



Felix Kubli



Jérôme Gygax

Webapplikation zur Steuerung eines Automatisierungsframeworks

#	Name	Status	Scheduled	Started	Finished	Creator	Template	Abort Task	Renew Task	Detail View
11	Leaf interfaces	FINISHED	4.12.2020, 11:54:45	4.12.2020, 11:54:45	4.12.2020, 11:54:45	horbert	Get Interfaces			
10	Ping Google from Spine1	SCHEDULED	8.12.2020, 10:30:00			thomastest	Ping Device			
9	Spine1 config	FINISHED	4.12.2020, 11:10:42	4.12.2020, 11:10:44	4.12.2020, 11:10:44	thomastest	Get Configuration			
3	Get interfaces of NS lab	CREATED				horbert	Get Interfaces			

Task Dashboard mit einer Übersicht über alle erstellten, ausgeführten oder geplanten Tasks. Eigene Darstellung

Task Details

Result

Status: SUCCESS

```
spine1 / 10.20.0.201
-----
Notification: Result: netmiko_send_command
neighbor: leaf1, platform: CSR1000V, capability: 8 1, local_interface: Gig 4, neighbor_interface: Gig 2
neighbor: leaf2, platform: CSR1000V, capability: 8 1, local_interface: Gig 1, neighbor_interface: Gig 2
```

Execution Parameters

Attribute	Value
ID	14
Name	CDP neighbors Spine1
Status	FINISHED
Date Scheduled	
Date Started	4.12.2020, 12:01:11
Date Finished	4.12.2020, 12:01:12
Variables	
name	CDP neighbors Spine1

Detailsansicht des Resultat eines abgeschlossenen Tasks. Eigene Darstellung

nornir

INVENTORY | JOB TEMPLATES | TASKS DASHBOARD | PRECONFIGURED TASKS | CONFIGURATION | LOGOUT

Task Wizard

1 Select Inventory | 2 Select Template | 3 Set Variables | 4 Finish

BACK NEXT

Task Name *

Save as Preconfigured Task

Set Variables

Run Task Now

Date Scheduled: 12.12.2020

Time Scheduled: 18:41

Task Wizard, letzter Schritt vor Fertigstellung. Eingabe von Variablen, Task Name und Definition des Startdatum. Eigene Darstellung

Einleitung: Nornir ist ein Netzwerk-Automatisierungs-Tool, mit welchem Netzwerkgeräte programmatisch konfiguriert oder nach Daten abgefragt werden können. Dies verlangt Kenntnisse der Programmiersprache Python, um einen Automatisierungstask zu schreiben und anschliessend auszuführen und auswerten zu können. Netzwerkadministratoren kommen aber häufig aus der Ecke der Systemtechnik und haben entsprechend keine grossen Erfahrungen im Programmieren, was die Benutzung solch praktischer Tools wie Nornir erschwert.

Ergebnis: Mit der Nornir Webapp sollen diejenigen Netzwerkadministratoren die Vorteile von Nornir nutzen können, ohne sich mit Python auszukennen. In der Webapp können Netzwerkadministratoren Tasks starten sowie die Resultate der Tasks einsehen, ohne eine Zeile Python zu schreiben. Ein Task kann über einen geführten Wizard zusammenglickt werden und direkt oder zu einer gewünschten Zeit ausgeführt werden. Um die Details der Task-Ausführung kümmert sich das in Python geschriebene Backend, das Frontend wird mittels React Framework und der Material UI für eine moderne UI realisiert. Die Applikation ist erweiterbar ausgelegt, so dass Pythonversierte Netzwerkadministratoren die Webapp um weitere Automatisierungs-Tasks erweitert werden können. Mittels Unit-, Integration- und Usability-Tests wurde die Software getestet.

Zukünftig könnte die Applikation mit einem Skript-Builder erweitert werden, mit diesem Administratoren ihre Tasks mit vorgefertigten Code-Blöcken zusammenklicken können. So kann die Anwendung um weitere Skripts erweitert werden, ebenfalls ohne Python-Kenntnisse zu besitzen.