inovex

inovex classes

Linux

Session 3

Februar 2019

Agenda

17:00 - Recap Session 1
17:15 - Shell Scripting
18:15 - VMs, Container & Co

Recap Session 1

Recap Session 1

Aktualisierte Aufgabenstellung:

https://github.com/inovex/linux-class

Shell Scripting

Shell Scripting

Basics

Was ist ein Shell Script?

→ Eine sequentielle (*) Auflistung von Kommandos, die auszuführen sind

Shell Scripting

Basics

Weshalb Shell Scripting?

→ Automatisierung wiederkehrender Aufgaben

```
#!/bin/bash

ENVIRONMENT=test
BACKUPDIR=/opt/postgresqlmigration/
DATABASES=(dbtest1 dbtest2 dbtest3)
MYDOMAIN=class.inovex.local.invalid

echo "deleting old files"
ssh db01.${ENVIRONMENT}.dwh.fhm.de "rm -fr ${BACKUPDIR}/mig*"
# this is a comment without any sense.

for database in ${DATABASES[*]}
    do
        echo "dumping ${database}"
        ssh db01.${ENVIRONMENT}.${MYDOMAIN}  "pg_dump -C --schema-only ${database} >
${BACKUPDIR}/ddl_${database}.sql"
        ssh db01.${ENVIRONMENT}.${MYDOMAIN} "pg_dump --data-only --disable-triggers -d ${database} -C -F d
-j 24 -Z 4 -f ${BACKUPDIR}/mig_${database}"
    done
```

Shell Scripting - Let’s go :-)

Aufgabe: Benutzeranlage

Environment: Ubuntu 18.04, Bash

./benutzeranlegen.sh <Benutzername>

Regeln:

- Benutzername muss zwischen vier und 31 Zeichen lang sein
- Benutzername darf nur aus Kleinbuchstaben bestehen
- Benutzername darf keine Sonderzeichen enthalten
- Benutzername darf keine Leerzeichen enthalten
- darf nur unter uid 0 (zumeist "root") ausgeführt werden

Fallstricke:

- Kein Parameter angegeben
- Benutzername entspricht nicht den o.g. Regeln
- Benutzername existiert bereits auf dem System
- Home-Verzeichnis existiert bereits auf dem System

Shell Scripting - kleines cheat-sheet

```
if [[ ! $1 =~ ^[a-z]+$ ]];
```

```
$1 -> erster Übergabeparameter
```

```
$EUID -> effective UID
```

```
IF

-e file exists
-z Variable is set
-lt less than -> kleiner als
-gt greater than -> größer als
-ne not equal (to) -> entspricht nicht, ist nicht
```

Shell Scripting

Ein möglicher Weg (1/6)

```
#!/bin/bash

# benutzeranlegen.sh <login>
#
# inovex class: Linux, Session 3, Shell-Scripting
#
# Aufgabe: Benutzeranlage
# Environment: Ubuntu 18.04, Bash
# Scriptname: benutzeranlegen.sh
# Syntax: ./benutzeranlegen.sh <Benutzername>
```

Shell Scripting

Ein möglicher Weg (2/6)

```
# pruefen, ob das Script unter uid 0 ausgefuehrt wird

if [ "$EUID" -ne 0 ];
  then echo "Fehler: das Script kann nur unter uid 0 ausgefuehrt werden."
  exit
fi
```

Shell Scripting

Ein möglicher Weg (3/6)

```
# pruefen, ob ein Parameter angegeben wurde

if [ -z $1 ];
    then
    echo "Fehler: Kein login angegeben";
    exit 1;
fi
```

Shell Scripting

Ein möglicher Weg (3/6)

```
# pruefen, ob der Parameter zu kurz oder zu lang ist

if [ ${#1} -lt 4 ];
    then echo "Fehler: login ist zu kurz";
    exit 1;
elif [ ${#1} -gt 31 ];
    then echo "Fehler: login ist zu lang";
    exit 1;
fi
```

Shell Scripting

Ein möglicher Weg (4/6)

```
# pruefen, ob der Parameter ausschließlich aus Kleinbuchstaben besteht

if [[ ! $1 =~ ^[a-z]+$ ]];
    then echo "Fehler: login enthaelt nicht ausschließlich Kleinbuchstaben";
    exit 1;
fi
```

Shell Scripting

Ein möglicher Weg (5/6)

```
# pruefen, ob das ein entsprechendes Home-Verzeichnis bereits existiert

if [ -e /home/${1} ];
    then echo "Fehler: home-Verzeichnis (oder Datei mit identischem Namen!) existiert schon";
    exit 1;
fi
```

Shell Scripting

Ein möglicher Weg (6/6)

```
# Benutzer anlegen

useradd -m $1
```

Containers & Co

Containers & Co

Abstract

- Difference between Virtualization & Containers
- Virtualization with VirtualBox
  - Examples
- Run Containers with Docker
  - Easy-going: install Docker via official Ubuntu repository
  - The most important docker commands
  - Task: run NGINX as container, see our GIT Repository
  - Extra Task: install Docker “CE” from external repository

Containers & Co

Difference between Virtualization & Containers

Virtualization

- Nachbildung von (auch unterschiedlicher) Hardware durch einen Software Layer (Hypervisor)
  - z.B. XEN, KVM, VMWare, VirtualBox, QEMU u.a.
- Hauptgrund Kosteneinsparung: mehrere phys. Server mit geringer Auslastung können auf einem Server per Virtualisierung betrieben werden
  - Bessere Ausnutzung der teuren HW (zzgl. Strom, Stellfläche), Einsparung von freigewordener HW

Containers & Co

Difference between Virtualization & Containers

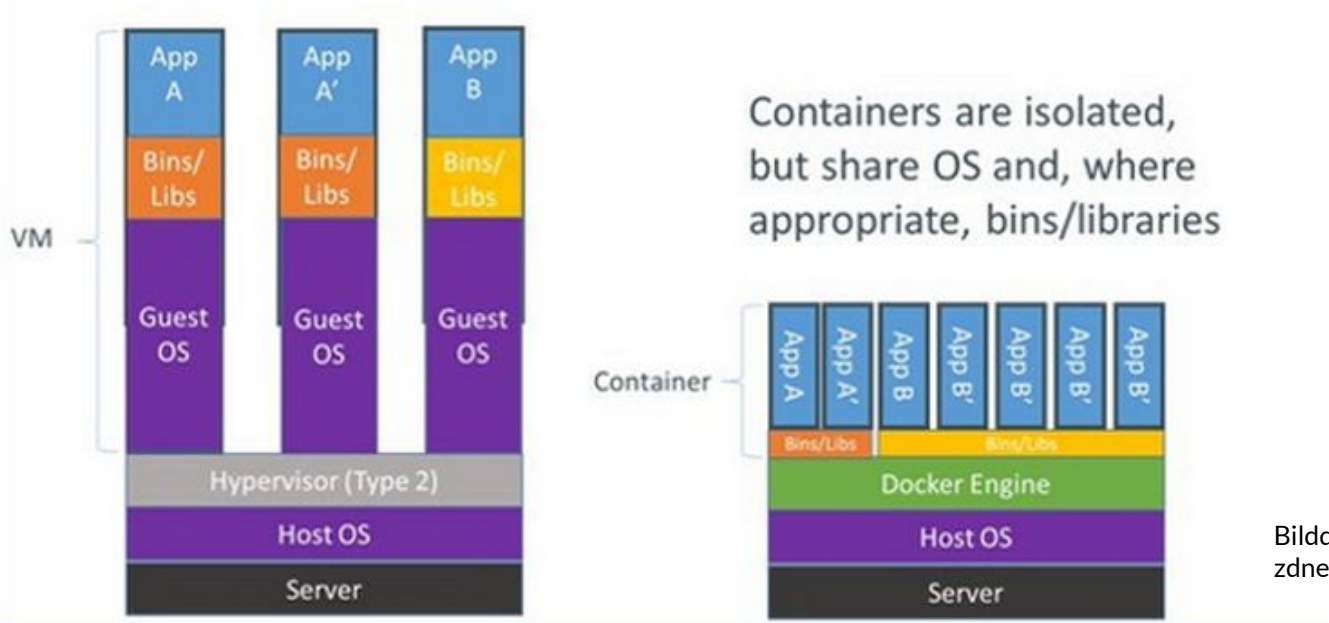
Container

- Ähnlich wie Transportcontainer für Handelsgüter verpacken Container Software für einfacheres Handling in ein Standardformat
- Container sind schlanker und verbrauchen weniger Ressourcen (CPU, Memory) als VMs
- Container nutzen gemeinsam den Linux Kernel, virtuelle Trennung durch Abschottung (Namespaces)
- 1 Application == 1 Container

Containers & Co

Difference between Virtualization & Containers

The Big Picture



Bildquelle:
zdnet.com

inovex

Containers & Co

Virtualization with VirtualBox

- Free & open source hypervisor
- runs on Linux, OSX, Windows
- initially developed by Innotek (Germany)
- mostly for Linux Systems
- but is also able to emulate Windows Systems

Containers & Co

Virtualbox Look & Feel

< see demo >

Containers & Co

For the lazy ones: Vagrant

- Developed by HashiCorp
- Let’s think of it as a wrapper for virtual machine.
- We can easily provide a VirtualBox with one Vagrant command

https://www.vagrantup.com/

Containers & Co

Vagrant handling

< see demo >

$ vagrant init ubuntu/trusty64

$ vagrant up

Hint: this is just additional stuff for the ones who want to dig deeper into virtualization.

Containers & Co

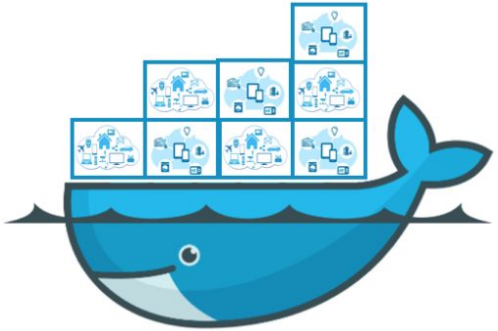
Resources

- VMs are independent
- e.g. the bring everything they need with them

https://www.vagrantup.com/

Containers & Co

Run Containers with Docker: Installation



install Docker via official Ubuntu repository

```
$ sudo apt-get update
$ sudo apt-get install -y docker.io
$ sudo systemctl enable docker
$ systemctl status docker
$ docker -v
Docker version 18.06.1-ce, build e68fc7a
```

Containers & Co

Run Containers with Docker: most important commands

The most important docker commands

- `sudo docker`
- `sudo docker <command>`
- `sudo docker <command> --help`
- `sudo docker images`
- `sudo docker search alpine`
- `sudo docker pull alpine`
- `sudo docker rmi <image-id>`

Containers & Co

Run Containers with Docker: most important commands

The most important docker commands

- `sudo docker ps -a`
- `sudo docker run -it alpine`
- `sudo docker start -ai <container-id>`
- `sudo docker attach <container-id>`
- `sudo docker stop <container-id>`
- `sudo docker rm -f <container-id>`

Containers & Co

Run Containers with Docker: run NGINX as container

Task: run NGINX (at Port 8080) as container; see our GIT Repository

Containers & Co

Run Containers with Docker: run NGINX as container

Task: run NGINX as container; see our GIT Repository

Solution:

```
$ sudo docker run --rm -d -p 8080:80 nginx
```

Containers & Co

Run Containers with Docker: Extra Task: install Docker “CE”

Extra Task: install Docker “CE” from external repository*

***) see: https://docs.docker.com/install/linux/docker-ce/ubuntu/**

Vielen Dank