



David Meister



Andreas Stalder

Studenten/-innen	David Meister, Andreas Stalder
Dozenten/-innen	Prof. Beat Stettler
Co-Betreuer/-innen	
Themengebiet	Internet-Technologien und -Anwendungen

## Network Unit Testing

### Unit Testing System für Netzwerke

## Network Unit Testing



docker



SALTSTACK

ARISTA



python



```

TestCases:
Name: Ping 8.8.8.8, Command: connectivity, Devices: arch, Parameter: ['8.8.8.8'], Operator: =,
Expected: True
Name: Ping www.google.com, Command: connectivity, Devices: arch, Parameter: ['www.google.com'],
Operator: =, Expected: True

Devices:
Name: arch, OS: linux, Destination: arch, Username: -, Password: -
Name: DS1, OS: ios, Destination: DS1, Username: admin, Password: 12345678
Name: AS1, OS: ios, Destination: AS1, Username: admin, Password: 12345678
Name: arista, OS: arista, Destination: 10.0.2.205, Username: eapi, Password: password

Start Test Ping 8.8.8.8
[=====] 100%
Start Test Ping www.google.com
[=====] 100%

Ping 8.8.8.8: Test bestanden -----
Expected: True = Actual: True
Ping www.google.com: Test bestanden -----
Expected: True = Actual: True

```

**Ausgangslage:** Unit Testing ist in der Softwareentwicklung ein tägliches Werkzeug für automatisiertes White-Box Testing von einzelnen Softwaremodulen. Im Bereich der Computernetze ist ein solches Vorgehen nicht verbreitet. Häufig werden Devices «von Hand» auf Korrektheit und Funktionalität überprüft. Um zukünftig im Netzwerkkumfeld ein automatisiertes Testen zu ermöglichen, besteht die Nachfrage nach einem geeigneten Tool, welches in der Lage ist, die wichtigsten Funktionalitäten im Netzwerk zu überprüfen und auszuwerten.

**Vorgehen/Technologien:** In einem ersten Schritt wurden relevante, häufig benötigte Testfälle im Netzwerkkumfeld erarbeitet. Diese Testfälle sollen mit der entwickelten Software überprüft werden können. Damit geschriebene Unit Tests auf unterschiedlichen Devices ausgeführt werden können, wurde SaltStack als geeignetes Configuration Management Tool evaluiert. Mittels SaltStack werden Verbindungen auf unterschiedliche Netzwerkdevices über SSH oder RESTful HTTP hergestellt und gerätespezifische Kommandos ausgeführt. Um für Unit Tests notwendige Vergleiche zwischen angenommenen und tatsächlichen Resultaten durchführen zu können, müssen geeignete In- und Outputformate für ein reibungsloses Parsing gefunden werden.

**Ergebnis:** Aus dieser Arbeit ist die Software «nuts» (Network Unit Testing System) auf Basis von Python hervorgegangen. Es konnten viele relevante Testfälle im Netzwerkkumfeld umgesetzt werden. Es hat sich gezeigt, dass die vielen unterschiedlichen Outputformate der Hersteller schwierig zu Parsen sind. Häufig bleibt nur der Weg über manuelles Parsing mittels regulären Ausdrücken. So bleibt zu hoffen, dass die Hersteller zukünftig auf die Ausführung der Kommandos über RESTful HTTP setzen und deren Outputs als strukturierte JSON Objekte zur Verfügung stehen werden.

