

ANGA COM 2025 | Axians In-Booth Theater

Lab as a Service im Axians ITC

Ahmad Cheikh-Moussa, Teamleiter PS, Axians

Rüdiger Müller, Senior Consultant, Axians



[ITC Testing made easy]

„Quality is never an accident; it is always the result of intelligent effort.“

John Ruskin

[Innovation & Testing Center: Mehr als ein LAB]

Das Innovation & Testing Center der Axians ist die leistungsfähige und flexible Umgebung zur Umsetzung und Prüfung von komplexen Netzkonzepten und Proof of Concepts.

Anwendungen

ITC Überblick



DEVELOP

- + Entwicklung von Lösungen
- + Proof of Concept Tests
- + Physische und virtuelle Umgebungen
- + Aktuelle Systeme unserer Partnerhersteller verfügbar



TEST

- + Systeme
- + Software
- + Konfigurationen
 - automatisiert
 - reproduzierbar
 - dokumentiert
 - versionssicher



DEBUG

- + Nachstellen von Fehlerbildern
- + Digitaler Zwilling
- + Entwicklung und Test von Workarounds



LEARN

- + Gelerntes in realer Umgebung üben



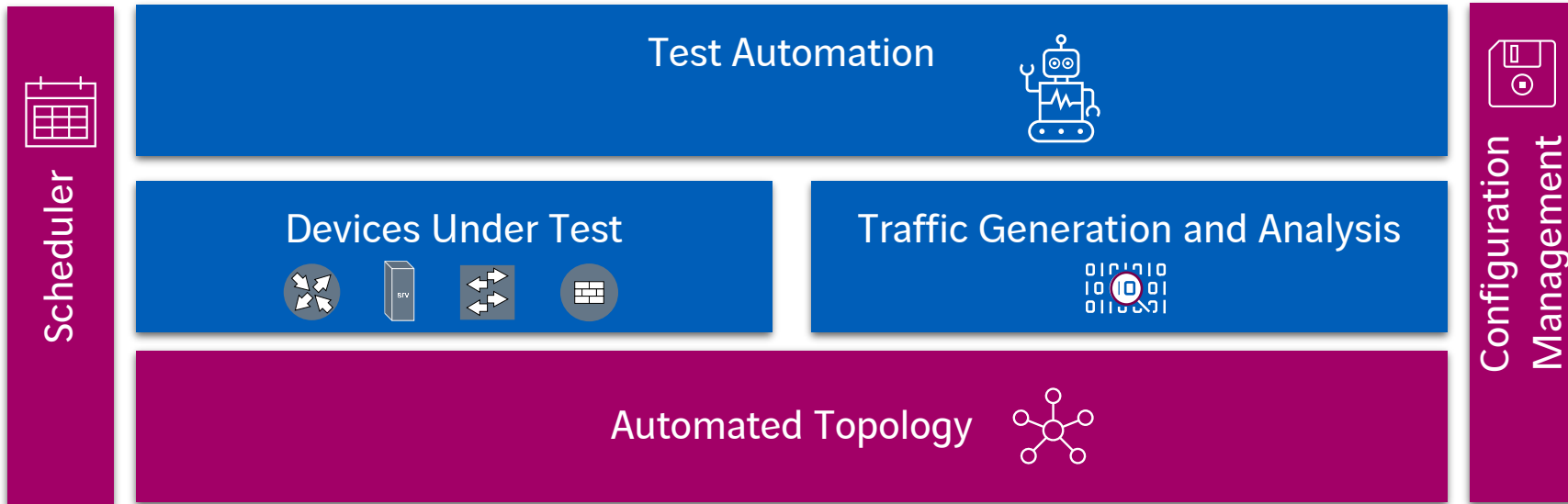
axians

[Das ITC als Basis für die effiziente Durchführung von PoCs]

Technologie, Automatisierung und Kompetenz der Axians sind die optimale Basis für die zielführende Erprobung und Verifizierung von Konzepten und Konfigurationen.

Lab as a Service Bausteine

ITC repit



Leistungen

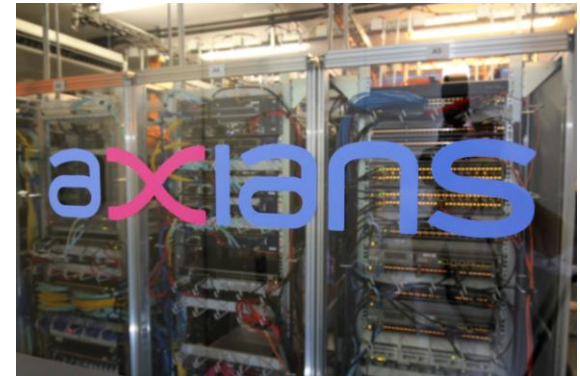
ITC-Details

LEISTUNGSFÄHIGE UMGEBUNG MIT STATE-OF-THE-ART SYSTEMTECHNIK

- Routing & Switching (Juniper, Nokia, Cisco)
- Access (Nokia, Adtran, Waystream)
- BRAS / BNG (Juniper MX, CUPS, 6Wind)
- Security (Fortinet, Juniper, Corero)
- Virtualisierung und Infrastruktur (Proxmox, VMWare, Open Source)
- WDM (Adtran, Infinera/Nokia)

LASTGENERIERUNG UND AUSWERTUNG

- Integrierte IXIA-Messtechnik (10/100GE)
- Flexible Erstellung von Testfällen und Lastprofilen
- Automatisierte Durchführung und Auswertung
- Generierung von Reports
- Flexibles Packet Capturing



Leistungen

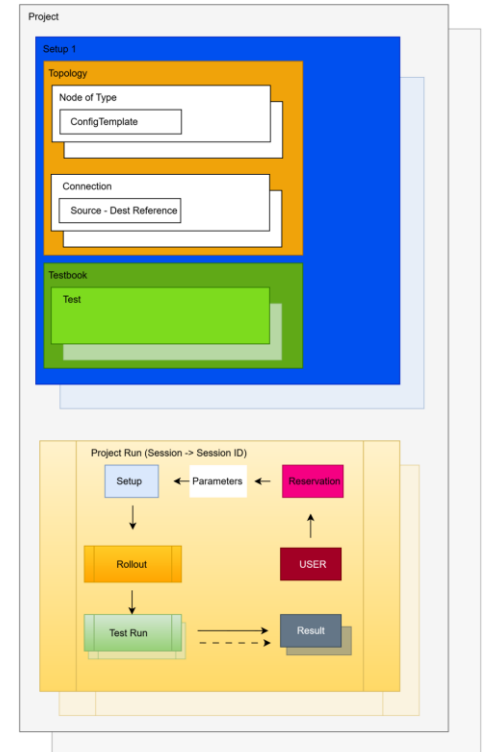
FLEXIBLES TOPOLOGIEMANAGEMENT

- Flexible Verschaltungen und Topologien auf Basis L1-Switch
- Lab as Code (LaC): Beschreibung der Umgebung in Form von yaml-Files
- Erstellen, Speichern und Laden von Setups bestehend aus Topologie und Konfiguration

VERSIONSSICHERES KONFIGURATIONSMANAGEMENT

- Zusammenfassung von Topologien, Konfigurationen und Testbooks zu Setups
- Strukturierte Verwaltung der Setups zusammen mit Testergebnissen in Projekten
- Versionssichere Verwaltung der Projekte in gitlab
- Testsessions basieren auf reproduzierbaren Setups

ITC-Details



Leistungen

ITC-Details

TESTAUTOMATISIERUNG

- Definition von Testcases und ganzen Testsuites mittels robot-Framework
- Zugriff auf bestehende Bibliotheken
- Automatisierte und wiederholte Durchführung der Tests
- Erhöhung der Reproduzierbarkeit und Qualitätssicherung
- Zeit- und Kosteneinsparung

```

1 *** Settings ***
2 Suite Setup      Open Router
3 #Suite Setup     Connect aLine=${ALIAS} host=${HOST} port=${PORT} username=${USERNAME} password=${PASSWORD} hostname=${hostname}
4 #Suite Teardown  Close Session aLine=${ALIAS}
5 Suite Teardown
6 Library          Collections
7 Library          MongoClientLibrary
8 Library          String
9 Library          SMC
10 Library          SSHLibrary
11
12 *** Variables ***
13 ${USERNAME}     lab
14 ${PASSWORD}     janeger
15 ${HOST}         ptx10000.lab.mhnetworts.com
16 ${PORT}        5000
17 ${ALIAS}        noclient
18
19 *** Keywords ***
20 Open Router
21   Open Connection    ${HOST}
22   Login              ${USERNAME}    ${PASSWORD}
23
24 GET_RT
25   [Documentation]   show metadata
26   [Output]          Execute Command    show route
27
28
29 *** Test Cases ***
30 GET data
31   GET_RT
32
33
  
```

PROFESSIONAL SERVICES SUPPORT

- Design und Konfiguration
- Definition von Testcases
- Automatisierung von Testcases
- Fehlersuche
- Alle Fragen rund um das ITC





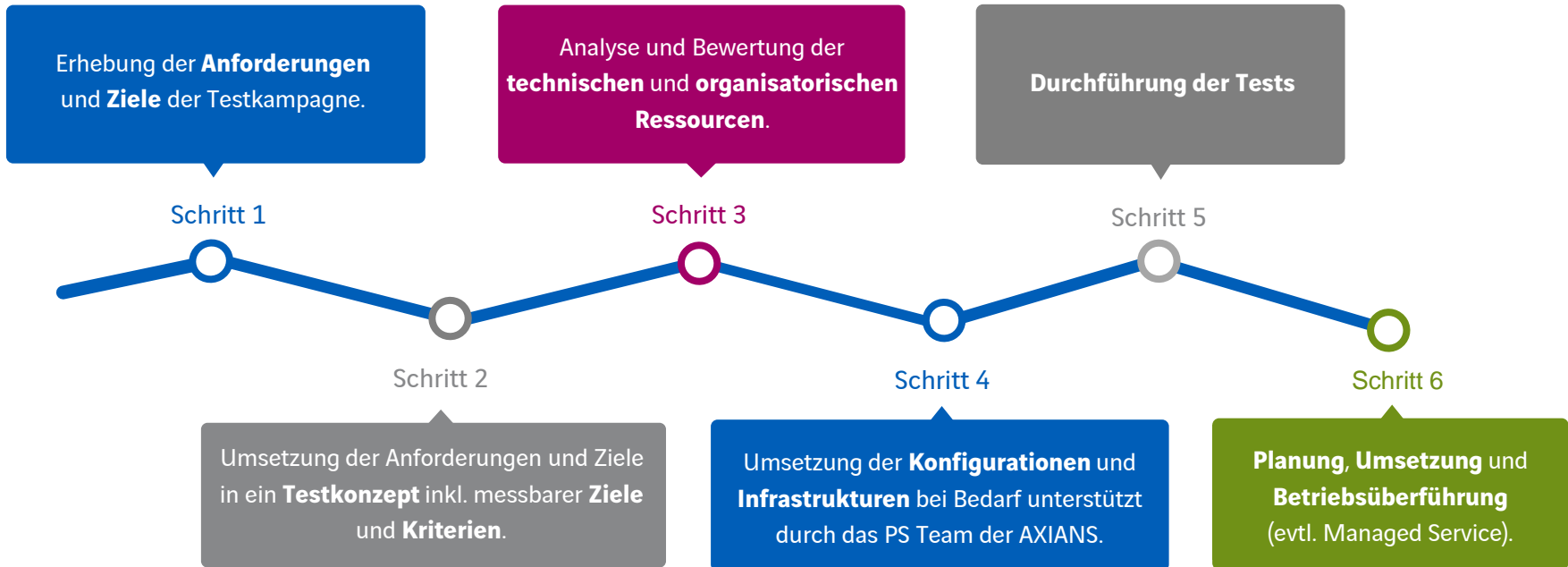
axians

Schrittweise zur erfolgreichen Testkampagne

Die Schritte einer Testkampagne

Axians ist ihr verlässlicher Partner in allen Phasen

Praxis





axians

Mit repit: im ITC eigenständig
Tests durchführen

Ein Test von A-Z

ITC-Use Case

Nach dem Login:
Ein Überblick

ITC DASHBOARD

Wort des Tages

low key – Worttrennung: low key – Herkunft: englisch low-key = unaufdringlich, zurückhaltend

Webseiten

[Netbox DEV](#)[FreelPA](#)[Booked DEV](#)[LAB GitLab](#)[Axians GitLab](#)[Power Overview](#)[LibreNMS](#)[GrayLog](#)[Storage Server](#)[Container Management](#)[ITC Dash DEV](#)

Buchungen: 2
Heute: Beispielbuchung

[Alle Buchungen anzeigen](#)

Ein Test von A-Z

ITC-Use Case

1. Schritt

Festlegung und
Reservierung der
erforderlichen
Ressourcen

Reservierungen Kalender

◀ Vorheriger Monat 6/2025 Nächster Monat ▶

Mo	Di	Mi	Do	Fr	Sa	So
26	27	28	29	30	31	1
2 ACME Network Testing 07:00 - 23:59 Gebucht von: Thomas Funke	3 ACME Network Testing Samstags Gebucht von: Thomas Funke	4 ACME Network Testing Samstags Gebucht von: Thomas Funke	5 ACME Network Testing Samstags Gebucht von: Thomas Funke	6 ACME Network Testing 00:00 - 18:00 Gebucht von: Thomas Funke	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Ein Test von A-Z

ITC-Use Case

2. Schritt Definition der Topologie

Select Source and Destination

Source: select host
Select...

Source: select port
Select...

Destination: select host
Select...

Destination: select port
Select...

Add link

Topologies

Select topology file
axms_network_RL_test.yml

Load

Deploy

Save

Save as

Network Planning View

```

graph TD
    srv7[srv7] --- beirut
    beirut --- mx1
    mx1 --- dubai
    mx1 --- dublin
  
```

Link-Table

Select all Delete selected

	link_id	source host	source port	destination host	destination port
	filter data...				
<input type="checkbox"/>	mx1_xe-0/0/1-to-dublin_xe-0/1/0	mx1	xe-0/0/1	dublin	xe-0/1/0
<input type="checkbox"/>	mx1_xe-1/0/0-to-dubai_xe-0/1/0	mx1	xe-1/0/0	dubai	xe-0/1/0
<input type="checkbox"/>	srv7_enp134s0f0-to-beirut_xe-0/1/0	srv7	enp134s0f0	beirut	xe-0/1/0
<input type="checkbox"/>	beirut_et-0/0/0-to-mx1_et-2/0/2	beirut	et-0/0/0	mx1	et-2/0/2

Successfully loaded to plan

Ein Test von A-Z

ITC-Use Case

3. Schritt
Erarbeitung der Konfigurationen

The screenshots display various network configuration commands and their outputs. Key elements include:

- show ip int brief**: Lists interfaces and their IP addresses. For example, 'ineth.0' has IP 212.0.119.2, 'ineth.1' has 212.0.119.41, 'ineth.6' has 212.0.119.114, 'l3vpn.0' has 212.0.119.119, and 'l3vpn-ineth.6' has 212.0.119.124.
- show ip int brief**: Shows a table of interfaces with columns for IP, AS, Input, Output, Flags, Last Up/Down, and State. For example, 'ineth.0' is in state 'Established'.
- show ip int brief**: Shows a table of interfaces with columns for IP, AS, Input, Output, Flags, Last Up/Down, and State. For example, 'ineth.0' is in state 'Established'.
- show ip int brief**: Shows a table of interfaces with columns for IP, AS, Input, Output, Flags, Last Up/Down, and State. For example, 'ineth.0' is in state 'Established'.
- show ip int brief**: Shows a table of interfaces with columns for IP, AS, Input, Output, Flags, Last Up/Down, and State. For example, 'ineth.0' is in state 'Established'.

Ein Test von A-Z

ITC-Use Case

4. Schritt Erarbeitung der Testcases

ROBOT
FRAME
WORK/

```
MYTestrobot 798 B
1 *** Settings ***
2 Suite Setup   Open Router
3 #Suite Setup  Connect alias=${ALIAS} host=${HOST} port=${PORT} username=${USERNAME} password=${PASSWORD} hostkey_verify=false
4 #Suite Teardown  Close Session alias=${ALIAS}
5 Suite Teardown
6 Library       Collections
7 Library       RestClientLibrary
8 Library       String
9 Library       XML
10 Library      SDSLlibrary
11
12 *** Variables ***
13 ${USERNAME}  lab
14 ${PASSWORD}  Juniper
15 ${HOST}     pts10003.lan.mnwtworks.com
16 ${PORT}    808
17 ${ALIAS}   notliant
18
19 *** Keywords ***
20 Open Router
21   Open Connection  ${HOST}
22   Login            ${USERNAME}  ${PASSWORD}
23
24 GET_RT
25   [Documentation]  www.nochocolate
26   [Output]        Execute Command  show route
27
28
29 *** Test Cases ***
30 Get RT Data
31   GET_RT
32
33
```

Ein Test von A-Z

ITC-Use Case

5. Schritt

Durchführung der
Tests und
Auswertung der
Ergebnisse

Build-Testbed-Full Report

Generated: 20240828 12:13:56 UTC+02:00
281 days 0 hours ago

Summary Information

Status: All tests passed
Documentation: Techupdate ITC Breakout ... = Main Automation =
Start Time: 20240828 12:13:10.916
End Time: 20240828 12:13:56.176
Elapsed Time: 00:00:45.260
Log File: [log.html](#)

Test Statistics

Total Statistics	Total	Pass	Fail
All Tests	1	1	0
Statistics by Tag			
No Tags			
Statistics by Suite			
Build-Testbed-Full	1	1	0

Test Details

Suite:
 Test:
 Include:
 Exclude:

[Help](#)

Build-Testbed-Full Report

Generated: 20240828 12:13:56 UTC+02:00
281 days 0 hours ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:45	
Statistics by Tag						
No Tags						
Statistics by Suite						
Build-Testbed-Full	1	1	0	0	00:00:45	

Test Execution Log

```

- [SUITE] Build-Testbed-Full 00:00:45.260
  Full Name: Build-Testbed-Full
  Documentation: Techupdate ITC Breakout ... = Main Automation =
  Source: app/robot/build-testbed-full.robot
  Start / End / Elapsed: 20240828 12:13:10.916 / 20240828 12:13:56.176 / 00:00:45.260
  Status: 1 test total, 1 passed, 0 failed, 0 skipped

- [SUITE] Update hosts 00:00:00.000
  Start / End / Elapsed: 20240828 12:13:11.308 / 20240828 12:13:11.308 / 00:00:00.000
  Documentation: Runs the given command in the system and returns the output.
  Start / End / Elapsed: 20240828 12:13:11.301 / 20240828 12:13:11.307 / 00:00:00.000
  12:13:11.305 [INFO] Running command "cat app/robot/req/hosts >> /etc/hosts".

- [SUITE] Build Topology 00:00:44.866
  Full Name: Build-Testbed-Full:Build Topology
  Documentation: Benötigte Verschaltungen zwischen den DUT werden hergestellt für alle Gruppen. Die Links für Infos sind in grpd
  Start / End / Elapsed: 20240828 12:13:11.308 / 20240828 12:13:56.174 / 00:00:44.866
  Status: [SUITE]

- [SUITE] Jenkins load Case topology-grp1-full 00:00:14.999
- [SUITE] Jenkins load Case topology-grp2-full 00:00:14.128
  Start / End / Elapsed: 20240828 12:13:26.310 / 20240828 12:13:40.438 / 00:00:14.128
- [SUITE] $[job_output] = os.system.Run python3 app/robot/Library/Use-jenkins.py -j pluriBus_SigType[ case-p-$[topology].yaml
  Documentation: Runs the given command in the system and returns the output.
  Start / End / Elapsed: 20240828 12:13:26.311 / 20240828 12:13:40.435 / 00:00:14.124
  12:13:26.314 [INFO] Running command "python3 app/robot/Library/Use-jenkins.py -j pluriBus_load_case -p topology-grp2-full.yml 2>&1".
  12:13:40.435 [INFO] $[job_output] = pluriBus_load_case Job is Running...
  Let ID 228, Current ID 228
  Started by remote host 172.18.24.121
  [Pipeline] Start of Pipeline
  [Pipeline] node
  Running on Jenkins in /var/jenkins_home/...

+ [SUITE] os.system.Run Log $[job_output] 00:00:00.001
+ [SUITE] os.system.Run $[job_output] Finished: SUCCESS 00:00:00.000
+ [SUITE] Jenkins load Case topology-grp3-full 00:00:15.735
    
```

Zusammenfassung

Zusammenfassung

Testkampagnen und PoCs können im ITC der Axians effizient umgesetzt werden

Kosten für den Aufbau und den Betrieb eines On-Premise-Lab werden vermieden

Zusätzliche Systeme und Ressourcen können flexibel eingebunden werden

Umsetzung komplexer Szenarien erfolgt gemeinsam mit dem Professional Service

Kommunikation zwischen Kunde, Hersteller und Integrator wird vereinfacht

**Vielen Dank
für Ihren Besuch!**

**Wie hat Ihnen der
Vortrag gefallen?**

Titel: Lab as a Service im Axians
ITC Referent: Ahmad Cheikh-
Moussa, Axians



Jetzt abstimmen