



Akademie für
Lehrerfortbildung
und Personalführung



Cisco Virtual Partner Technical eXchange

NETZWERK AUTOMATISIERUNG IN DER PRAXIS

Timm Wächter – Solution Manager



Computacenter

20. Nationaler Akademetag der Bildungsinitiative Networking
22./23. April 2021

#CiscoPTX20

AGENDA

- Vorstellung
- Netzwerk Automatisierung
- Einführung in Ansible
- Praxisbeispiel
- Netzwerk Automatisierungs Demo



2005 Allgemeine Hochschulreife

- CCNA R&S

2008 Studium der Elektro und Informationstechnik

- CCNA R&S erneuert

2014 Einstieg bei Computacenter als Trainee

- CCNP R&S
- CCNA DC

2019 Solution Manager Networking

- DevNet Specialist



Timm Wächter

Solution Manager
Networking

Computacenter AG & Co. oHG
Mainzer Landstraße 178-180, 60326 Frankfurt
Email: tim.waechter@computacenter.com
www.computacenter.de



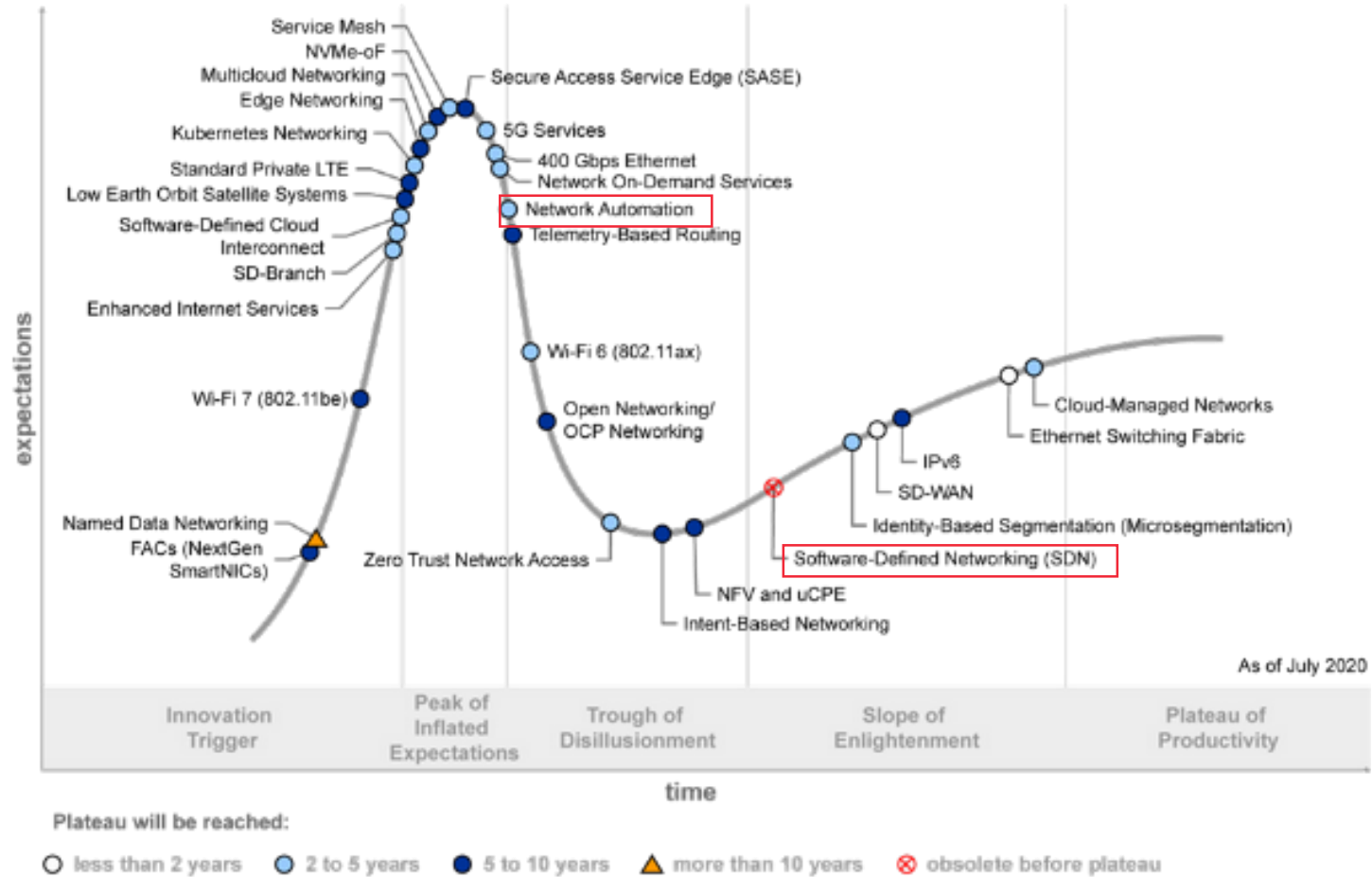
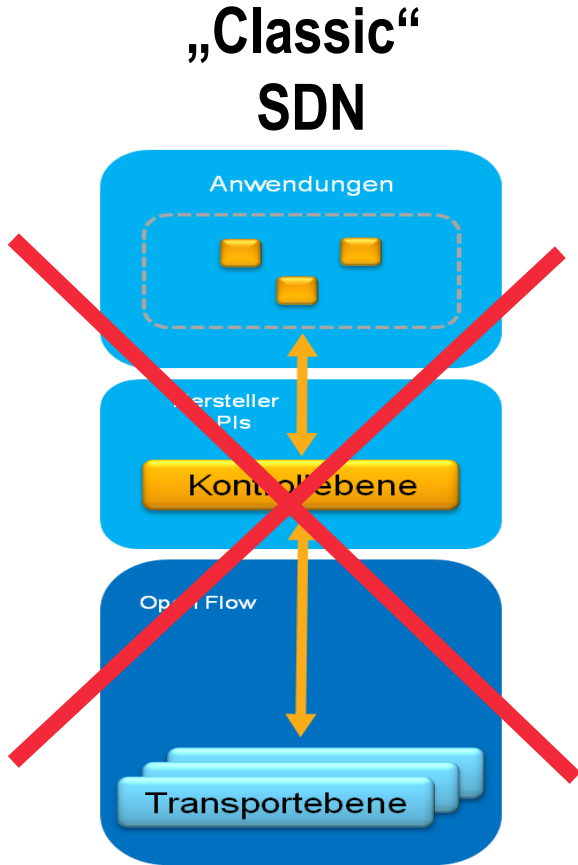
NETZWERK AUTOMATISIERUNG

Trends & Innovation



SOFTWARE-DEFINED NETWORK

Hype Cycle for Enterprise Networking, 2020



Source: Gartner
ID: 441509



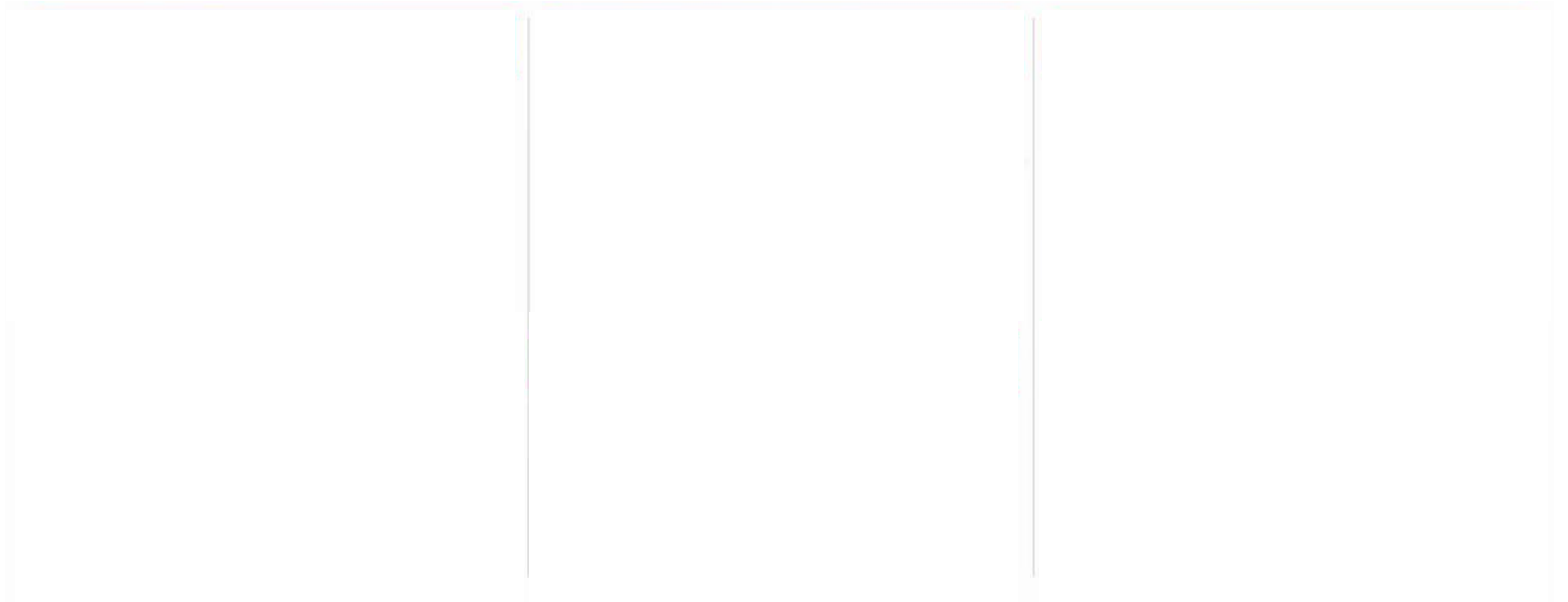
TRENDS & INNOVATION

NETZWERK AUTOMATISIERUNG



NETZWERK AUTOMATISIERUNG

VORTEILE



NETZWERK AUTOMATISIERUNG

WAS HAT SICH GEÄNDERT

Früher

- Bash Skripte, vorgeschriebene Text-files oder Templates
- Individuell, „Selbstgebastelte“ Skripte aus der „not“ heraus
- Kommunikation über Command-line (cmd) per SSH oder proprietäre Schnittstellen
- Dezentral, jedes Netzwerkgerät einzeln
- Kaum Interaktion, Betrachtung anderer Bereiche
- Text-files oder Excellisten

Heute

- Professionelle Automatisierungs Software, Tools oder Frameworks
- Gemeinsam entwickelte Skripte, nach Firmen Standards
- Standardisierte offene Schnittstellen (REST)
- Zentralisiert & Versioniert
- Übergreifende Interaktion mit anderen Bereichen (Orchestrierung)
- Datenbanken



TRENDS & INNOVATION

RESTFUL API

Fast jedes aktuelle Produkt im Netzwerk besitzt eine RESTful API

- **R**epresentational **S**tate Transfer **A**pplication **P**rogrammer Interface (REST API)
- Die REST-API erlaubt externen Clients **CURD-** (**C**reate, **R**ead, **U**ppdate, **D**elete)
- Standard Methode, bekannt von Internet (Formular-) Daten-Transfer zwischen Programms
- Gedacht für Maschine zu Maschine Kommunikation
- Zustandslos
- Die API benutzt XML oder JSON Syntax
- Syntaxcheck vor der Ausführung: „all or nothing“.
- HTTP Fehler Behandlung



Welche Cisco Geräte bieten eine REST API:

- ACI Controller (APIC)
- DNA Center
- IOS XE
- NXOS Switche
- NSO
- UCS Manager / Hyperflex
- CloudCenter
- ...



NETZWERK AUTOMATISIERUNG

```
# list of packages that should be imported for this code to work
import cobra.mit.access
import cobra.mit.naming
import cobra.mit.request
import cobra.mit.session
import cobra.model.infra
from cobra.internal.codec.xmlcodec import toXMLStr

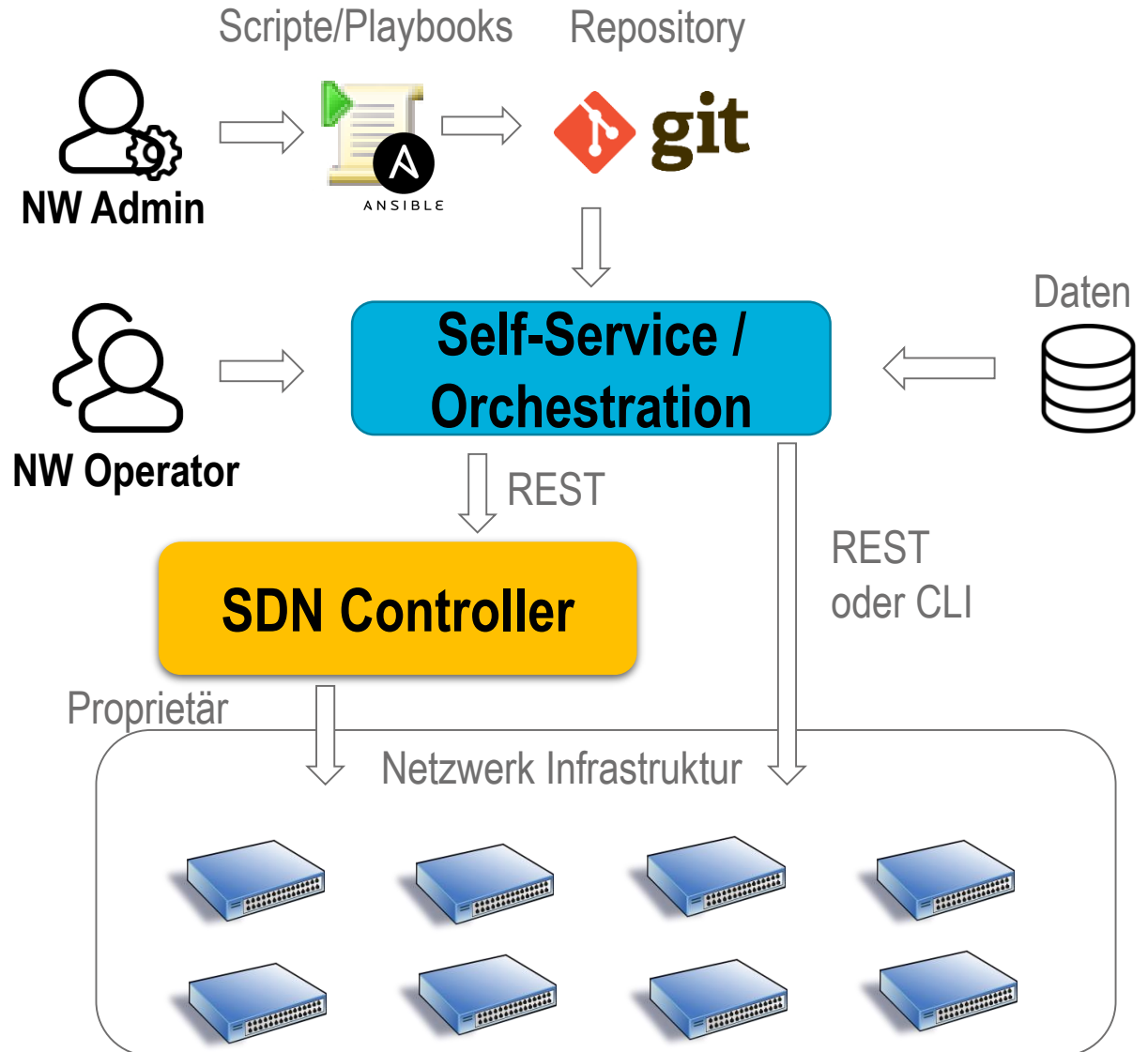
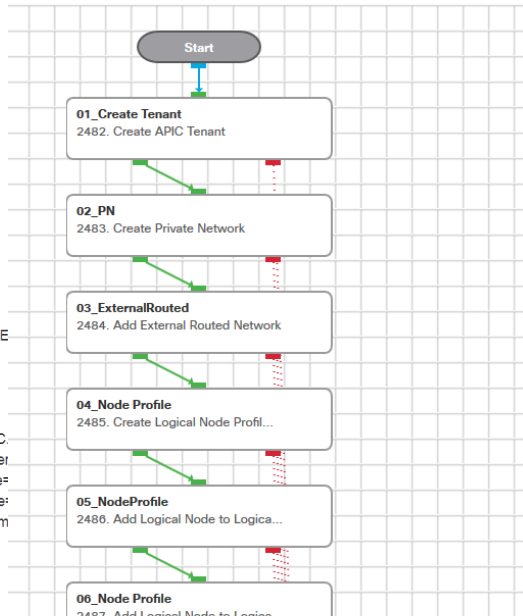
# log into an APIC and create a directory object
ls = cobra.mit.session.LoginSession('https://1.1.1.1', 'admin', 'password')
md = cobra.mit.access.MoDirectory(ls)
md.login()

# the top level object on which operations will be made
# Confirm the dn below is for your top dn
topDn = cobra.mit.naming.Dn.fromString('/uni/infra/funcprof/accbundle-CC.Group:SE
topParentDn = topDn.getParent()
topMo = md.lookupByDn(topParentDn)

# build the request using cobra syntax
infraAccBndlGrp = cobra.model.infra.AccBndlGrp(topMo, lagT='u'node', name='u'CC
infraRsAttEntP = cobra.model.infra.RsAttEntP(infraAccBndlGrp, tDn='u'uni/infra/atter
infraRsCdpIfPol = cobra.model.infra.RsCdpIfPol(infraAccBndlGrp, tnCdpIfPolName=
infraRsLldpIfPol = cobra.model.infra.RsLldpIfPol(infraAccBndlGrp, tnLldpIfPolName=
infraRsLacpPol = cobra.model.infra.RsLacpPol(infraAccBndlGrp, tnLacpLagPolNam

# commit the generated code to APIC
print(toXMLStr(topMo))
```

1	IP	NM	Name	Model	Port#	C
2	1.1.1.1	255.255.255.255	S-01-EN02-E210-AC	WS-C3560CX-8PC-S	8	
3	1.1.1.1	255.255.255.255	S-01-EN02-E211-AC	WS-C3560CX-8PC-S	8	
4	1.1.1.1	255.255.255.255	S-01-EN02-E212-AC	WS-C3560CX-8PC-S	8	
5	1.1.1.1	255.255.255.255	S-01-EN02-E213-AC	WS-C3560CX-8PC-S	8	
6	1.1.1.1	255.255.255.255	S-01-EN02-E215-AC	WS-C3560CX-8PC-S	8	
7	1.1.1.1	255.255.255.255	S-01-EN02-E216-AC	WS-C3560CX-8PC-S	8	
8	1.1.1.1	255.255.255.255	S-01-EN02-E217-AC	WS-C3560CX-8PC-S	8	
9	1.1.1.1	255.255.255.255	S-01-EN02-E218-AC	WS-C3560CX-8PC-S	8	
10	1.1.1.1	255.255.255.255	S-01-EN02-E220-AC	WS-C3560CX-8PC-S	8	
11	1.1.1.1	255.255.255.255	S-01-EN02-E224-AC	WS-C3560CX-8PC-S	8	
12	1.1.1.1	255.255.255.255	S-01-EN02-E226-AC	WS-C3560CX-8PC-S	8	



ANSIBLE



ANSIBLE

AUTOMATISIERUNGSLÖSUNGEN



ANSIBLE



POSTMAN



SALTSTACK



CHEF™



ANSIBLE

- Open Source Automatisierungs Werkzeug zur Orchestrierung von verschiedensten Systemen
- Keine zusätzliche Software auf den Ziel Systemen notwendig (Agent less)
- Ursprünglich als Server Provisionierung-tool entwickelt
- Ansible Module für verschiedene Systeme verfügbar > 600 Module
- Unter den Top 10 open source Projekten 2014
- 2015 wurde Ansible von Red Hat gekauft, bleibt aber open source
- Ansible Tower als Kommerzielles Orchestrierung Tool
- Ansible Galaxy: mehr als 18.000 Teilnehmer
- 250.000+ Downloads pro Monat
- Aktuell: Ansible Core 2.10 / 3.0 Community



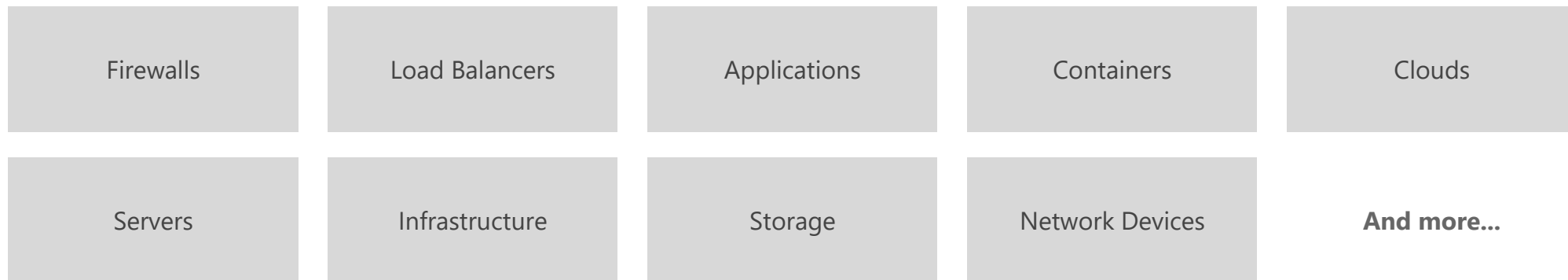
ANSIBLE

EINSATZBEREICHE

Do this...



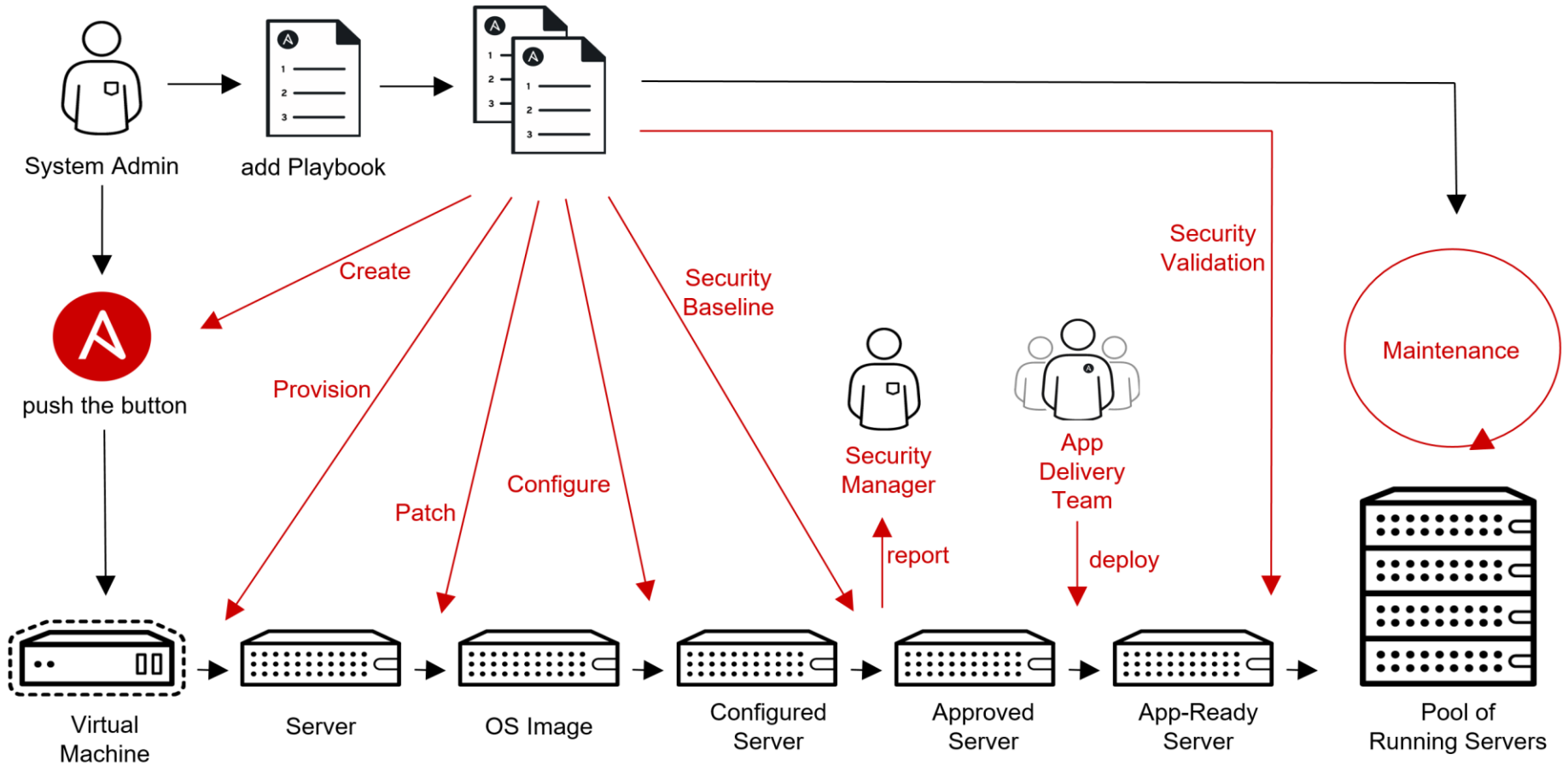
On these...



ANSIBLE

AUTOMATISIERUNG AUF KNOPDRUCK

und das alles nur
1x



ANSIBLE PLAYBOOKS

```

1 ---
2 - name: Interface configuration on Cisco N9k in NX-OS mode
3   hosts: dc_core_sc_muc
4   gather_facts: no
5   connection: local
6
7 tasks:
8
9   - name: set hostname
10     nxos_config:
11       lines: hostname dc-core-1-test
12       save_when: never
13
14   - name: configure interface
15     nxos_config:
16       lines:
17         - no description used for ansible testing
18         - no mtu 9000
19         - shutdown
20     parents: interface eth1/33
21     save_when: never
22

```

Playbook

Connection

Tasks

Modul

Modul

- Formatierung in YAML code
- In Form von Ansible Playbooks
- Von Menschen lesbar
- Schnell und leicht zu erlernen
- Sehr gute Dokumentation
- Basiert auf Python
- Vorgefertigte Ansible Module



IT@M

Erfolgreichen Beispiel aus der Praxis



Landeshauptstadt
München



AUSGANGSLAGE

Ist-Stand

- 2 RZ Standorte mit Cisco Nexus 7000 und 2000
- Ca. 80 Switche mit ca. 2500 Access Ports
- Über Jahre gewachsene komplexe Konfiguration
- (fast) keine Automatisierung im Einsatz
- Keine Zentralisierung und Versionsmanagement bei vorhandenen Skripten
- Wenig programmier Know-How im Team

Soll-Stand:

- „Alles neu -> modern“
- „ganz viel Automatisierung“
- Umstellung im laufenden Betrieb ohne Ausfälle!

Ziel:

- Komplette automatisierter Aufbau
- Automatisierter Ablauf der Migration inc. Testen
- Live Demo und Ergebnispräsentation für die Werkleitung



Entscheidung für die Cisco Application Centric Infrastructure (ACI) als Software-Defined Netzwerk (SDN) Lösung mit Zentralisierter REST Schnittstelle



Implementierung von Ansible als open source Automatisierungstool. Erarbeitung von verschiedenen Playbooks zum Bereitstellung und Betrieb der Cisco ACI Fabric



Einfügung von GIT als zentrale Versionsverwaltung



Automatisierungs Workshops und Cisco ACI Schulungen des Netzwerk Teams

- Komplette neue, moderne RZ Netzwerk Infrastruktur von Cisco
- Automatisierter und Standardisierter Aufbau der ACI Fabric
- Komplette automatisierte Migration ohne Downtime
- Von Beginn bis Abschluss nur ca. 1,5 Jahre
- Ansible Automatisierungs Umgebung mit Versions-tool
- > 50 Playbooks
- > 120 Templates
- > 500 commits
- Deutlich reduzierte Anzahl an Fehlern
- Viel schnellere Bereitstellungszeiten durch automatisierte Prozesse



NETZWERK AUTOMATISIERUNG DEMO



NETZWERK AUTOMATISIERUNGS DEMO

CONFIGURE CISCO NEXUS 9000 VIA ANSIBLE

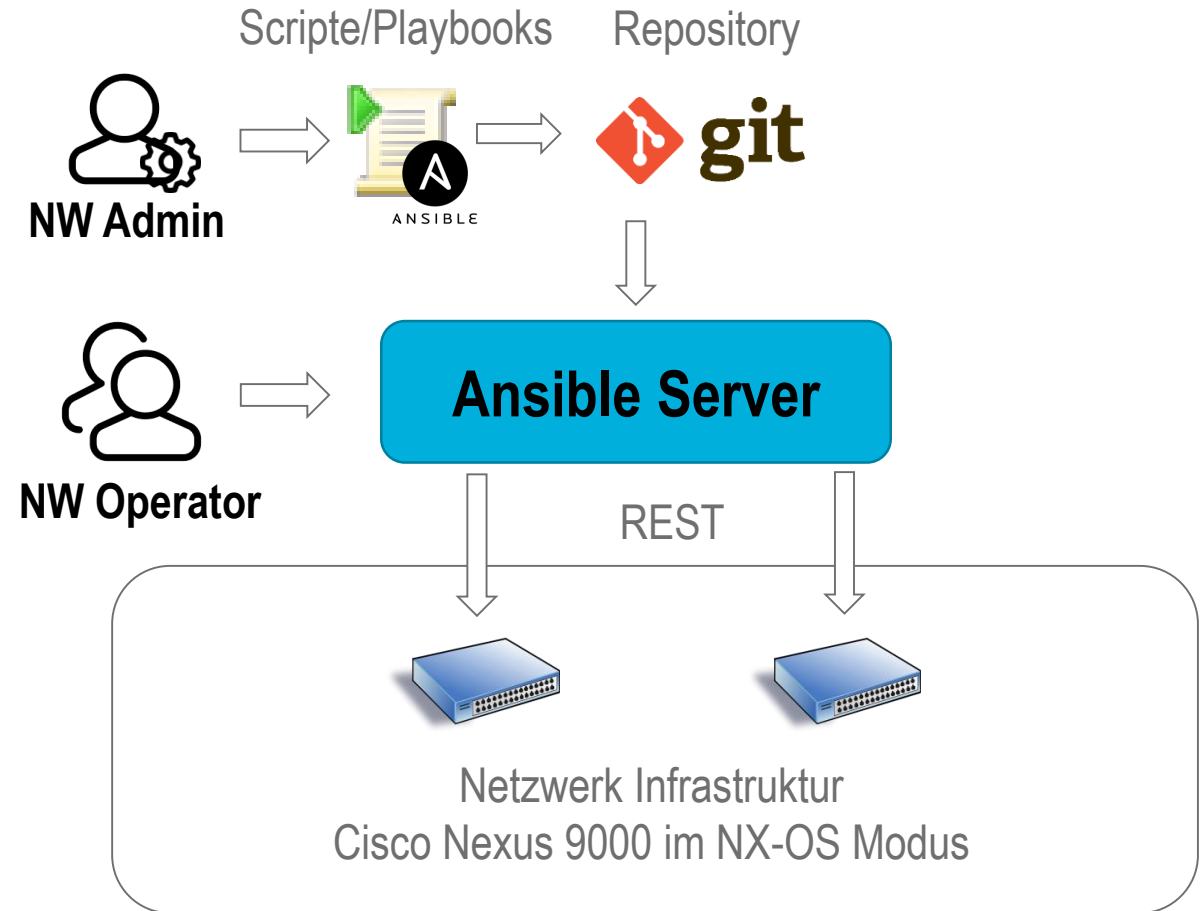
Step 1: enable feature nxapi on the device switch

Step 2: Ansible Inventory Anpassen

Step 3: Ansible Playbook erstellen

Step 4: Konfiguration Parameter einfügen

Step 4: Playbook ausführen



FRAGEN

