Neues vom Spielplatz: Kubernetes auf Deinem Smartphone

Dr. Christoph Zimmermann

1.7.2023 @ Tübix

cat /etc/motd

- Wie koche ich?
- Zutatenliste
- Rezept
- Es ist serviert!
- Pimp my recipe!
- Nachtisch
- Q & A

whoami

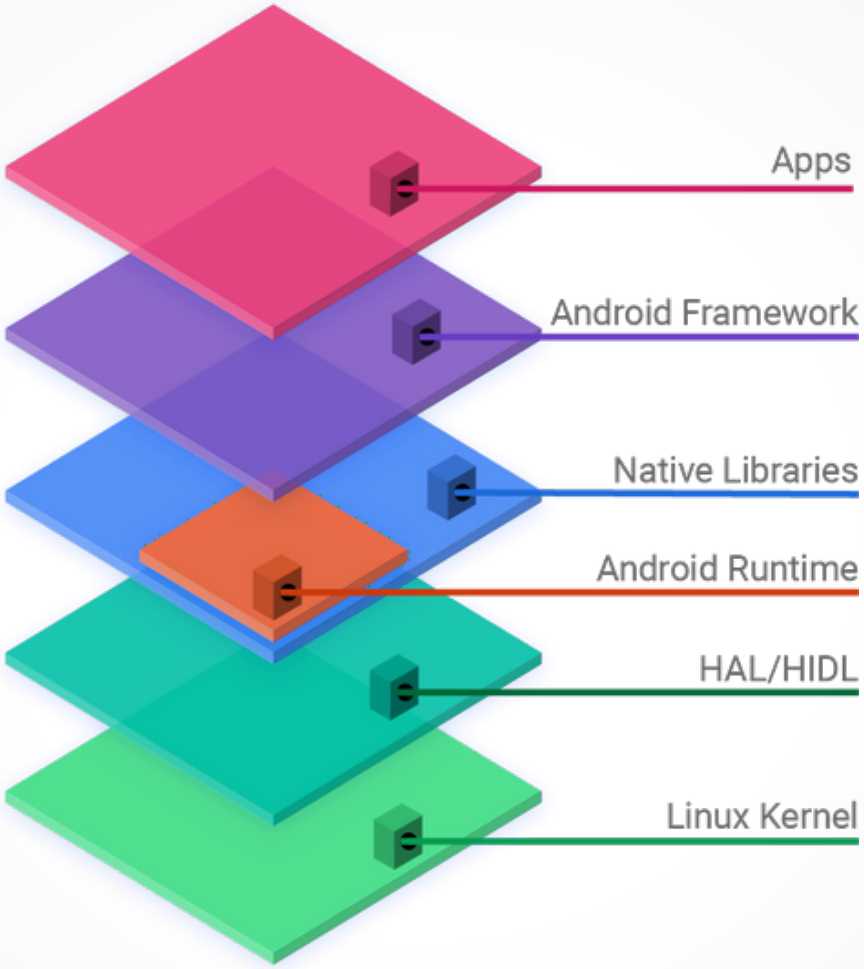
- Promotion über reflektive Betriebssysteme
- Linux: seit Kernel 0.95
- Tech Support + mehr @ FraLUG
- Podcaster
- Hobbies u.a.:
  - SDLC
  - IT Sicherheit und andere schwarze Kunst
  - Anderer Leute Computer
  - Beratung

linuxinlaws.eu

cat .android_history

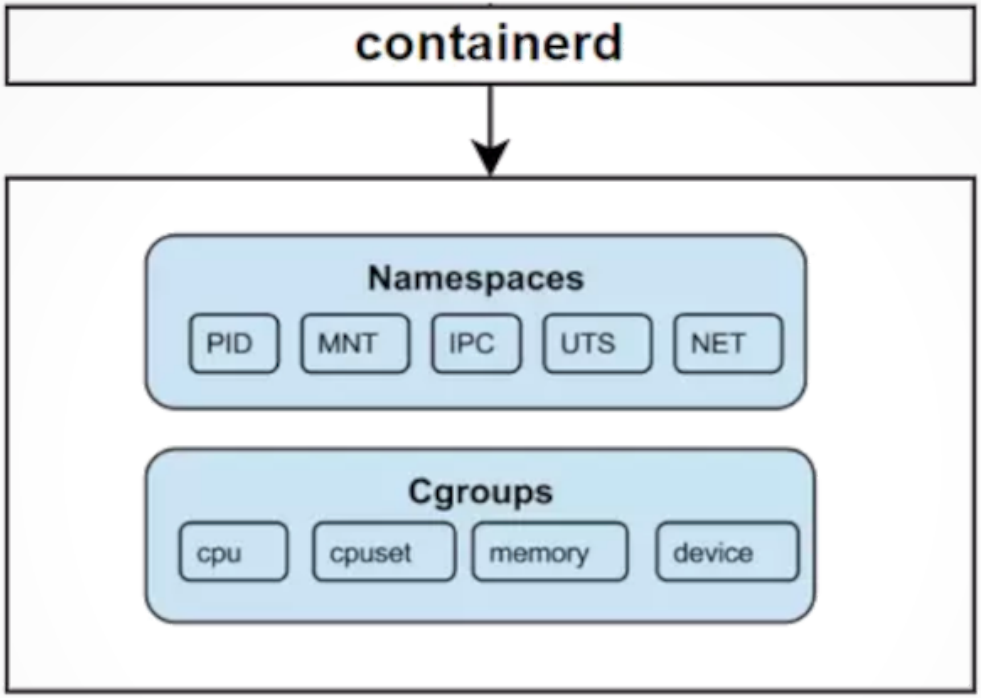
- 2003: Gründung (A. Rubin, R. Miner, N. Sears, C. White)
- 2005: Akquisition durch Google
- 2008: HTC Dream
- 2013: A. Rubin => S. Pichai
- 2019: Android Q(10) != Nachtisch
- 2019: Java => Koitlin
- 2023: Container, k8s & Android :-)

man android



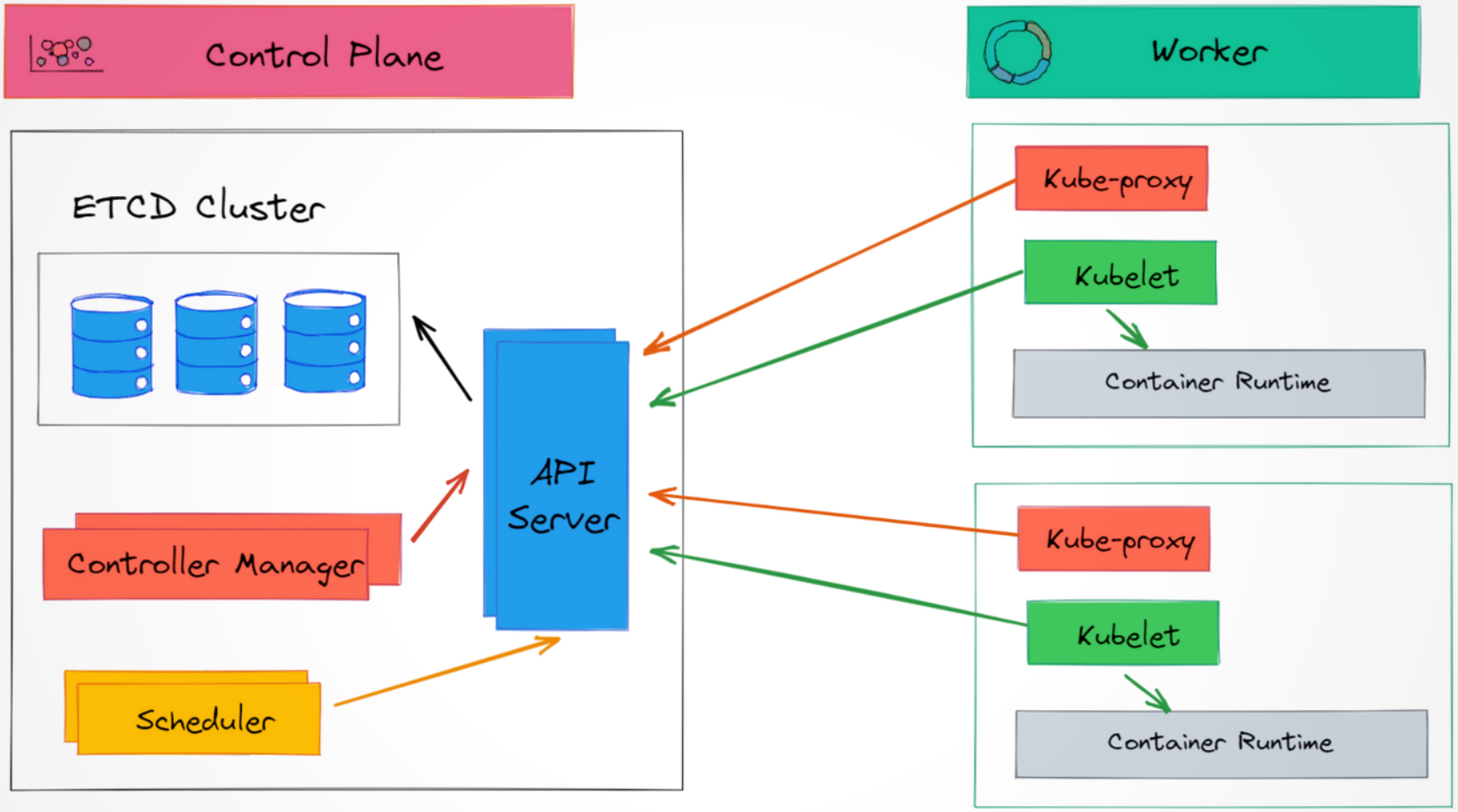
Quelle:https://en.wikipedia.org/wiki/Android_(operating_system)#/media/File:Android_open_source_project.png

man containerd



Quelle: https://viblo.asia/p/container-story-linux-namespaces-and-cgroups-what-are-containers-made-from-gDVK2r7jKLj

man k8s



Quelle:https://medium.com/devops-mojo/kubernetes-architecture-overview-introduction-to-k8s-architecture-and-understanding-k8s-cluster-components-90e11eb34ccd

cat /usr/share/ingredients

- Kernel-Konfiguration:

```
 CONFIG_CGROUPS=y
# CONFIG_CGROUP_DEBUG is not set
CONFIG_CGROUP_FREEZER=y
# CONFIG_CGROUP_PIDS is not set
# CONFIG_CGROUP_DEVICE is not set
CONFIG_CGROUP_CPUACCT=y
CONFIG_CGROUP_SCHEDTUNE=y
# CONFIG_CGROUP_PERF is not set
CONFIG_CGROUP_SCHED=y
```

```
CONFIG_BLK_CGROUP=y
# CONFIG_DEBUG_BLK_CGROUP is not set
CONFIG_CGROUP_BPF=y
CONFIG_CGROUP_WRITEBACK=y
CONFIG_SOCK_CGROUP_DATA=y
# CONFIG_NETFILTER_XT_MATCH_CGROUP is not set
# CONFIG_NET_CLS_CGROUP is not set
# CONFIG_CGROUP_NET_PRIO is not set
# CONFIG_CGROUP_NET_CLASSID is not set
```

cat /usr/share/ingredients (ff.)

- Kernel-Konfiguration:

```
CONFIG_CGROUPS=y
# CONFIG_CGROUP_DEBUG is not set
CONFIG_CGROUP_FREEZER=y
# CONFIG_CGROUP_PIDS is not set
# CONFIG_CGROUP_DEVICE is not set
CONFIG_CGROUP_CPUACCT=y
CONFIG_CGROUP_SCHEDTUNE=y
# CONFIG_CGROUP_PERF is not set
CONFIG_CGROUP_SCHED=y
```

```
CONFIG_BLK_CGROUP=y
# CONFIG_DEBUG_BLK_CGROUP is not set
CONFIG_CGROUP_BPF=y
CONFIG_CGROUP_WRITEBACK=y
CONFIG_SOCK_CGROUP_DATA=y
# CONFIG_NETFILTER_XT_MATCH_CGROUP is not set
# CONFIG_NET_CLS_CGROUP is not set
# CONFIG_CGROUP_NET_PRIO is not set
# CONFIG_CGROUP_NET_CLASSID is not set
```

cat /usr/share/ingredients (ff.)

- Kernel-Konfiguration:

```
 CONFIG_CGROUPS=y
# CONFIG_CGROUP_DEBUG is not set
CONFIG_CGROUP_FREEZER=y
# CONFIG_CGROUP_PIDS is not set
# CONFIG_CGROUP_DEVICE is not set
CONFIG_CGROUP_CPUACCT=y
CONFIG_CGROUP_SCHEDTUNE=y
# CONFIG_CGROUP_PERF is not set
CONFIG_CGROUP_SCHED=y
```

```
CONFIG_BLK_CGROUP=y
# CONFIG_DEBUG_BLK_CGROUP is not set
CONFIG_CGROUP_BPF=y
CONFIG_CGROUP_WRITEBACK=y
CONFIG_SOCK_CGROUP_DATA=y
# CONFIG_NETFILTER_XT_MATCH_CGROUP is not set
# CONFIG_NET_CLS_CGROUP is not set
# CONFIG_CGROUP_NET_PRIO is not set
# CONFIG_CGROUP_NET_CLASSID is not set
```

cat /usr/share/ingredients (ff.)

`https://termux.dev`

cat /usr/share/ingredients (ff.)

- Neueres Android (>= 11.0)
- Quad-/Octa-Core aarch64 (z. B. neuere Snapdragons)
- >= 4 GB Hauptspeicher
- >= 4 GB freier Flash o.ä.
- Linux/BSD Host

cat /usr/share/recipe

1. Android SDK (für adb)
2. Termux (mittels F-Droid installieren!)
3. (scrcpy)
4. Termux:

   `ssh` installieren & konfigurieren

   `termux-setup-storage` (falls via ssh: zuerst Grant Storage Permission!)
5. In Termux:

   ```
   pkg install qemu-utils qemu-common qemu-system-aarch64-headless wget curl bash
   ```

cat /usr/share/recipe (ff.)

6. Alpine:

   https://dl-cdn.alpinelinux.org/alpine/latest-stable/releases/aarch64/alpine-virt-*.iso

7. UEFI BIOS:

   https://releases.linaro.org/components/kernel/uefi-linaro/latest/release/qemu64/QEMU_EFI.fd

8. Virt. Festplatte anlegen:

```
qemu-img create -f qcow2 alpine.img 10G
```

9. Alpine-Installation:

```
qemu-system-aarch64 -machine virt -cpu cortex-a57 -m 2048M -smp 6 -nographic -bios QEMU_EFI.fd -drive format=raw,readonly=on,file=storage/downloads/alpine-virt-3.xx.x-aarch64.iso -drive file=alpine.img,media=disk,if=virtio -netdev user,id=n0,hostfwd=tcp::2222-:22,dns=1.1.1.1 -device virtio-net,netdev=n0
```

cat /usr/share/recipe (ff.)

10. In der VM: `/etc/udhcpc/udhcpc.conf : RESOLV_CONF="no"`
11. Zwei Nameserver konfigurieren, die UDP können:

    `/etc/resolv.conf: nameserver 8.8.8.8 \n nameserver 1.1.1.1`

12. `setup-alpine`:

    openssh: Allow root login / no key generation

    vdb Disk-Option: sys

13. VM starten:

```
qemu-system-aarch64 -machine virt -cpu cortex-a57 -m 2048M -smp 6
-nographic -bios QEMU_EFI.fd -drive file=alpine.img,media=disk,if=virtio
-netdev user,id=n0,hostfwd=tcp::2222-:22,hostfwd=tcp:8080-:8080,dns=1.1.1.1
-device virtio-net,netdev=n0
```

cat /usr/share/recipe (ff.)

14. In der VM: Community-Repo konfigurieren und updaten

```
vi /etc/apk/repositories; apk update
```

15. k3s-Abhängigkeiten installieren:

```
apk add k3s --repository=https://dl-
cdn.alpinelinux.org/alpine/edge/community
```

16. Services hinzufügen / löschen:

```
rc-update add iptables; rc-update add containerd;
rc-update del k3s
```

17. Leere iptables-Konfiguration erzeugen:

```
/etc/init.d/iptables save
```

cat /usr/share/recipe (ff.)

18. `reboot`
19. „Richtiges“ k3s installieren:

```
curl -sfL https://get.k3s.io |
INSTALL_K3S_EXEC="--disable-cloud-controller
--disable traefik –disable metrics-server" |
sh-s
```

20. Standard k3s-Binary “umbiegen”:

```
rm /usr/bin/k3s; ln -s /usr/local/bin/k3d /usr/bin
```

21. Services starten:

```
service [iptables containerd k3s] start;
```

cat /usr/share/recipe (ff.)

22. kubectl installieren:
    https://kubernetes.io/docs/tasks/tools/install-kubectl-linux
    (ARM64!)

    ```
    chmod +x kubectl; mv kubectl /usr/local/bin
    ```

23. Pre-flight Check: Laufen Container (8 an der Zahl)?

    ```
    ctr c list
    ```

24. Cluster-Authentifizierung:

    ```
    mkdir ~/.kube; cp /etc/rancher/k3/k3s.yaml ~/.kube
    ```

25. Optional: Zugriff auf Control-Plane von aussen

    ```
    # kubectl proxy -p 8080 -address=0.0.0.0
    ```

    # adb forward tcp:8080 tcp:8080

    (~/.kube/config entsprechend anspassen)

md5sum

- k8s zum Mitnehmen
- VM vs. Geschwindigkeit
- Momentan: Work in Progress
- Todo:
  - Installation / Konfiguration Use Case (s. OSCon ´23)
  - Custom Kernel mit vollständiger Container-Unterstützung?
  - Cluster mit mehreren Android-Geräten
  - Hybride k8s-Cluster (shipper/admiralty?)

cat /usr/share/doc/etc.txt

- Ursprüngliche Inspiration: https://gist.github.com/FreddieOliveira/efe850df7ff3951cb62d74bd770dce27
- Termux: https://termux.dev
- k3s: https://docs.k3s.io
- Alpine: https://docs.alpinelinux.org/user-handbook/0.1a/index.html
- Shipper: https://github.com/bookingcom/shipper
- Admiralty: https://github.com/admiraltyio/admiralty
- OSCon 23 Vortrag: https://events.opensuse.org/conferences/oSC23/program/proposals/4145

F & A

Danke!

© 2023 CC-BY

Dr. Christoph Zimmermann

monochromec at <ignore>space</ignore>gmail<dot></dot>com