LU13b - Monitoring

Es gibt viele Möglichkeiten, um gehostete Systeme zu überwachen, Logs auszuwerten etc.

Bei vielen Dockerumgebungen wird Kubernetes eingesetzt, welches für die meisten Punkte bereits geeignete Funktionen besitzt. Zusätzlich existiert Software wie Splunk, um immensen Datenmenge aller Systeme auszuwerten und herunterzubrechen.

Für unseren Use-Case wäre der Einsatz solcher Software Overkill, dennoch wollen wir das Monitoring nicht ignorieren.

Für Backends empfiehlt es sich einen Healthcheck-API einzubauen. Dadurch können Sie einfach testen, ob die Applikation noch läuft, ohne dass sie eine produktive (und allenfalls komplexe) API aufrufen müssen.

```
@app.route("/health")
    def health():
        return {"status": "ok"}, 200
```

```
CURL http://ec2-54-80-83-95.compute-1.amazonaws.com/api/cicd/develop/health
```

Shields.io

Github, Gitlab etc. bieten oft bereits Badges an, aber nur beschränkt. Die Webseite https://shields.io/ schafft da Abhilfe und bietet zusätzliche (statische und dynamische) Badges an.

Statische Badges Statische Badges sind im Prinzip hardcoded und dienen dazu vom Build unabhängige Informationen darzustellen. Shields.io erlaubt den Inhalt via URL-Parameter zu setzen: https://img.shields.io/badge/any%20text-you%20like-blue

Dynamische Badges Eine Auswahl dynamischer Badges finden Sie unter: https://github.com/AlexanderPeter/cicd

Es ist möglich, Status, Coverage, Versionen, Downloads etc. dynamisch abzufragen und eine entsprechende Bilddatei zu generieren. Diese Möglichkeit können wir nutzen, um einen System-Healthcheck grafisch darzustellen.

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/ffit/3-jahr/cicd/learningunits/lu13/b?rev=1779784831

Last update: **2026/05/26 10:40**