# Webapplikation zur Steuerung eines Automatisierungsframeworks


OST Ostschweizer Fachhochschule


Felix Kubli Jérôme Gygax


Studienarbeit Herbstsemester 2020

Themengebiet Networks

Betreuer: Beat Stettler

Projektpartner: INS Institute for networked solutions

**Ausgangslage**

Der klassische Netzwerkadministrator konfiguriert seine Netzwerkgeräte manuell. In den immer grösser werdenden Netzwerken ist dies eine mühsame Aufgabe. Deshalb wurden in der Industrie Anwendungen entwickelt, welche es ermöglichen, Netzwerke automatisiert zu verwalten.

Netzwerkautomatisierungs-Lösungen bauen oft auf dem «Network as Code»-Prinzip auf. Das bedeutet, dass ein Netzwerkadministrator seine Geräte nicht wie gewohnt über eine Benutzeroberfläche oder die Shell konfiguriert, sondern Code entwickelt, welcher in seiner Logik die entsprechenden Aktionen zur Konfiguration des Netzwerks beinhaltet.

Der «klassische» Netzwerkadministrator hat häufig nicht viel mit der reinen Softwareprogrammierung zu tun, weshalb diese «Network as Code»-Lösungen eine hohe Einstiegshürde für die Administratoren in die Automatisierung darstellen.

Ziel dieser Arbeit war es, für das beliebte «Network as Code»-Tool «Nornir» eine Weboberfläche zu entwickeln, so dass auch Netzwerk-administratoren ohne Programmier-kenntnisse ihre Netzwerke mit automatisierten Aufgaben verwalten können.

**Ergebnisse**

In dieser Arbeit wurde eine Webanwendung mit React.js erstellt, welche über ein RESTful HTTP API mit dem Python Backend kommuniziert. Das Backend stellt mit der API eine Art «Wrapper» für Nornir dar.

In der Webanwendung können Geräte gesucht, gefiltert oder sortiert werden und anschliessend auf eine Auswahl ein sogenannter Task ausgeführt werden. Um in seinem Netzwerk die Downtime so gering wie möglich zu halten, können Tasks zu bestimmten Zeiten geplant werden, etwa während des offiziellen Wartungsfensters.

Das Backend stellt bereits eine kleine Auswahl an Automatisierungstasks zur Verfügung. Versierte Netzwerkadministratoren können diese um eigene Routinen erweitern.

Man stelle sich eine Firma vor, welche zehn Netzwerkadministratoren beschäftigt. Nur einer oder zwei muss dabei Python können und für die Anwendung Tasks entwickeln, aber alle des Teams können von dem enormen Effizienzgewinn durch die Automatisierung profitieren.

**Ausblick**

Gewisse Komponenten der Anwendung sind noch statisch, so muss das Inventar an Netzwerkgeräten manuell aktualisiert werden sowie die Job Templates manuell als Skript gespeichert und in der Datenbank eingetragen werden.

Als weitere Ausbaustufen könnte für Anwendung Konnektoren zu bekannten IPAM-Systemen entwickelt werden, um so stehts ein aktuelles Inventar zu besitzen.

Ebenfalls kann der Anwendung ein «Task-Builder» hinzugefügt werden, welcher es unerfahrenen Programmieren erlaubt, mit vorgefertigten «Aktions-Blöcken» neue Automatisierungstasks zu basteln, ohne dabei Python zu programmieren.

**Programmiersprachen**

- Python
- JavaScript
- JSX

**Frameworks**

- Django
- Django Rest Framework
- Material-UI
- Nornir
- React.js
- Redux