

## Installation vom Server

- req

## Installation von Openfaas

- multipass ✓
  - --disk 10GB
  - docker engine running
- arkade ✗
  - kind, kubectl, faas-cli, openfaas, docker
  - kind local registry
  - port forwarding
  - docker daemon.json: localhost:5000

## Struktur der Funktion

- Headers
- laufende Tests in einem dict gespeichert
- wie läuft ein locust test?
  - script ✗
    - exec code from string
      - nicht sicher
      - Klassenkonflikte
      - kein richtiges Importieren
  - subprocess ✓
    - Popen ✓
      - läuft im Hintergrund
      - wird terminiert
    - Call ✗
      - läuft im aktuellen Thread
      - kann nicht terminiert werden
- Tests Ergebnisse werden in Verzeichnissen auf der Festplatte gespeichert
- Commands
- wieso nutzen wir keine Openfaas Async-Funktionsaufrufe?
- sklearn und matplotlib zum erstellen von diagrammen
- wie wird die Funktion eingesetzt?
- wie wird die Funktion aufgerufen?

## Struktur der Requirements-Datei

## Struktur der Yaml-datei

## **Struktur des Servers**

- Html-Seiten
  - was sind diese Seiten?
  - was machen diese Seiten?
- proxy
  - wieso proxy?
- SSE für jeden Test. Kann darauf verzichtet werden, indem wir http-requests in einem loop benutzen: nicht effizient.
- Threads Count
- Ausführen vom Server

## **Struktur der Html-Seiten**

- Javascript, JQuery
- Tests als separate Komponente mit deren Funktionen: Sehr übersichtlich

## **Wie wird der Webservice benutzt?**

- Funktion einsetzen
- Server starten
- Benutzerinterface nutzen

## **Mögliche Probleme**

- faas-cli deploy auf multipass: Fehler 500: wird nur angezeigt, Prozess wird im Hintergrund trotzdem ausgeführt, dauert länger je größer das image.
- waitress Threads Count kann nicht ausreichen wenn mehr Tests als Threads laufen, weil jeder Test hat sein SSE, was ein Thread besitzt. Lösung: Alle Tests mit einem einzigen SSE verbinden: code weniger übersichtlich. Kann auf jeden Fall gemacht werden.
- Funktionsaufrufe außerhalb des Benutzerinterface z.B. Stats von einem nicht existierenden Test abfragen: Warum gibt es keine Sicherheitsmechanismen dafür? weil wir mehr Wert auf Performance legen.