



Orchester in der Wolke –
oder im Rechenzentrum –
FOLIO Systembetrieb - SysOps

Ingolf Kuss (hbz)

Deployment für Produktionsbetrieb

Deployment ~ Bereitstellung, Einsatz

Konzepte, Begrifflichkeiten

Früher: **LAMPP** (steht für Linux Apache MySQL PHP Perl)

- Manuelles Aufsetzen einer Software-Infrastruktur zur Bereitstellung dynamischer Webanwendungen

Heute: **Vagrant**

- moderne Entwicklungsumgebung für virtuelle Maschinen (VM)
- frei, quelloffen
- schnelle Erstellung von VMs auf Basis von Konfigurationsdateien, enthalten LAMPP-Stack, unkomplizierte Änderungen der Konfiguration
- Installation von Anwendungen innerhalb der „Vagrant-Boxen“
- Installation von zueinander passenden Anwendungen und Versionen
- Erstellung einer Entwicklungsumgebung, die der späteren Produktivumgebung ähnelt

Container Deployment

Container Deployment

- eine neue Variante der Virtualisierung, schlanker als virtuelle Maschinen
- Virtualisierung auf der Anwendungsebene – das spart Systemressourcen
- Container = ein Behälter mit Anwendungen, einschließlich Tools, Softwarebibliotheken und Konfigurationen. Für sich alleine lauffähig (stand-alone).
- Container isolieren Software von ihrer Umgebung – gut geeignet für den Austausch zwischen Teams, verteilte Entwicklung und Tests
- viele Mandanten nutzen denselben Container

Container Deployment - Docker

- **Docker**

- die bekannteste Container-Lösung
- ein OpenSource-Projekt
- basiert auf Linux-Techniken wie Cgroups und Namespaces, um Container zu realisieren
- kann mithilfe von Hyper-V oder VirtualBox auch unter Windows und macOS eingesetzt werden

- **Docker Image**

- ein *Schnappschuss* des Zustandes eines Containers
- ein Docker Image ist aus einer Abfolge von Schichten gebaut. Jede Schicht wird durch eine Anweisung in einem **Dockerfile**, ein Skript zum Aufbau des Containers, dargestellt.
- Mehrere Container könne sich den Zugang zu demselben Image teilen, und doch auf ihren eigenen Daten arbeiten.

- **Docker-Hub**

- offizielle, öffentliche Repositories mit Docker-Containern
- vorgefertigte Container für FOLIO-Module (für CI) auf : <https://hub.docker.com/u/folioci/>

Dockermodule

```
$ vagrant init folio/snapshot
$ vagrant up
$ vagrant ssh
$ docker ps
IMAGE                                PORTS
stripes                               0.0.0.0:3000->80/tcp
folioci/mod-inventory:7.0.1-SNAPSHOT.71 0.0.0.0:9137->9403/tcp
folioci/mod-inventory-storage:7.2.2-SNAPSHOT.74 0.0.0.0:9132->8081/tcp
folioci/mod-circulation:7.2.0-SNAPSHOT.97 0.0.0.0:9134->9801/tcp
folioci/mod-circulation-storage:4.3.0-SNAPSHOT.62 0.0.0.0:9131->8081/tcp
folioci/mod-users-bl:2.2.1-SNAPSHOT.9 0.0.0.0:9145->8081/tcp
folioci/mod-users:14.4.1-SNAPSHOT.13 0.0.0.0:9133->8081/tcp
folioci/mod-codex-ekb:0.0.5-SNAPSHOT.57 0.0.0.0:9147->8081/tcp
folioci/mod-codex-inventory:1.0.3-SNAPSHOT.29 0.0.0.0:9146->8081/tcp
folioci/mod-login:4.0.1-SNAPSHOT.9 0.0.0.0:9144->8081/tcp
folioci/mod-permissions:5.0.1-SNAPSHOT.10 0.0.0.0:9143->8081/tcp
folioci/mod-codex-mux:2.1.3-SNAPSHOT.43 0.0.0.0:9142->8081/tcp
folioci/mod-notes:2.0.2-SNAPSHOT.46 0.0.0.0:9141->8081/tcp
folioci/mod-notify:1.1.6-SNAPSHOT.37 0.0.0.0:9140->8081/tcp
folioci/mod-login-saml:1.0.2-SNAPSHOT.16 0.0.0.0:9139->8081/tcp
folioci/mod-authtoken:1.2.0-SNAPSHOT.15 0.0.0.0:9138->8081/tcp
folioci/mod-kb-ebSCO:0.1.1-SNAPSHOT.18 0.0.0.0:9136->8081/tcp
folioci/mod-configuration:4.0.1-SNAPSHOT.25 0.0.0.0:9135->8081/tcp
```

stripes = GUI aller Apps (Single Page Application, SPA)

Business-Logic-Module, z.B. für Kombination von Daten verschiedener Module

Systemmodule, das direkt auf PostgreSQL-Datenbank zugreift

Okapi nimmt Anfragen vom Benutzerbrowser und den Modulen auf Port 9130 entgegen und leitet sie weiter an 9131-9147.

Container-Orchestrierung

Für Produktivbetrieb: **Orchestrierung**

- Produktivumgebungen beinhalten viele Microservices, die jeweils in einem eigenen Container laufen.
- Container-Orchestrierung ermöglicht eine Automatisierung der Bereitstellung, Verwaltung und Skalierung von Container-basierten Anwendungen.
- Orchestrierung wird um so wichtiger, je mehr Container verwaltet werden müssen.
- Aufgaben eines Orchestrierungs-Tools:
 - Container starten und stoppen
 - Zusammenfassen von Containern (Clustering)
 - Vernetzung der Container über definierte Schnittstellen
 - Failover-Verfahren bei Ausfall eines Containers
 - **Skalierung:** Anpassung des Clusters an die aktuelle Last durch automatisches Hinzufügen oder Entfernen von Containern

Orchestrierungs-Tools

- **Kubernetes**
 - OpenSource
 - Kubernetes ist die Infrastruktur hinter Google – sehr komplex – Einrichtung und Betrieb aufwändig
 - für eine einzelne große Bibliotheken zu schwergewichtig
 - kann aber für Cloud Services z.B. in einer Verbundzentrale sinnvoll sein
- **Docker Machine, Swarm & Compose**
 - im Februar 2015 von Docker vorgestellter Software-Stack für Container-Orchestrierung
 - OpenSource
 - einfacher zu bedienen als Kubernetes
- **Apache Mesos**
 - ein OpenSource Project
 - mit Orchestrierungs-Plattform „Marathon“

Orchestrierung in der Cloud

Kommerzielles Cloud Hosting:

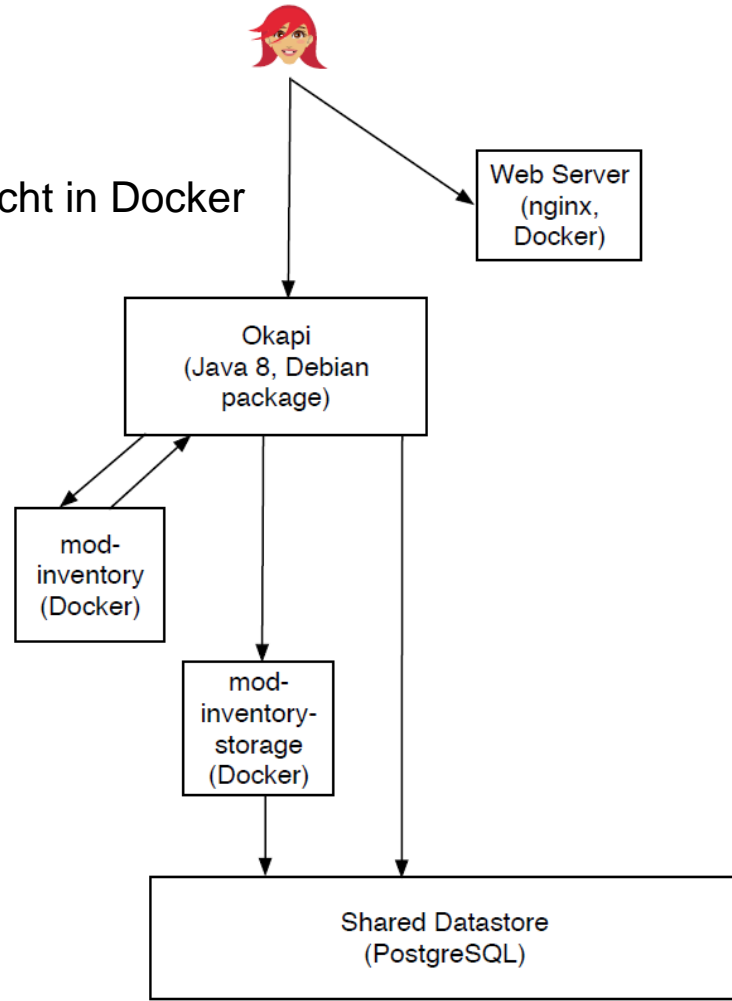
- AWS – Amazon Web Services
 - on-demand Cloud Computing Dienste von amazon.com
 - Kostengröße für unsere Anforderungen noch unbekannt ; wird von Datentransferraten abhängen, weniger von Speicher und CPU
- Microsoft Azure
 - eine Sammlung von Cloud-Diensten von Microsoft

In Deutschland eher:

- Private Cloud
- gehostet von einer Verbundzentrale oder einem Hochschulrechenzentrum

Deployment auf VM (z.B. Vagrant)

- 1 Docker Container pro FOLIO Modul
- Okapi, Webserver und Datenbank laufen nicht in Docker

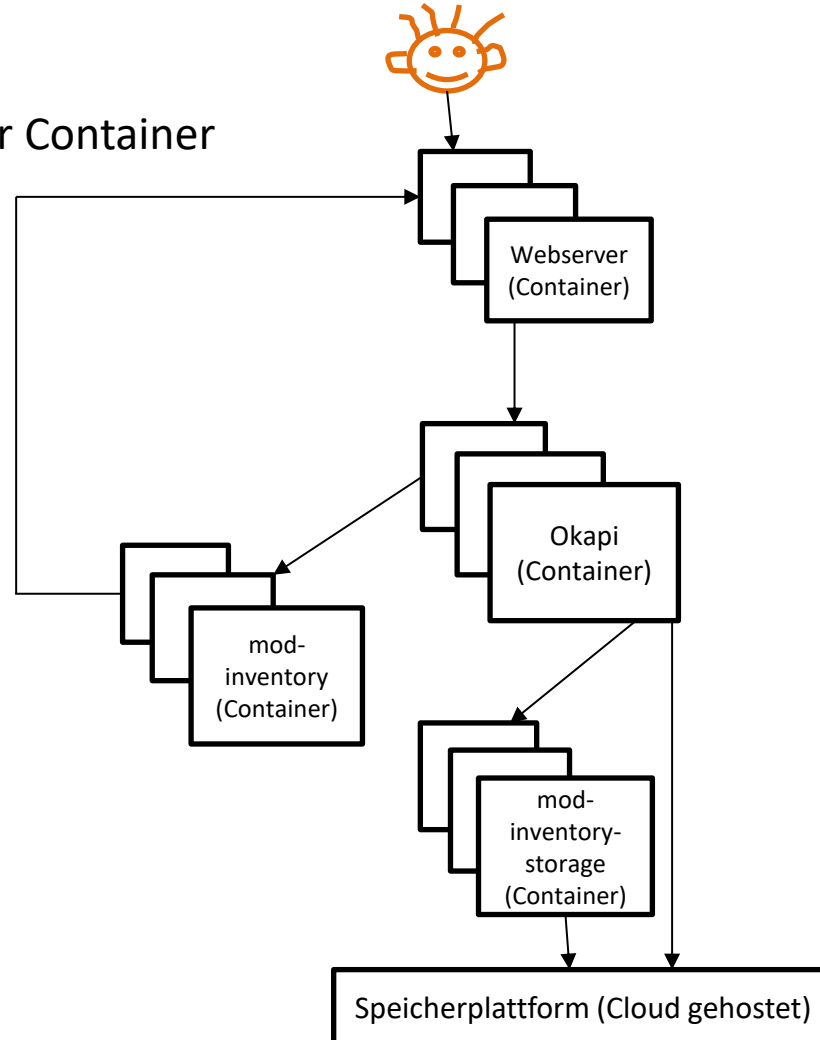


Vagrant und Docker

- Derzeit startet jedes Storage-Modul eine eigene PostgreSQL-Instanz
 - Nützlich für die Softwareentwicklung
- Jedes Storage-Modul kann aber per Parameter an eine externe PostgreSQL-Installation angebunden werden.
 - Ermöglicht Hochverfügbarkeit und Replikation durch PostgreSQL-Cluster

Cloud Deployment

Okapi gesteuerte
Orchestrierung der Container



Aktuelle FOLIO Instanzen

Aktuelle Demo – und Entwicklungsinstanzen von GBV und hbz:

<https://folio-demo.gbv.de>

<http://folio-dev.hbz-nrw.de>

- laufen mit Docker-Containern auf virtuellen Maschinen (Vagrant bzw. Virtualisierungscluster)
- Okapi als Orchestrierungstool im Einsatz

Erstanwender - Early Adopters

4 amerikanische Bibliotheken wollen im Juli 2019 Live gehen.

Texas A&M University

- z.Zt. Deployment auf einer Oracle Enterprise Linux 7 VM, auf der eine Docker Engine läuft; benutzen VMWARE integrated containers
- Deployment der Container z.Zt. manuell
- FOLIO in ungefähr 20 Docker-Containern installiert, 15 braucht man mindestens.

University of Chicago

- z.Zt. auf OLE
- erste Tests in einer Cloud-Umgebung; aber auch lokale VM möglich

Cornell University

- Deployment mit AWS und Docker geplant

Lehigh University (Pennsylvania)

Anstehende Aufgaben SysOps SIG

- Installationsleitfaden für verschiedene Deployment Umgebungen „best practices“ (lokale VM, private Cloud, gehostete Cloud)
- Datenmigration
 - Datenmapping – für die verschiedenen bisherigen Systeme
 - Entwicklung und Test von Import Loadern (FOLIO Module)
 - bisher User-Daten geladen (Chicago); Performanz muss verbessert werden (2h 28m für 90,000 User)
- Integration externer Systeme, Bereitstellung der Schnittstellen
- Datensicherheit
- Monitoring
- Hochverfügbarkeit
- Performance
- Ausfallsicherheit („disaster recovery“)

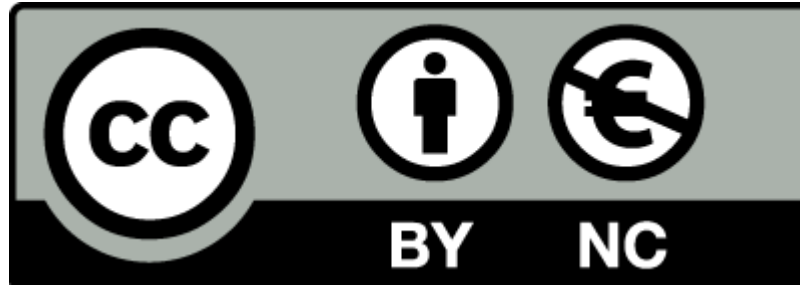
Bisherige Dokumentationen

- Installation auf einem Einzelserver:
 - <https://github.com/folio-org/folio-install/blob/master/single-server.md>
- FOLIO Ansible Playbook:
 - <https://github.com/folio-org/folio-ansible>
- CI-generierte Docker Images:
 - <https://hub.docker.com/r/folioci/>
- SysOps & Management SIG Wiki:
 - <https://wiki.folio.org/display/SYSOPS>

Vielen Dank!

Ingolf Kuss

kuss@hbz-nrw.de



Der Text dieser Präsentation wird unter der Lizenz Creative Commons Namensnennung-Nicht kommerziell 4.0 International (CC BY-NC 4.0) veröffentlicht: <https://creativecommons.org/licenses/by-nc/4.0/>

Davon ausgenommen sind die verwendeten, nicht von den Autoren erstellten Grafiken, Screenshots und Bilder, deren jeweilige Rechte und Lizenzbedingungen fortgelten.

Für die Erlaubnis zur Verwendung der Grafik auf Folie 9 geht mein Dank an Wayne Schneider, Fa. Index Data.

Maßgeblich für diese Präsentation ist das gesprochene Wort.

Online verfügbar unter ...