

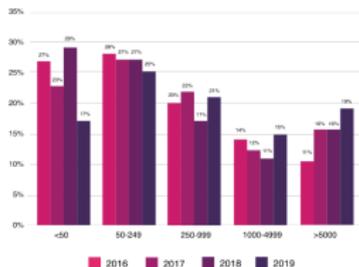
## Docker und Singularity

Daniel Lenz

Hochschule für Technik und Wissenschaft

09. März 2020

## Containerisierung im Allgemeinen



**Abbildung:** Cloud Native Computing Foundation Umfrage [1]

- ▶ in Abgrenzung zu VMs leichtgewichtiger, dafür weniger hardwarenah, aber auch eigener Applikationskontext
- ▶ Zunahme von Containerisierung klar erkennbar
- ▶ Docker ist de facto Standard, aber auch andere wie z.B. rkt und Singularity
- ▶ verschiedene Anwendungsfälle

## Gemeinsamkeiten zwischen Docker und Singularity

- ▶ **Containerisierungsansatz** → schaffen abgeschirmten Kontext für Applikationen
- ▶ **Lizenz** → Open Source [2],[3]
- ▶ **Terminologie** → Repository, Images, Container
- ▶ **Interoperabilität** → Singularity kann auch Docker Images benutzen

→ hohe Kongruenz

### Gemeinsamkeiten

```

dan@pop-os: ~/project/presentation
dan@pop-os: ~/project/presentation$ docker --version
Docker version 19.03.13, build cd8016bb0c

dan@pop-os: ~/project/presentation$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5d3b2c2d21bb: Pull complete
3fc2062ea667: Pull complete
75adf526d75b: Pull complete
Digest: sha256:b4f9e18267eb98998f6130342baacae9553f136142d40959a1b46d6401f0f2b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

dan@pop-os: ~/project/presentation$ ls
ubuntu.simg

dan@pop-os: ~/project/presentation$ sudo docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED
ubuntu              latest         4dd97cefde62   3 days a
go                  72.9MB

dan@pop-os: ~/project/presentation$

dan@pop-os: ~/project/presentation$ singularity --version
2.6.1-dist

dan@pop-os: ~/project/presentation$ singularity pull docker://ubuntu
WARNING: pull for Docker Hub is not guaranteed to produce the
WARNING: same image on repeated pull. Use Singularity Registry
WARNING: (shub://) to pull exactly equivalent images.
Docker image path: index.docker.io/library/ubuntu:latest
Cache folder set to /home/dan/.singularity/docker
Importing: base Singularity environment
Exploding layer: sha256:5d3b2c2d21bba59850dac063bcb574fddcb6aefb444ffcc63843355d878d54f.tar.gz
Exploding layer: sha256:3fc2062ea6672189447be7510fb7d5bc2ef2fda234a04b457d9dda4bba5cc635.tar.gz
Exploding layer: sha256:75adf526d75b82eb4f9981cce0b23608ebe6ab85c3e1ab2441f29b302d2f9aa8.tar.gz
Exploding layer: sha256:c6a9ef4b09995d615851d7786fbc2fe72f72321bee1a87d66919b881a0336525a.tar.gz
WARNING: Building container as an unprivileged user. If you run th
ls container as root
WARNING: it may be missing some functionality.
Building Singularity image...
Singularity container built: ./ubuntu.simg
Cleaning up...
Done. Container is at: ./ubuntu.simg

dan@pop-os: ~/project/presentation$ ls
ubuntu.simg

dan@pop-os: ~/project/presentation$

```

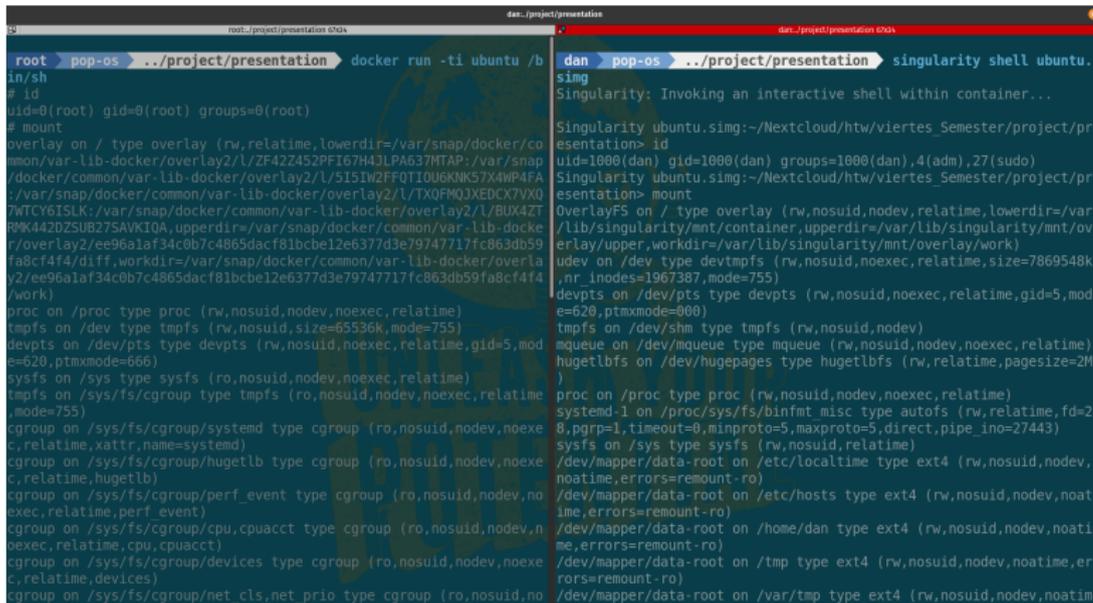
## Unterschiede zwischen Docker und Singularity

- ▶ **Dockerdemon** → Demon muss im Hintergrund laufen und erfordert root-Rechte
- ▶ **Container** → in Docker werden Images ausschließlich vom Demon verwaltet, in Singularity gibt es die .simg Dateien
- ▶ **Reproduzierbarkeit** → .simg Dateien können signiert, verifiziert, validiert werden; in Docker über docker-compose files realisiert [4]
- ▶ **Mobilität** → Singularity überträgt und speichert Container so, dass sie mit Standard Werkzeugen wie rsync oder NFS kompatibel sind [4]

## Unterschiede zwischen Docker und Singularity

- ▶ **Nutzer in Docker** → Dockernutzer ist der Nutzer, der den Container gestartet hat, d.h. der Nutzer ist festgelegt
- ▶ **Singularity Mantra** → “A user inside a Singularity container is the same user as outside the container” [5]
- ▶ **Sicherheitsmodell** → vertrauenswürdige Nutzer in vertrauenswürdigen Container im Gegensatz zu nicht vertrauenswürdige Nutzer in nicht vertrauenswürdigen Container [4]
- ▶ **Kompatibilität** → Singularity läuft auch auf älteren Kernels, die Features wie den user namespace nicht unterstützen [4]

## Unterschiede



```

root@pop-os: ~/project/presentation
root@pop-os: ~/project/presentation # docker run -ti ubuntu /bin/sh
Singularity: Invoking an interactive shell within container...
Singularity ubuntu.simg:~/Nextcloud/htw/viertes_Semester/project/presentation> id
uid=1000(dan) gid=1000(dan) groups=1000(dan),4(adm),27(sudo)
Singularity ubuntu.simg:~/Nextcloud/htw/viertes_Semester/project/presentation> mount
OverlayFS on / type overlay (rw,relatime,lowerdir=/var/snap/docker/common/var-lib-docker/overlay2/L/ZF422452PF167H4JLPA637MTAP:/var/snap/docker/common/var-lib-docker/overlay2/L/5I5IW2FFQTI0UG6KX57X4WP4FA:/var/snap/docker/common/var-lib-docker/overlay2/L/TX0FMQJXEDCX7VX0TWTCY6ISLK:/var/snap/docker/common/var-lib-docker/overlay2/L/BUX4ZTRPK442DZSUB27SAVKIQA,upperdir=/var/snap/docker/common/var-lib-docker/overlay2/ee96a1af34c0b7c4865dacf81bcbel2e6377d3e79747717fc863db59fa8cf4f4/diff,workdir=/var/snap/docker/common/var-lib-docker/overlay2/ee96a1af34c0b7c4865dacf81bcbel2e6377d3e79747717fc863db59fa8cf4f4/work)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev type tmpfs (rw,nosuid,size=65536k,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
sysfs on /sys type sysfs (ro,nosuid,nodev,noexec,relatime)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,relatime,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (ro,nosuid,nodev,noexec,relatime,xattr,name=systemd)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (ro,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/perf_event type cgroup (ro,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (ro,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/devices type cgroup (ro,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (ro,nosuid,no

dan@pop-os: ~/project/presentation
dan@pop-os: ~/project/presentation # singularity shell ubuntu.simg
Singularity: Invoking an interactive shell within container...
Singularity ubuntu.simg:~/Nextcloud/htw/viertes_Semester/project/presentation> id
uid=1000(dan) gid=1000(dan) groups=1000(dan),4(adm),27(sudo)
Singularity ubuntu.simg:~/Nextcloud/htw/viertes_Semester/project/presentation> mount
OverlayFS on / type overlay (rw,nosuid,nodev,relatime,lowerdir=/var/lib/singularity/mnt/container,upperdir=/var/lib/singularity/mnt/overlay/upper,workdir=/var/lib/singularity/mnt/overlay/work)
udev on /dev type devtmpfs (rw,nosuid,noexec,relatime,size=7869548k,_nr_inodes=1967387,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=27443)
sysfs on /sys type sysfs (rw,nosuid,relatime)
/dev/mapper/data-root on /etc/localtime type ext4 (rw,nosuid,nodev,noatime,errors=remount-ro)
/dev/mapper/data-root on /etc/hosts type ext4 (rw,nosuid,nodev,noatime,errors=remount-ro)
/dev/mapper/data-root on /home/dan type ext4 (rw,nosuid,nodev,noatime,errors=remount-ro)
/dev/mapper/data-root on /tmp type ext4 (rw,nosuid,nodev,noatime,errors=remount-ro)
/dev/mapper/data-root on /var/tmp type ext4 (rw,nosuid,nodev,noatime

```

## Unterschiede

```

dan@pop-os: ~/project/presentation$ ll
total 27076
drwxrwxr-x 3 dan dan 4096 Mar 7 18:33 ./
drwxrwxr-x 4 dan dan 4096 Mar 8 09:36 ../
-rwxrwxrwx 1 dan dan 0 Mar 7 18:11 testExtern*
-rwxr-xr-x 1 dan dan 27709471 Mar 7 17:36 ubuntu.simg*
drwxr-xr-x 5 dan dan 4096 Mar 8 09:34 vortrag/

dan@pop-os: ~/project/presentation$ ll
total 27076
drwxrwxr-x 3 dan dan 4096 Mar 7 18:33 ./
drwxrwxr-x 4 dan dan 4096 Mar 8 09:36 ../
-rwxrwxrwx 1 dan dan 0 Mar 7 18:11 testExtern*
-rwxr-xr-x 1 dan dan 27709471 Mar 7 17:36 ubuntu.simg*
drwxr-xr-x 5 dan dan 4096 Mar 8 09:34 vortrag/

```

```

Singularity ubuntu.simg:~> cd ~
Singularity ubuntu.simg:~> ls
Desktop  IdeaProjects  Postman  anaconda3  utilities
Documents  MediathekView  Public  dwhelper
Downloads  Music  Templates  eclipse-workspace
Games  Nextcloud  Videos  node_modules
Git  Pictures  Zotero  snap
Singularity ubuntu.simg:~> ls -sch
total 100K
4.0K Documents  4.0K snap  4.0K Videos  4.0K Music
4.0K Zotero  4.0K MediathekView  4.0K IdeaProjects  4.0K Public
1.20K Pictures  4.0K eclipse-workspace  4.0K Git  4.0K Templates
4.0K Downloads  4.0K anaconda3  4.0K Postman
4.0K utilities  4.0K node_modules  4.0K Desktop
4.0K Nextcloud  4.0K dwhelper  4.0K Games
Singularity ubuntu.simg:~> ll
bash: ll: command not found
Singularity ubuntu.simg:~> exit
exit

```

```

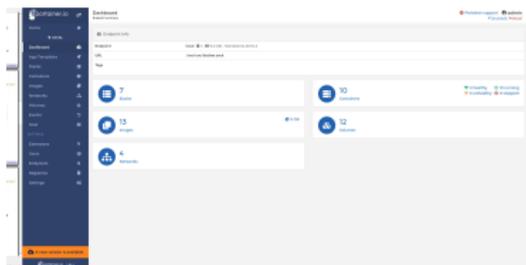
dan@pop-os: ~/project/presentation$ ll
total 27076
drwxrwxr-x 3 dan dan 4096 Mar 7 18:33 ./
drwxrwxr-x 4 dan dan 4096 Mar 8 09:36 ../
-rwxrwxrwx 1 dan dan 0 Mar 7 18:11 testExtern*
-rwxr-xr-x 1 dan dan 27709471 Mar 7 17:36 ubuntu.simg*
drwxr-xr-x 5 dan dan 4096 Mar 8 09:34 vortrag/

```

## Szenarien

- ▶ **Anwendungsfall Docker** → Service/Microservice-Architektur, mit vertrauenswürdigen Nutzern
  - ▶ **Anwendungsfall Singularity** → HPC, große Datenmengen, hohen Anzahl an Nutzern, Austausch von Containern
- Werkzeug je nach Anwendungsfall wählen

## Umgebungen



**Abbildung:** Portainer

- ▶ Docker de facto Standard → höhere Unterstützung von Umgebungen wie Kubernetes, Docker Swarm, Portainer
- ▶ durch höhere Nutzerzahlen sind Informationen zu Docker leicht zugänglich
- ▶ **aber** Docker ist eben kein “Silver Bullet”

## Zusammenfassung

- ▶ **Es gibt mehr als Docker** → Singularity für spezielle Einsatzzwecke entworfen
- ▶ **Unterschiede eher im Detail** → beide Werkzeuge haben hohes Maß an Funktionalität und Performance
- ▶ **Singularity anderes Sicherheitsmodell** → Austausch von Containern, Abstufung von Nutzerrechten, Nutzen großer Datenmengen wesentlich einfacher

## Quellen I



*CNFC Survey Report*. URL:

[https://www.cncf.io/wp-content/uploads/2020/08/CNCF\\_Survey\\_Report.pdf/](https://www.cncf.io/wp-content/uploads/2020/08/CNCF_Survey_Report.pdf/) (besucht am 06.03.2021).



*moby/moby*. GitHub. URL: <https://github.com/moby/moby> (besucht am 08.03.2021).



*License — Singularity*. URL: <https://singularity.lbl.gov/docs-license> (besucht am 08.03.2021).



*Frequently Asked Questions — Singularity*. URL: <https://singularity.lbl.gov/faq> (besucht am 05.03.2021).



*Changing Existing Containers — Singularity*. URL:

<https://singularity.lbl.gov/archive/docs/v2-3/docs-changing-containers> (besucht am 07.03.2021).

## Quellen II



Jannis Kalbermatten und Dr Henning Müller. "Container-based software distribution for high performance computing using Singularity". In: (), S. 1.



*Q04-HTW-Berlin-Logo-quer-pos-FARBIG-RGB.jpg (JPEG Image, 720 × 173 pixels)*. URL:  
<https://www.htw-berlin.de/files/Presse/Logos/Q04-HTW-Berlin-Logo-quer-pos-FARBIG-RGB.jpg>  
(besucht am 15.06.2020).