windows C/C++ backed Spring Boot App!

-  Windows box
 -  Ansible provisions Windows
3. Prepare Docker on Windows
 4. Run Spring Boot App on Docker Windows Container
 5. Scale Spring Boot Apps

3. Prepare Docker on Windows



Virtual Machines



Containers



„Ok... I am a bit confused. You're saying, Windows should be able to run Containers containing Windows itself?“

„Well, are there...? “



hub.docker.com/r/microsoft/

„Ohh... Does this mean, I can use the Docker API to work
with Windows containers...? “



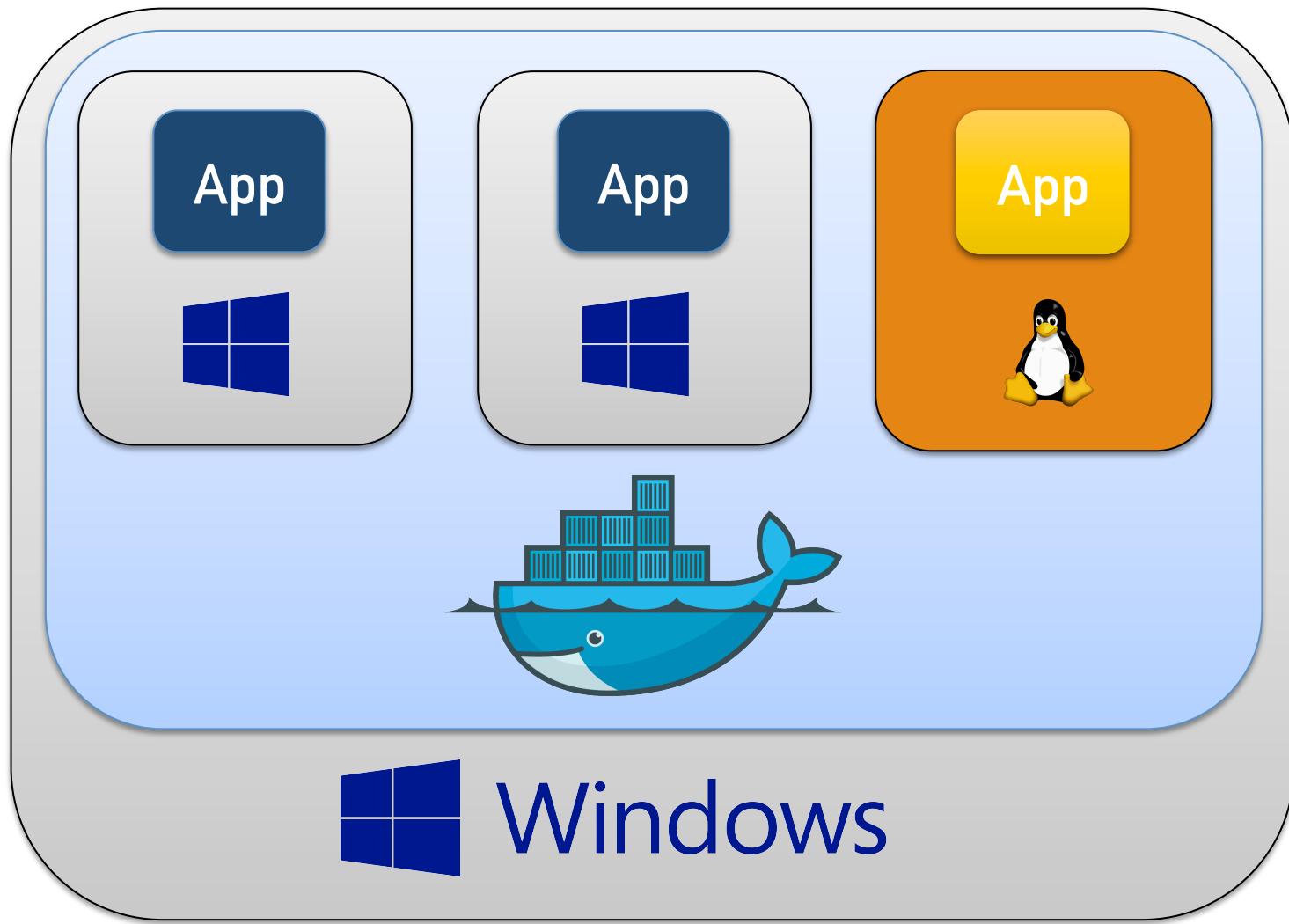
„Wait... Powershell??“

Yes.

„Ah, I see... but...“

What?

„It's Microsoft! They for sure developed their
own Docker I assume...“

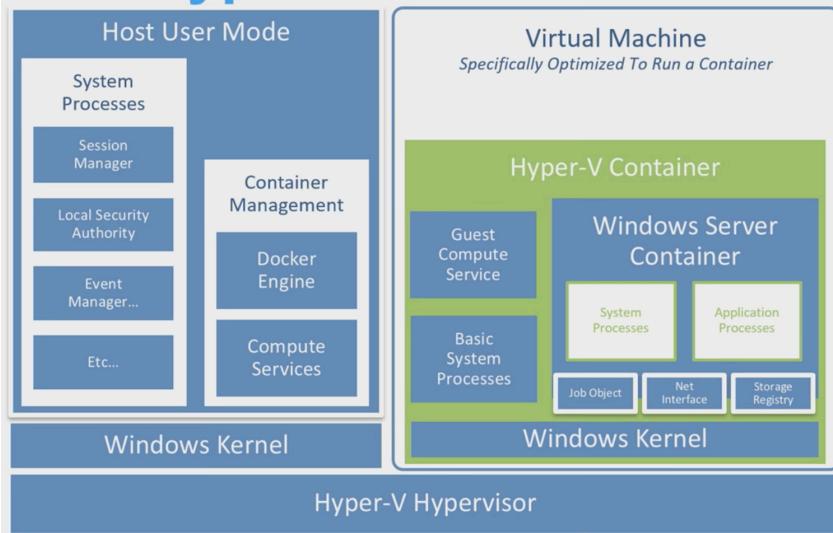


blog.docker.com/2016/09/docker-microsoft-partnership/

github.com/docker/for-win



Hyper-V Containers

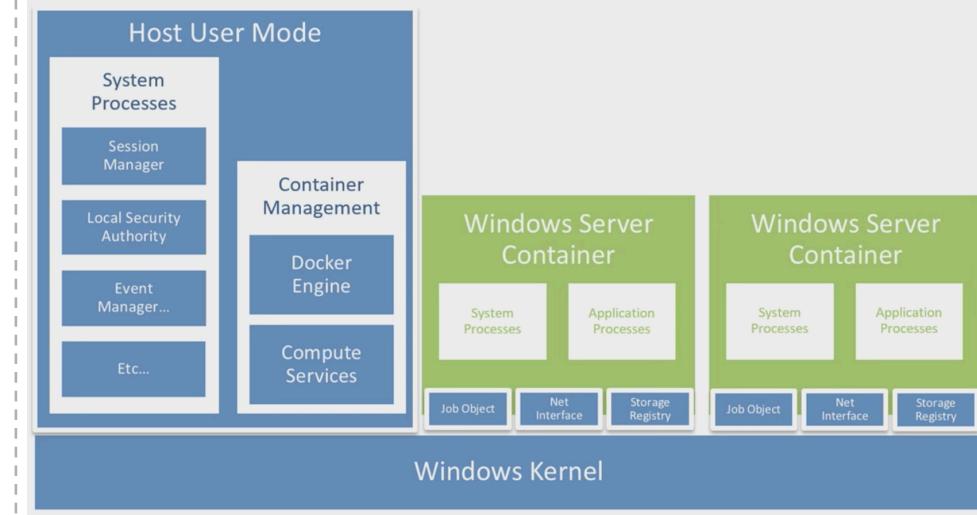


Windows Server 2016

Windows 10



Windows Server Containers



Windows Server 2016





dockercon 16



13:16 / 45:15

Base images

- Distributed by Microsoft
- Two options
 - windowsservercore: large (huge?), highly compatible
 - nanoserver: small, fast, smaller API surface

Administrator: C:\Windows\System32\cmd.exe					
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE	
windowsservercore	10.0.14300.1000	5bc36a335344	5 weeks ago	9.354 GB	
windowsservercore	latest	5bc36a335344	5 weeks ago	9.354 MB	
nanoserver	10.0.14300.1016	3f5112ddd185	5 weeks ago	810.2 MB	
nanoserver	latest	3f5112ddd185	5 weeks ago	810.2 MB	

dockercon 16

FROM microsoft/windowsservercore:latest



VS.

FROM microsoft/nanoserver:latest





1. Check Windows build number
2. Install Windows Features container
3. Install current Docker version service
4. Install docker-compose
5. Run a first Windows container inside your Windows
6. Building springboot-oraclejre-nanoserver





WinRM



1. **Check minimum build number**
2. Install Windows Features containers and Hyper-V
3. Install current Docker version & register Windows service
4. Install docker-compose
5. Run a first Windows container inside your Windows
6. Building springboot-oraclejre-nanoserver



1. Check minimum build number
- 2. Install Windows Features containers and Hyper-V**
3. Install current Docker version & register Windows service
4. Install docker-compose
5. Run a first Windows container inside your Windows
6. Building springboot-oraclejre-nanoserver



1. Check minimum build number
2. Install Windows Features containers and Hyper-V
- 3. Install current Docker version & register Windows service**
4. Install docker-compose
5. Run a first Windows container inside your Windows
6. Building springboot-oraclejre-nanoserver



1. Check minimum build number
2. Install Windows Features containers and Hyper-V
3. Install current Docker version & register Windows service
- 4. Install docker-compose**
5. Run a first Windows container inside your Windows
6. Building springboot-oraclejre-nanoserver



1. Check minimum build number
2. Install Windows Features containers and Hyper-V
3. Install current Docker version & register Windows service
4. Install docker-compose
- 5. Run a first Windows container inside your Windows**
6. Building springboot-oraclejre-nanoserver



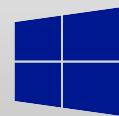
Demo!

github.com/jonashackt/ansible-windows-talk#3-prepare-docker-on-windows

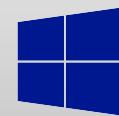


ANSIBLE

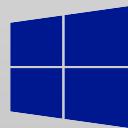
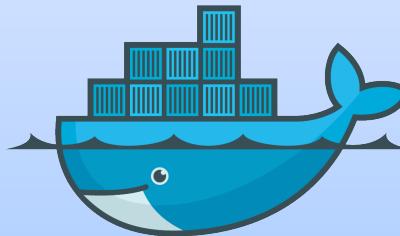
App



App



App



Windows



VAGRANT



VirtualBox

macOS



HashiCorp
Packer



Demo again!

github.com/jonashackt/ansible-windows-talk#3-prepare-docker-on-windows

Build (and scale) a Windows C/C++ backed Spring Boot App!

-  Windows box
 -  Ansible provisions Windows
 -  Prepare Docker on Windows
4. Run Spring Boot App on Docker Windows Container
 5. Scale Spring Boot Apps

4. Run Spring Boot App on Docker Windows Container

spring.io/projects



SPRING IO PLATFORM

Provides a cohesive, versioned platform for building modern applications. It is a modular, enterprise-grade distribution that delivers a curated set of dependencies.



SPRING FRAMEWORK

Provides core support for dependency injection, transaction management, web apps, data access, messaging and more.



SPRING BOOT

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.



SPRING INTEGRATION

Supports the well-known Enterprise Integration Patterns via lightweight messaging and declarative adapters.



SPRING BATCH

Simplifies and optimizes the work of processing high-volume batch operations.



SPRING SOCIAL

Easily connects your applications with third-party APIs such as Facebook, Twitter, LinkedIn, and more.



SPRING AMQP

Applies core Spring concepts to the development of AMQP-based messaging solutions.



SPRING CLOUD

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.



SPRING DATA

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.



SPRING MOBILE

Simplifies the development of mobile web apps through device detection and progressive rendering options.



SPRING FOR ANDROID

Provides key Spring components for use in developing Android applications.



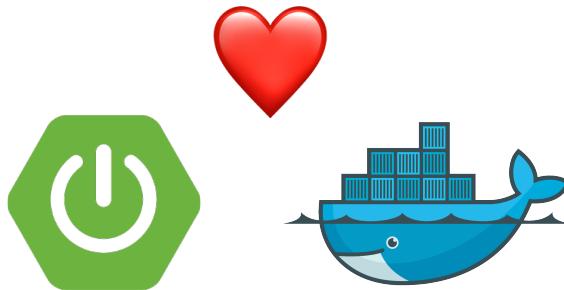
SPRING WEB FLOW

Supports building web applications with controlled navigation such as checking in for a flight or applying for a loan.



SPRING WEB SERVICES

Facilitates the development of contract-first SOAP web services.



„Takes an opinionated view of building **production-ready** Spring applications. **Spring Boot** favors convention over configuration and is designed to get you **up and running as quickly as possible**.“

```
java -jar weatherbackend-0.0.1-SNAPSHOT.jar
```



Demo!

github.com/jonashackt/ansible-windows-talk#4-run-spring-boot-app-on-docker-windows-container

Dockerfile



```
FROM springboot-oraclejre-nanoserver:latest

MAINTAINER Jonas Hecht

# Expose the apps Port
EXPOSE 8088

# Add Spring Boot app.jar to Container
ADD weatherbackend-0.0.1-SNAPSHOT.jar app.jar

# Fire up our Spring Boot app by default
CMD ["java.exe", "-jar app.jar --server.port=8088"]
```

```
docker build . --tag myRealCoolImage:latest
```

docs.docker.com/engine/reference/builder/



6. Building `springboot-oraclejre-nanoserver`

github.com/jonashackt/ansible-windows-talk#3-prepare-docker-on-windows



github.com/jonashackt/ansible-windows-talk#4-run-spring-boot-app-on-docker-windows-container



Demo!

github.com/jonashackt/ansible-windows-talk#4-run-spring-boot-app-on-docker-windows-container

Build (and scale) a Windows C/C++ backed Spring Boot App!

-  Windows box
 -  Ansible provisions Windows
 -  Prepare Docker on Windows
 -  Run Spring Boot App on Docker Windows Container
5. Scale Spring Boot Apps

5. Scale Spring Boot Apps

spring.io/projects



SPRING IO PLATFORM

Provides a cohesive, versioned platform for building modern applications. It is a modular, enterprise-grade distribution that delivers a curated set of dependencies.



SPRING FRAMEWORK

Provides core support for dependency injection, transaction management, web apps, data access, messaging and more.



SPRING BOOT

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.



SPRING INTEGRATION

Supports the well-known Enterprise Integration Patterns via lightweight messaging and declarative adapters.



SPRING BATCH

Simplifies and optimizes the work of processing high-volume batch operations.



SPRING SOCIAL

Easily connects your applications with third-party APIs such as Facebook, Twitter, LinkedIn, and more.



SPRING AMQP

Applies core Spring concepts to the development of AMQP-based messaging solutions.



SPRING CLOUD

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.



SPRING DATA

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.



SPRING MOBILE

Simplifies the development of mobile web apps through device detection and progressive rendering options.



SPRING FOR ANDROID

Provides key Spring components for use in developing Android applications.



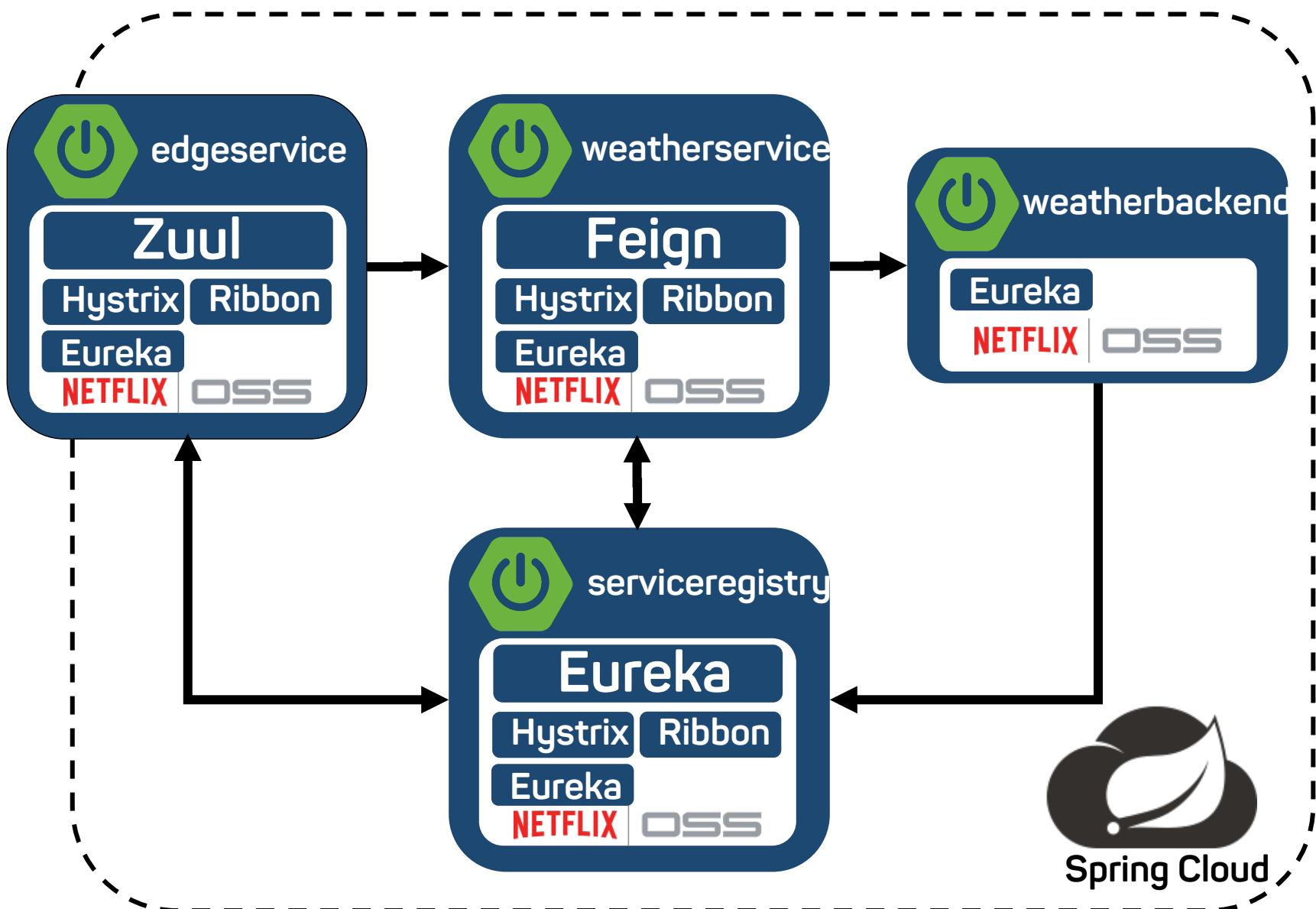
SPRING WEB FLOW

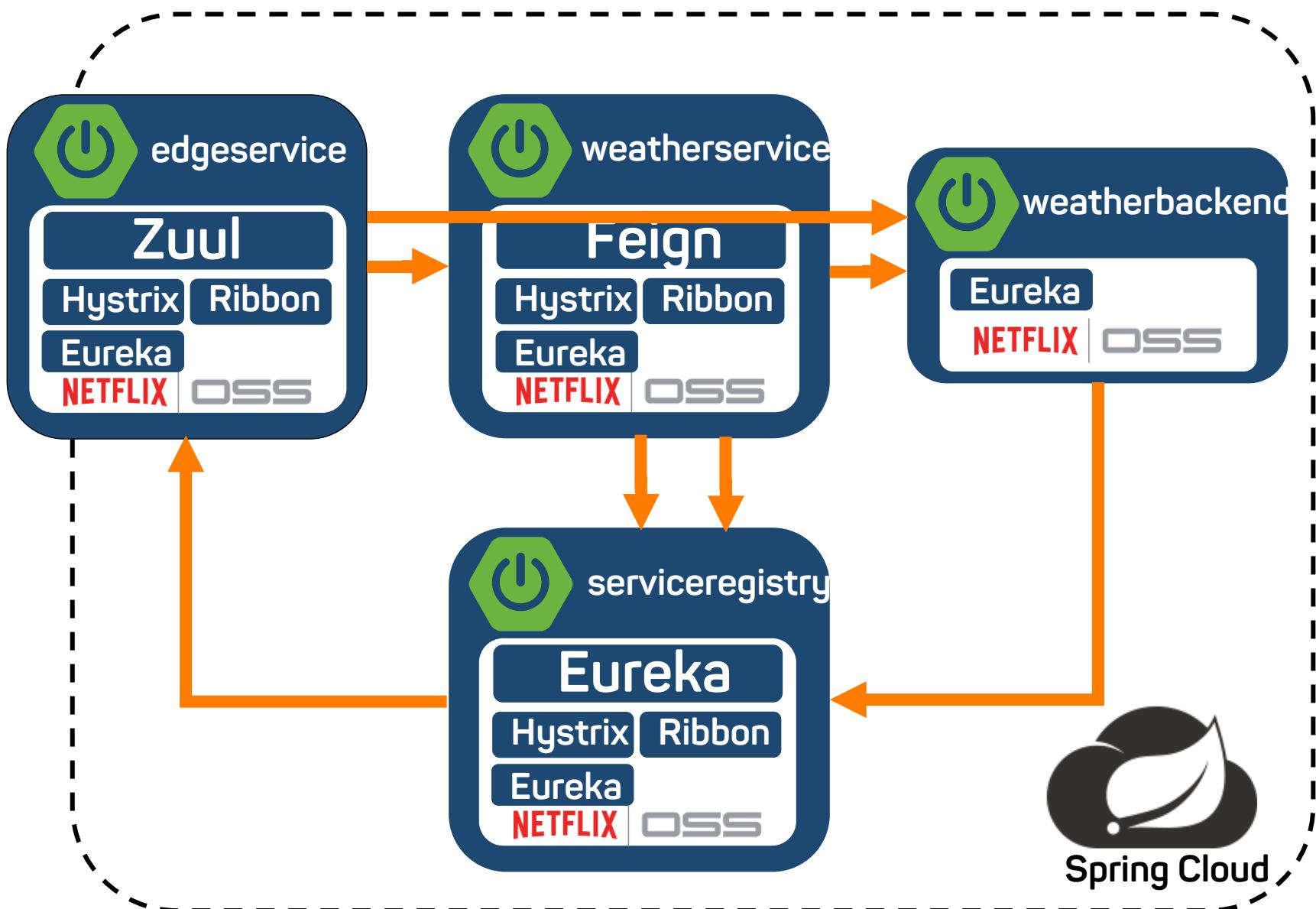
Supports building web applications with controlled navigation such as checking in for a flight or applying for a loan.



SPRING WEB SERVICES

Facilitates the development of contract-first SOAP web services.

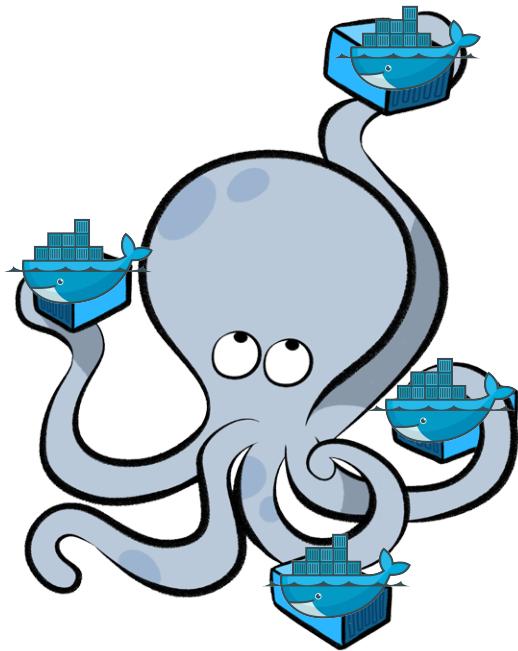




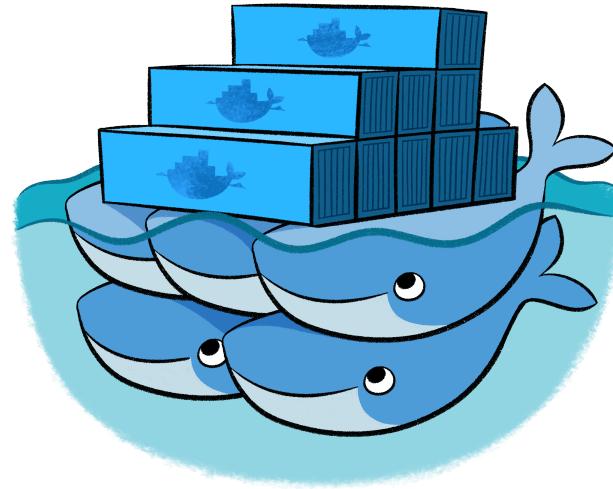
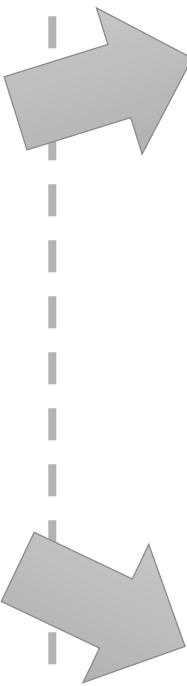


Demo!

github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps



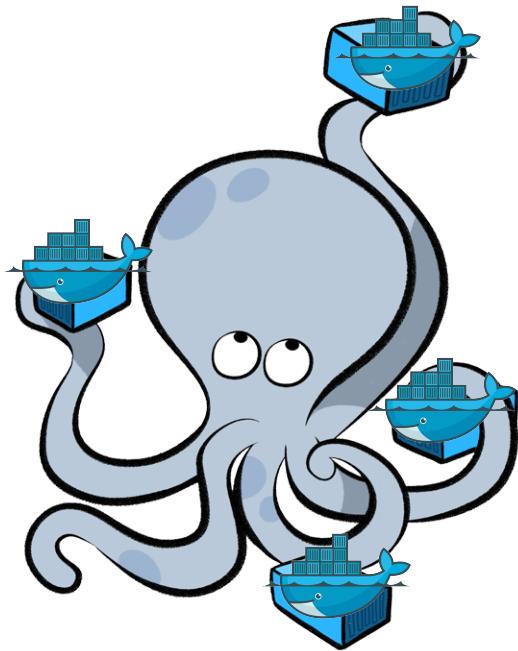
Docker Compose



Docker Swarm



kubernetes



Docker Compose

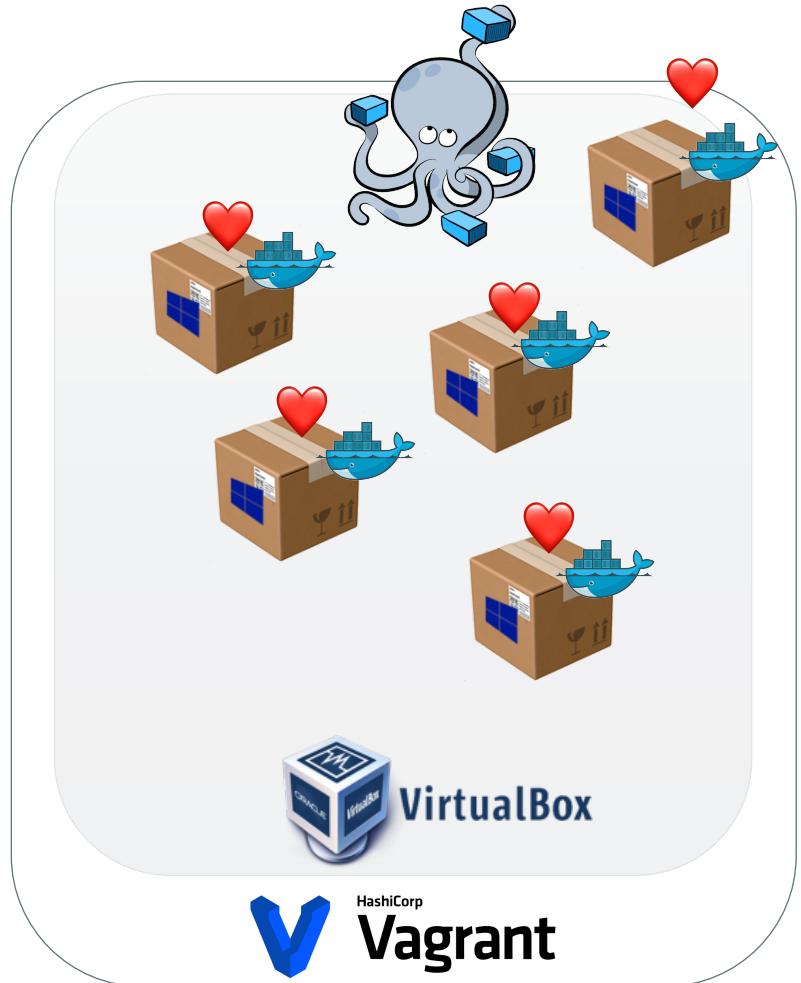
```
version: '3.1'

services:

  weatherbackend:
    build: ./weatherbackend
    ports:
      - "8090"
    tty:
      true
    restart:
      unless-stopped

  weatherservice:
    build: ./weatherservice
    ports:
      - "8095:8095"
    tty:
      true
    restart:
      unless-stopped

networks:
  default:
    external:
      name: "nat"
```



github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps

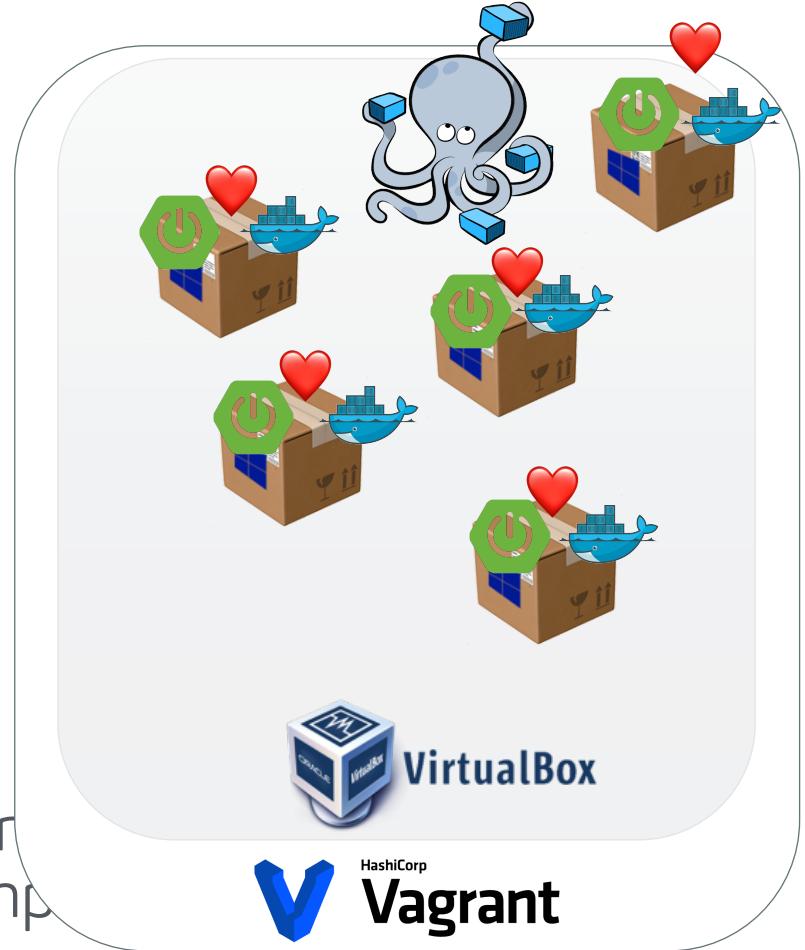


Demo!

github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps



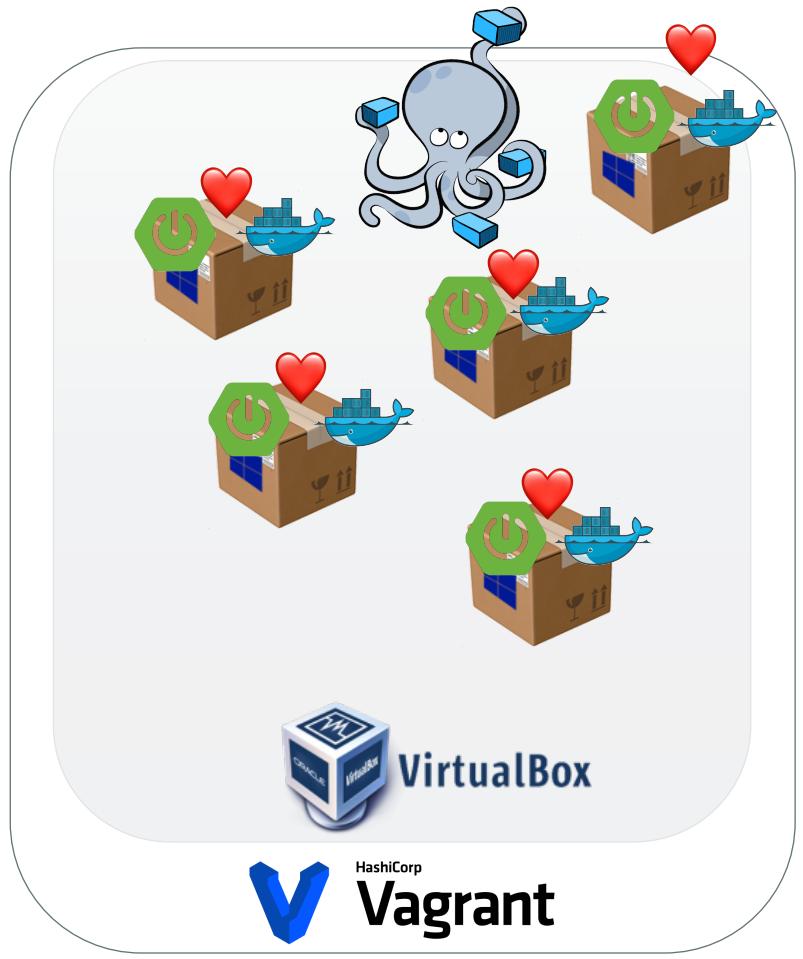
1. Preparing the App's Files for later
2. Run all Services with Docker Compose
3. Healthcheck all Services



github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps

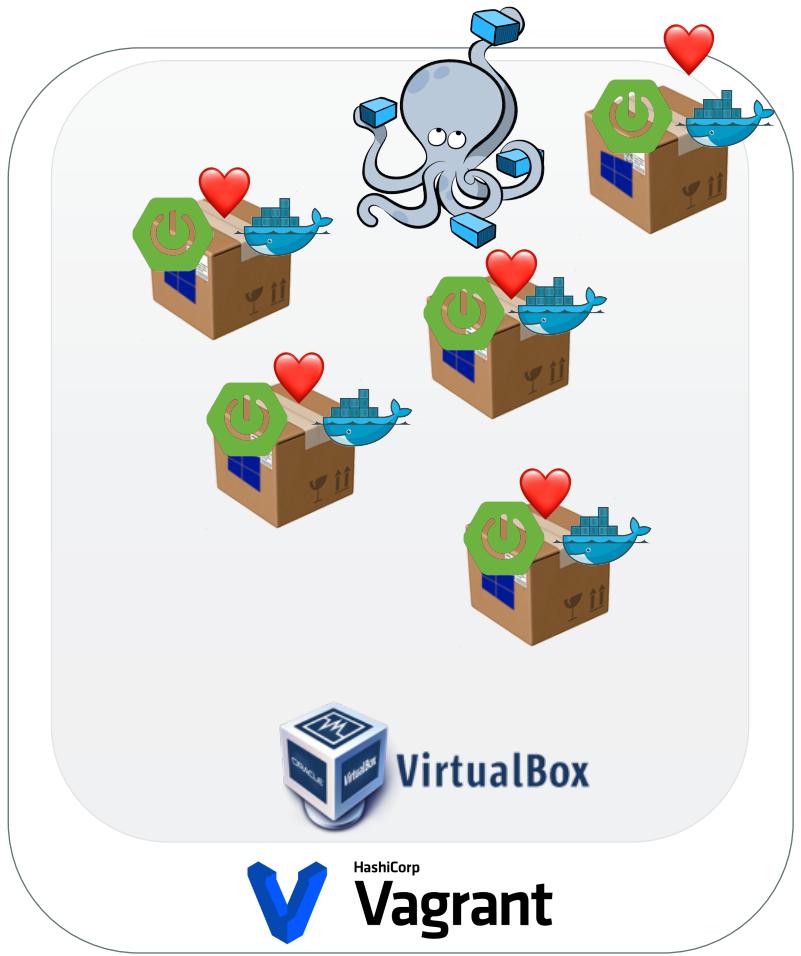


WinRM



1. **Preparing the App's Files for later docker-compose run**
2. Run all Services with Docker Compose
3. Healthcheck all Services

github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps

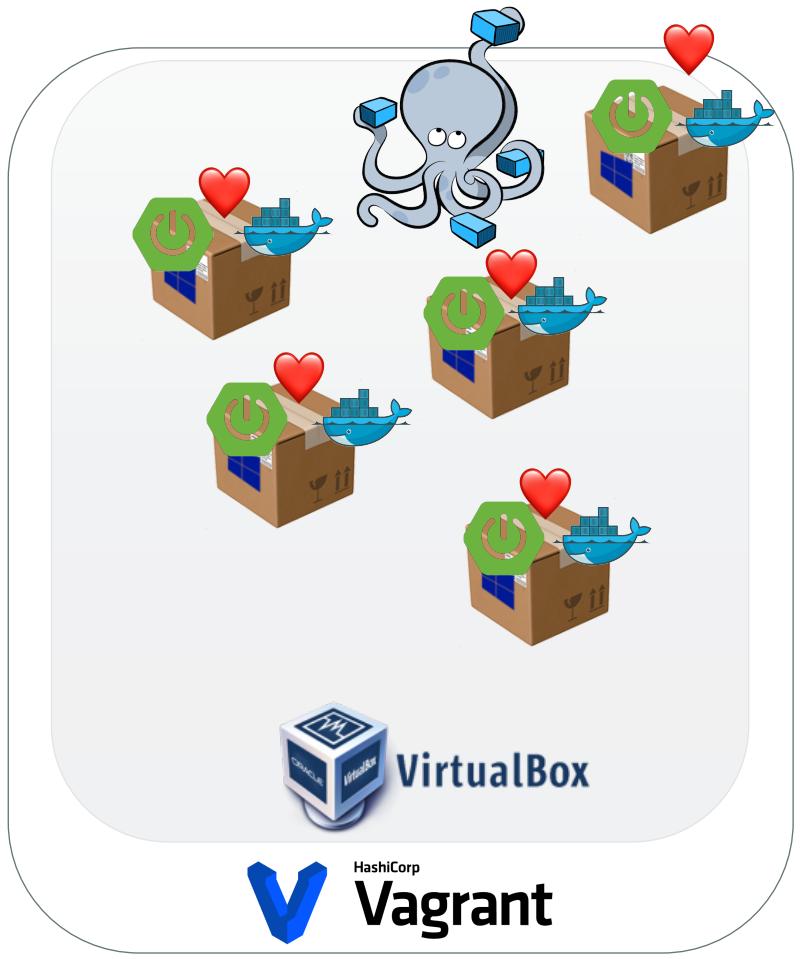


1. Preparing the App's Files for later docker-compose run
- 2. Run all Services with Docker Compose**
3. Healthcheck all Services

github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps

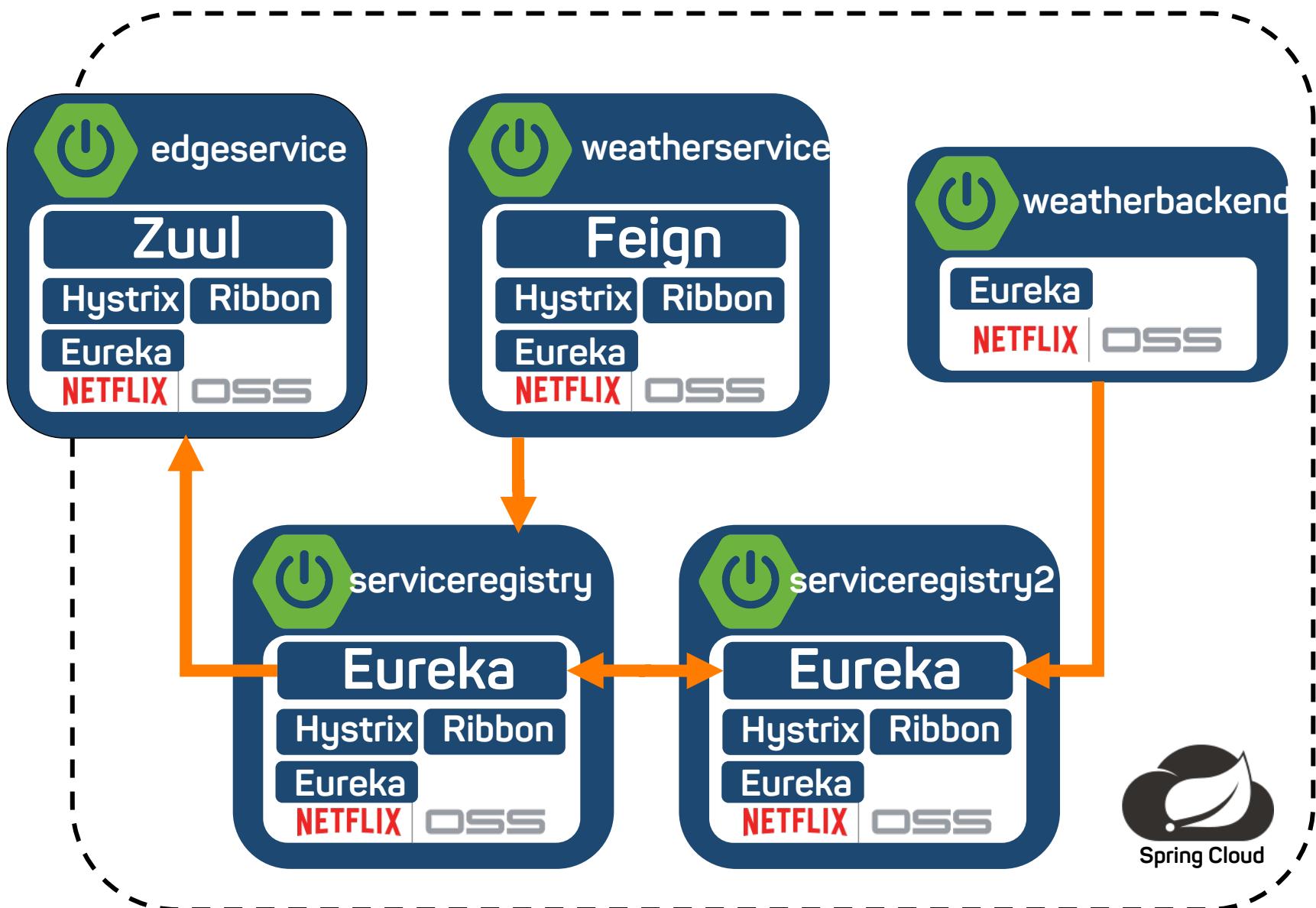


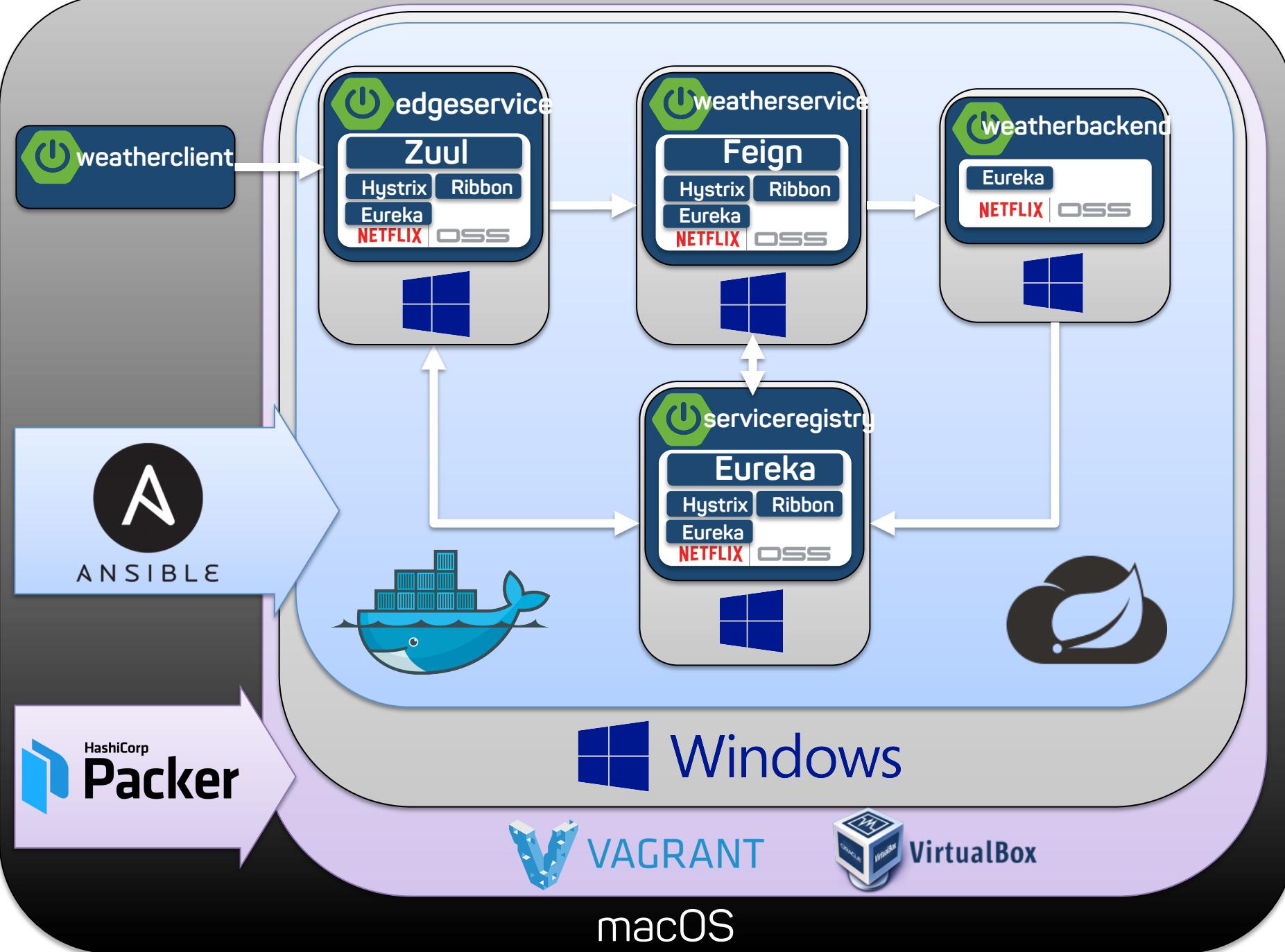
WinRM



1. Preparing the App's Files for later docker-compose run
2. Run all Services with Docker Compose
- 3. Healthcheck all Services**

github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps







Demo!

github.com/jonashackt/ansible-windows-talk#5-scale-spring-boot-apps

Vielen Dank



Sources of logos & graphics used:

emojies: <https://emojipedia.org/>

Docker Swarm <https://blog.docker.com/2014/12/announcing-docker-machine-swarm-and-compose-for-orchestrating-distributed-apps/>

Docker Compose <http://www.willhoeft-it.com/2016/06/03/docker-compose.html>

Windows

<https://upload.wikimedia.org/wikipedia/commons/c/c9/Windows-server-2016.png>

Docker <https://www.docker.com/>

Ansible <https://www.ansible.com/>

Packer <https://www.packer.io/>

Vagrant <https://www.vagrantup.com/>

VirtualBox <https://www.virtualbox.org/>

Spring Cloud <https://github.com/spring-cloud>

Spring Boot <https://projects.spring.io/spring-boot/>

Netflix OSS <https://netflix.github.io/>

Kubernetes <https://www.devopsnexus.com/consultancy-areas/containerization>

Powershell <https://blog.appliedis.com/tag/powershell/>

Difference Docker VMs

<https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-normal-virtual-machine>