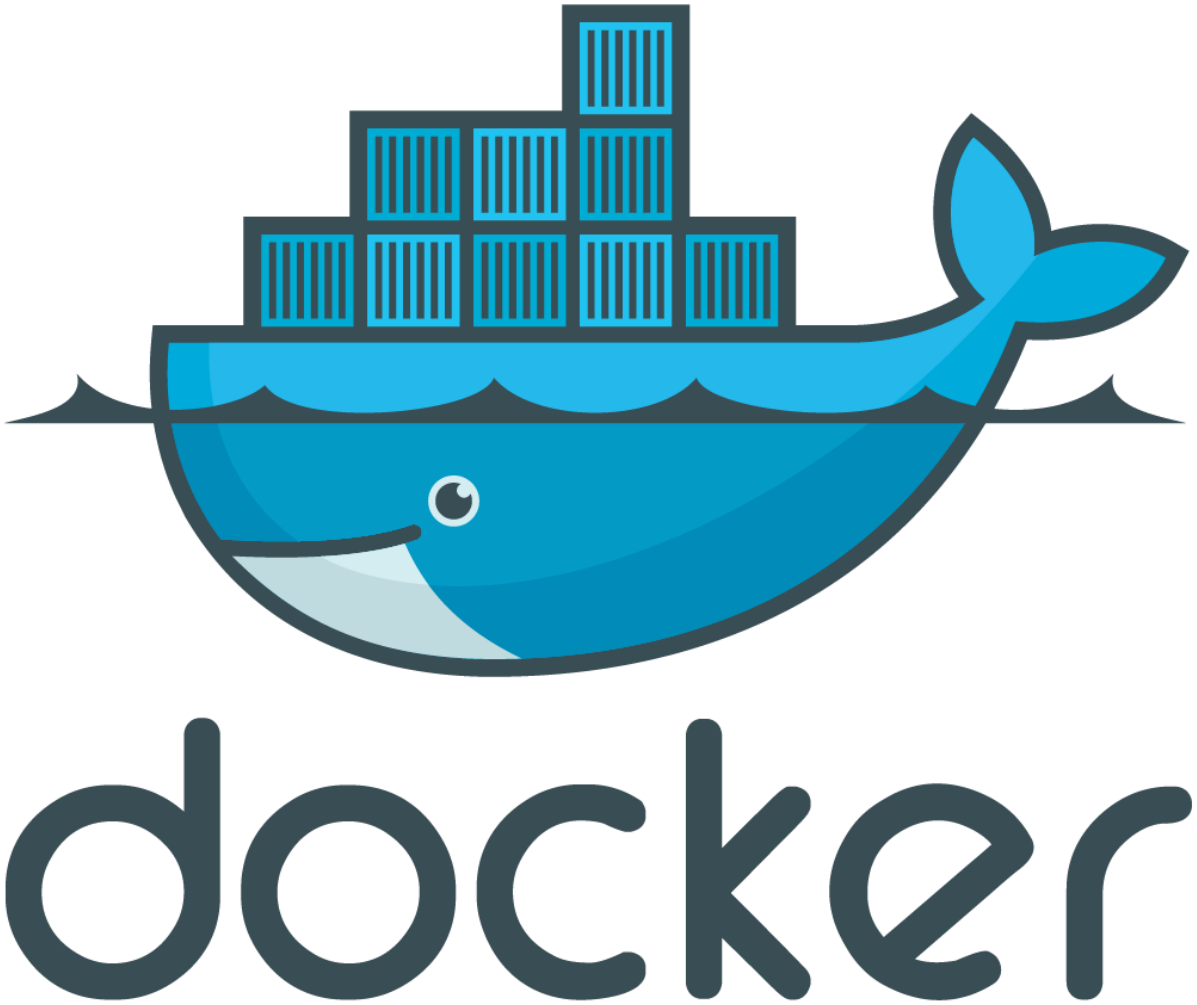
Scale out

Docker-powered Continuous Integration (mit GitLab CI)

Maximilian Florkowski
maximilian@florkowski.de

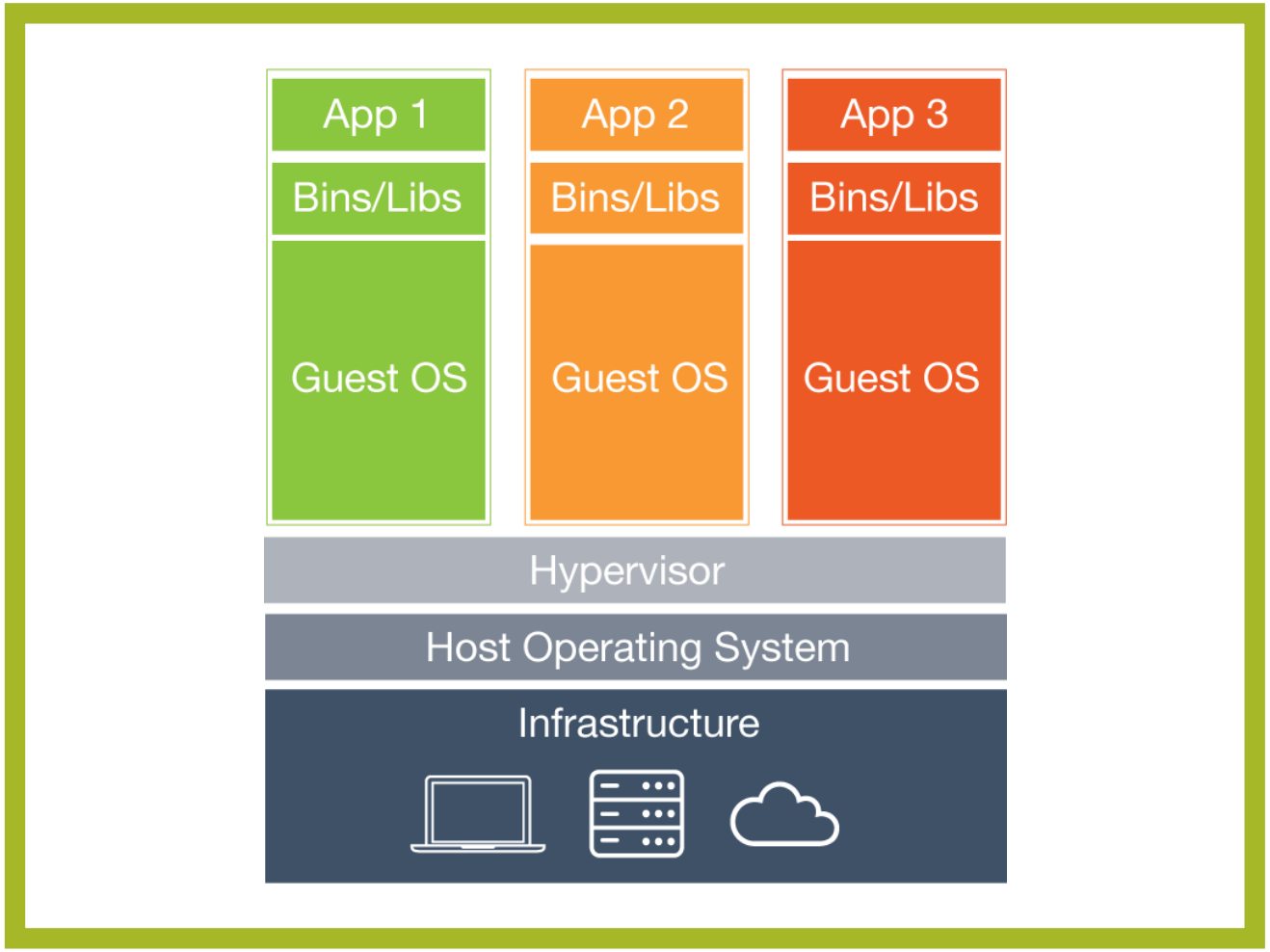
Kernthemen

- Docker
  - Was ist Docker
  - Wie funktioniert es
- GitLab CI
  - CI-Runner
  - Aufbau der Build-Jobs
  - Autoscaling


docker

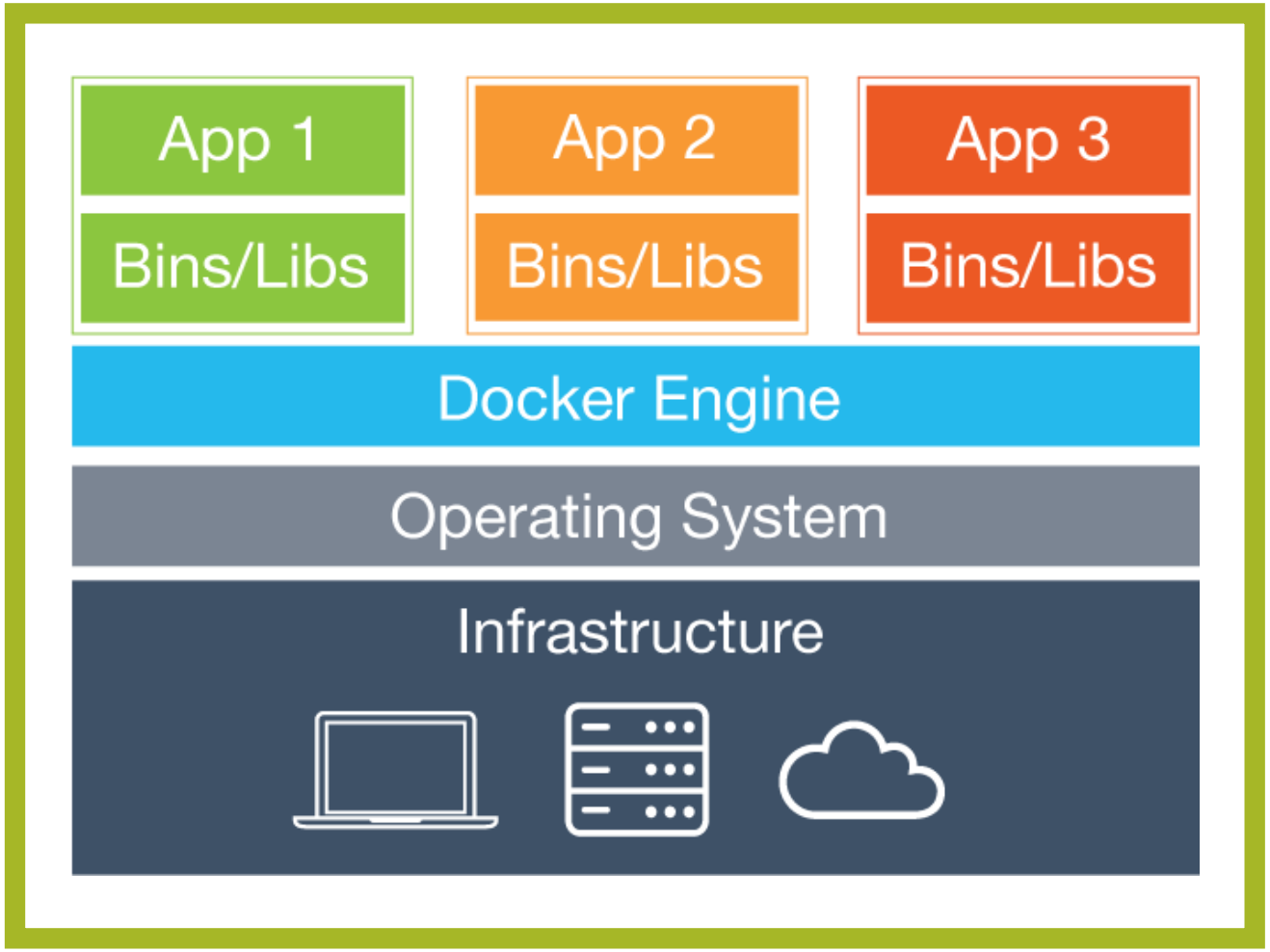
Was ist Docker

- Containerbasierte Virtualisierung
- Applikationsorientiert
- „Plattformunabhängig“
- Leichtgewichtig
- Ressourcensparend



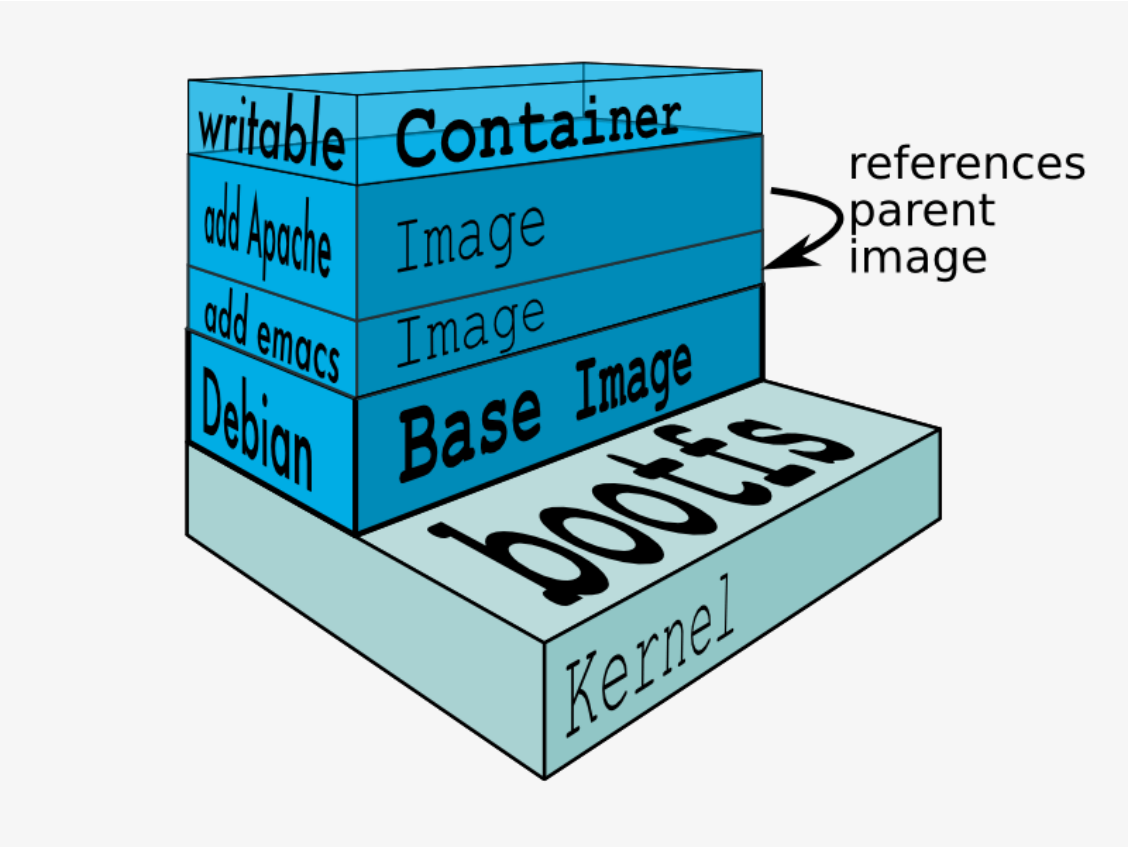
Hypervisor based Virtualization

- VMWare
- VirtualBox
- Hyper-V
- KVM
- Xen

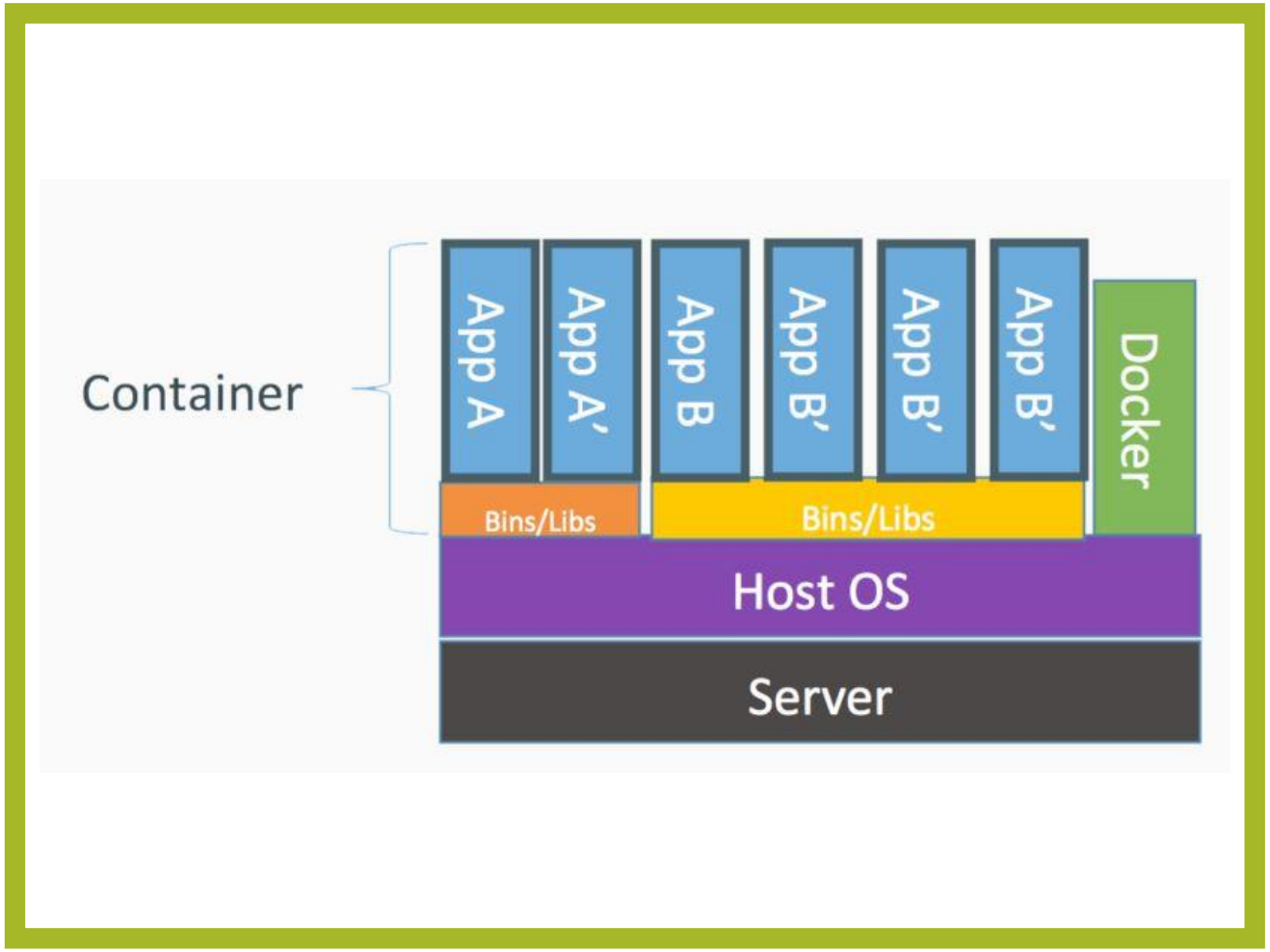


Container Virtualization

- Docker
- LXC
- Zones



Layered Filesystem



Ressourcen-teilung

Dockerfile

```
FROM debian

RUN  apt-get update \
     && apt-get --no-install-recommends install -y \
        build-essential \
     && apt-get clean
```

Kompilieren mithilfe des Containers

- `docker run --rm -v "$PWD":/usr/src/myapp -w /usr/src/myapp gcc:4.9 make`

• `docker run`	=> Startet den Container
• `--rm`	=> Container wird nach Beendigung gelöscht
• `-v "$PWD":/usr/src/myapp`	=> Mounten des aktuellen Ordners
• `-w /usr/src/myapp`	=> Setzen des Workdirs
• `gcc:4.9`	=> Das zu nutzende Image
• `make`	=> Der auszuführende Befehl


GitLab

Runner

Executor	Shell	Docker	Docker-SSH	VirtualBox	Parallels	SSH
Secure Variables	✓	✓	✓	✓	✓	✓
GitLab Runner Exec command	✓	✓	✓	no	no	no
gitlab-ci.yml: image	no	✓	✓	no	no	no
gitlab-ci.yml: services	no	✓	✓	no	no	no
gitlab-ci.yml: cache	✓	✓	no	no	no	no
gitlab-ci.yml: artifacts	✓	✓	no	no	no	no
Absolute paths: caching, artifacts	no	no	no	no	no	no
Passing artifacts between stages	✓	✓	no	no	no	no

Features

- Parallele Ausführung von Jobs
- Binärdatei ohne Abhängigkeiten
- Plattformunabhängig
- Kann Container bauen
- Kann Services starten
- Konfiguration eines Projektes via versionierter gitlab-ci.yml Datei

.gitlab-ci.yml

```
image: gcc:4.9

debug:
  script:
    - ./configure
    - make debug

release:
  script:
    - ./configure
    - make release
```

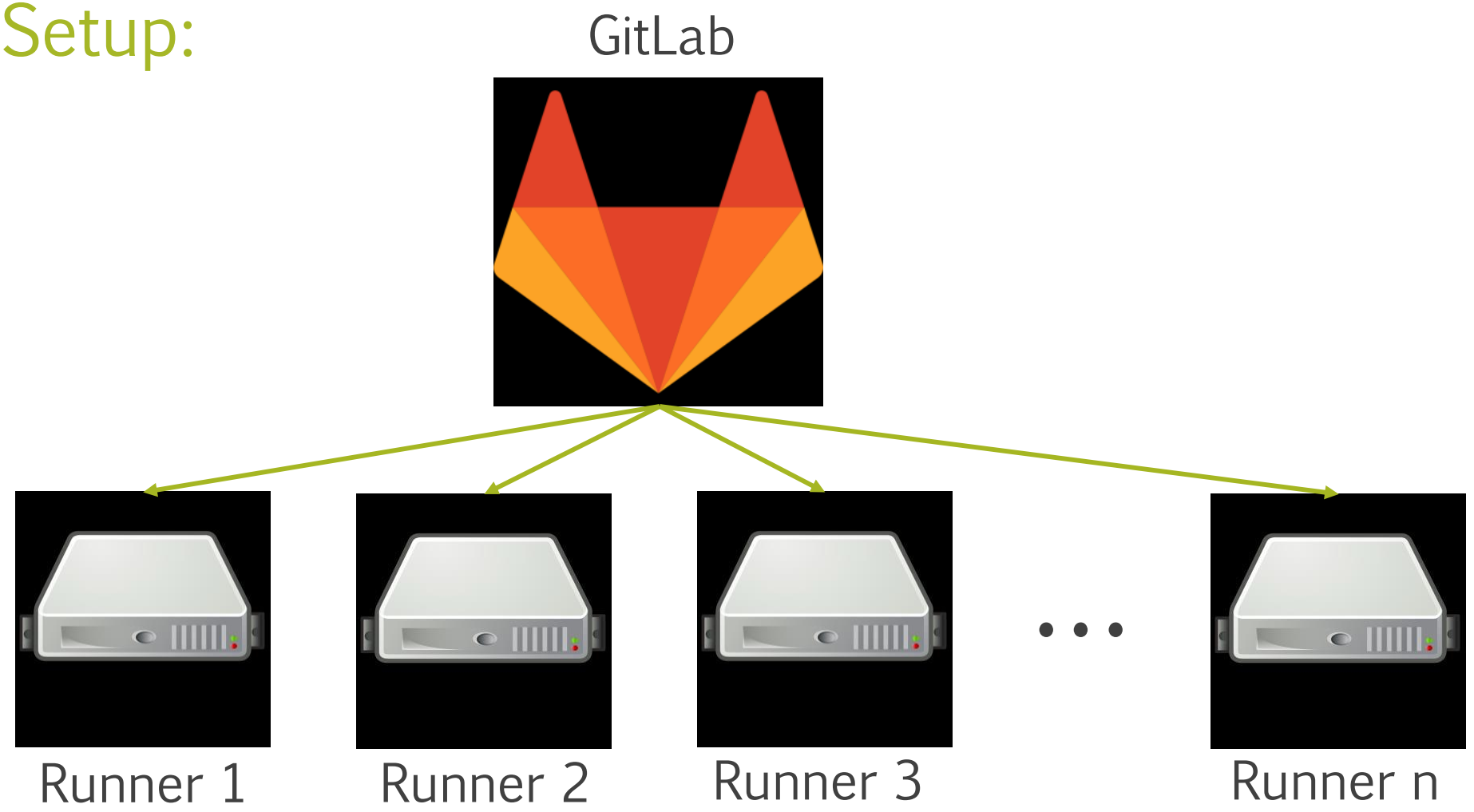
.gitlab-ci.yml

```
.job_template: &job_definition
  script:
    - ./configure
    - make

debian_8:
  <<: *job_definition
  image: debian_8


centos_7:
  <<: *job_definition
  image: centos_7
```

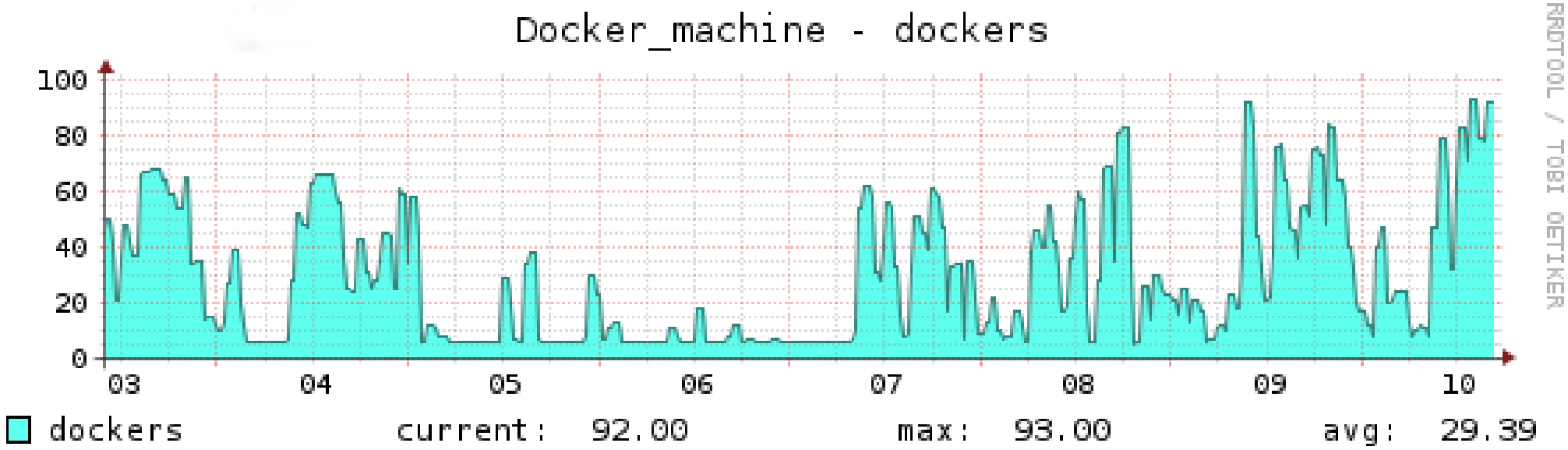
.gitlab-ci.yml

```
job1:
  image: ruby:2.1
  services:
    - postgres
    - redis
  script:
    - ./doFancyTesting
```

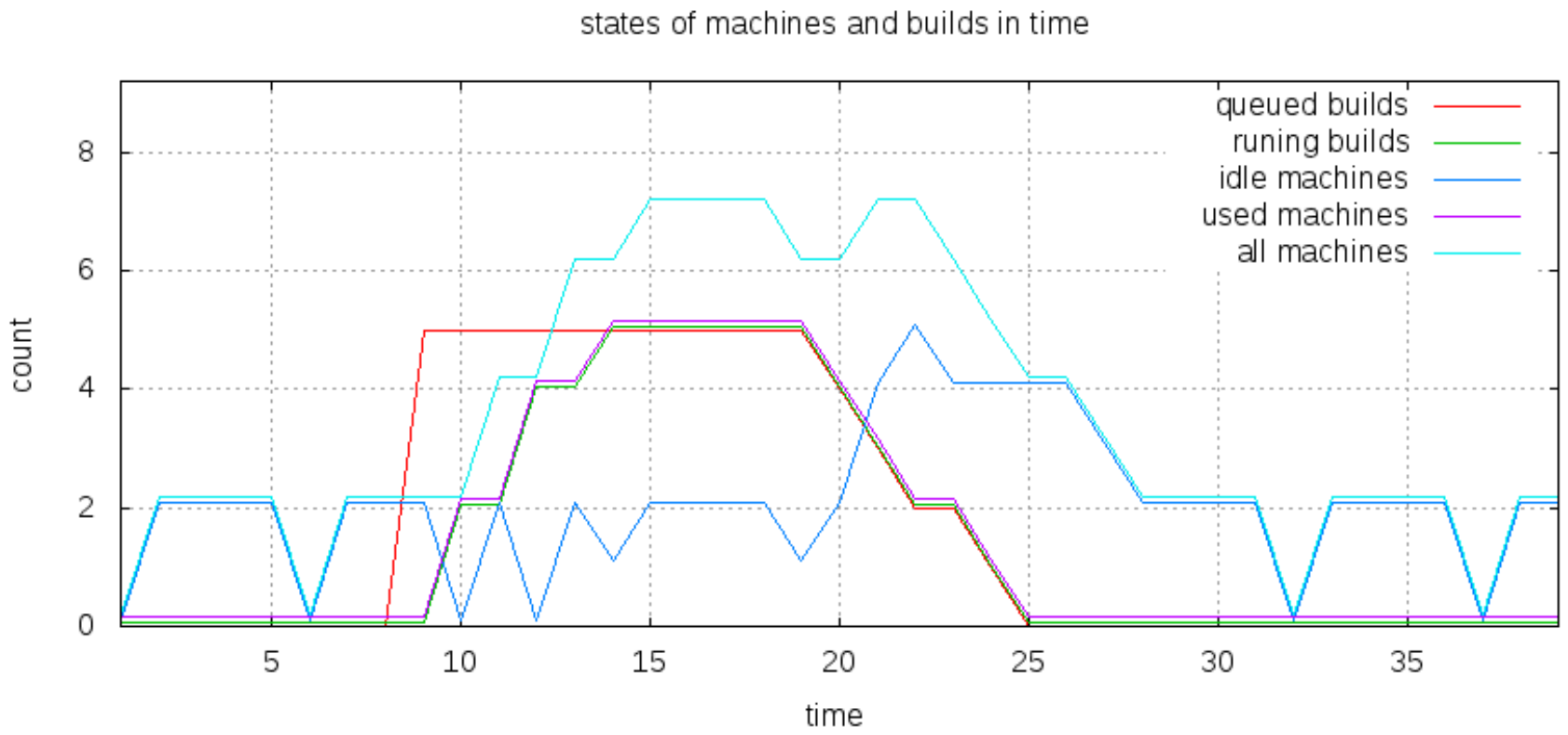
Setup:



Autoscaling



Autoscaling



Vorteile für die CI

- Keine Seiteneffekte durch vorherige oder nebenläufige Builds (Isoliert)
- Genaue Ermittlung von Abhängigkeiten möglich
- Unabhängig von dem Host-OS
- Skalierbar
- Bauumgebung „Versionierbar“
- Automatisierbarer Workflow
- Bessere Ausnutzung der Ressourcen

Links

- https://www.docker.com/
- https://hub.docker.com
- https://about.gitlab.com/

VIELEN DANK!

FRAGEN?

Maximilian Florkowski
maximilian@florkowski.de