IT Infrastructure Automation

Automatisierung als Antwort auf kontinuierliche Veränderungen in einer digital vernetzten Welt

Motivation: Modernisierung für eine digitale Welt

Kontinuierliche Veränderung

- Geschwindigkeit und Umfang der modernen Geschäftsinnovationen nehmen zu
- Kunden verlangen immer mehr, und das immer schneller.

Komplexe IT hemmt Innovationen

- IT-Operations-Teams haben genug damit zu tun, sich ständig verändernde, komplexe IT-Architekturen zu verwalten, die auf mehreren Plattformen und komplizierten Technologie-Stacks aufsetzen
- Geschäftsabläufe zu verbessern

Hilfe durch Automatisierung

Der aktuelle Stand des Netzwerkmanagements - Beispiel

Das Netzwerkmanagement hat sich langsam entwickelt

- Über Jahrzehnte hinweg kaum verändert
- Netzwerke werden manuell entwickelt, ausgeführt und gewartet
- Network Operators (NetOps) melden sich bei Routern, Switches, Load Balancern und Firewalls an, ändern die Konfiguration manuell
- Dient meist dazu, die in Geschäftsprozessen definierten Netzwerkrichtlinien zu implementieren und einzuhalten

Gründe:

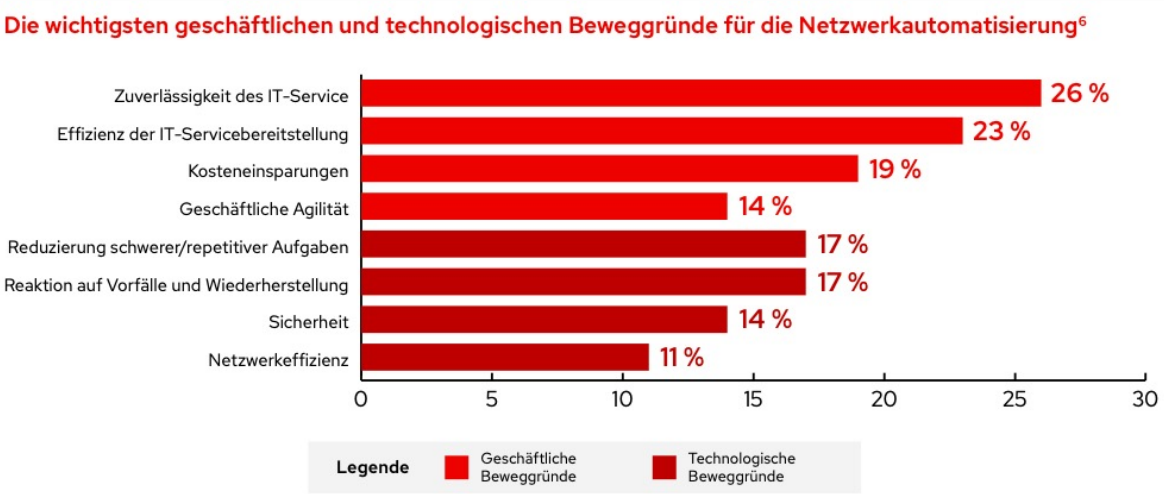
1. Stark isolierte Domains und Plattformen
2. Bindung an Produktfunktionen
3. Abhängigkeit von CLIs bei Netzwerkgeräten
4. Monolithische, proprietäre Plattformen

Was ist Automatisierung?

Infrastrukturautomatisierung ist der Prozess, bei dem die menschliche Interaktion mit IT-Systemen reduziert wird, indem Skripte oder Funktionen erstellt werden, die wiederholbar sind und entweder von anderer Software oder auf Befehl verwendet werden können.

Optimierung manueller Prozesse mit programmierbarer Logik

- Programmierbare Logik zur Verwaltung von Ressourcen und –diensten
- Mit ihr können Ops-Teams Infrastruktur (Schichten 1-3) und Anwendungsservices (Schichten 4-7) konfigurieren, skalieren, sichern und integrieren



6 Juniper Networks: „The 2020 State of Network Automation“, Oktober 2020.

Unterschiede zwischen Operating (Betrieb) und Operations

Automation unterscheidet *pures Operating (Betrieb)* von *Operations*

- **Pures Operating** (Betrieb von Infrastruktur)
  - Status der Hardware und Abfragen von Hardware-Informationen wie Lüfterstatus, Hardware-Fehlern, CPU, RAM
  - Abfrage der Ressourcenauslastung und der verfügbaren Bandbreiten

  → *Schwierig zu automatisieren*, weil extrem hersteller-, hardware- und produktabhängig *Sicherheitsaspekte* und *Beständigkeit* spielen eine grosse Rolle; teure, leistungsfähige Hardware (Solche Elemente haben eine deutlichere längere Lebensdauer als Software: etwa 5-6 Jahre oder auch oft bis zu 7 Jahre)
  → *Immer ein Read → Daten holen*

- **Operations** (Konfiguration und Analyse von Infrastruktur)
  - Konfiguration des Switches und der Infrastruktur
  - Orchestrierung und Automatisierung: Automatisiertes Deployment von Komponenten und der Konfiguration
  - Analytik und Daten: Sicherheitsüberwachung und Performance (Verfügbarkeit und Leistung)

  → *Automation wird in der Praxis über Hersteller-Tools oder ganzheitliche Tools wie Ansible schon gemacht*
  → *Aktive Konfiguration → Read & Write*

- **In der Praxis** (grosse Unterschiede zw. „Konfiguration/Analyse" und „Deployment/Orchestrierung")
  - ✓ Konfiguration und Backup von Geräten wird über Tools automatisiert
  - ✓ Analyse der Leistung und des Verkehrs ist heute schon automatisiert (praktisch jede Firewall hat eine REST-API-Schnittstelle) → Dashboards
  - *Aber:* Viel Diskussion mit grossem Marketing für automatisiertes Deployment und Orchestrierung der Infrastruktur: wird aktuell in der Praxis *nicht* gemacht; Komponenten haben Kinderkrankheiten
    - *Aber:* Virtuelle Infrastruktur wird sehr oft automatisiert: Grosse Unterschiede zwischen „On-Prem" und „in der Cloud"

Infrastruktur und Automatisierung: **Warum brauchen wir Automatisierung?**

- Die Vorteile der Infrastruktur-Automatisierung sind:
  - **Self-Service:** Man kann Infrastruktur on-demand erstellen. Dies ergibt Geschwindigkeit und Agilität
  - **Bedarfsgerechte Skalierung:** Apps und Plattformen müssen in der Lage sein, als Reaktion auf Verkehrs- und Workload-Anforderungen nach oben und unten zu skalieren und heterogene Kapazitäten zu nutzen.
  - **Beobachtbarkeit und Prüfung:** Ein beobachtbares System ermöglicht Benutzern, den internen Zustand eines komplexen Systems zu speichern und zu überwachen
  - **Tests und Wiederverwendbarkeit:** Die Apps und Plattformen sind so konzipiert, dass sie mehrfach erstellt und damit leicht getestet werden können.

Formen der Infrastrukturautomasierung

- Mehrere Anwendungsfälle
- Abhängig von der Zielsetzung gibt es unterschiedliche Arten der Infrastrukturkonfiguration.

***Walk*: Nur lesende Automatisierung**	***Run*: Richtlinien aktivieren und Self-Service über mehrere Domänen hinweg bereitstellen**	***Fly*: Vollautomatisierte Breitstellung eines Produkts, ggf. CI/CD**
• Mit Automatisierung werden Informationen gesammelt, Konfigurationen geprüft, (Netzwerk-)Verbindungen getestet • Kann Teil anderer Schritte sein.	• Durchsetzen von Richtlinien auf allen Komponeten • Onboarding-Workflows und tägliche Netzwerkkonfiguration en werden bereitgestellt.	• Für komplexe, vollautomatisierte Bereitstellung von einem Endprodukt von der Erstellung der Storage-, Compute und Netzressourcen bis zur Software

Herausforderung: **Day 1/Day 2-Problem in der Infrastruktur-Automatisierung**

- Day 1-Challenge: Wir haben noch rein gar nichts am Laufen. Wie kommen wir vom Stand „zero“ zum Stand „initiale Bereitstellung“? (anfängliche Bereitstellung)

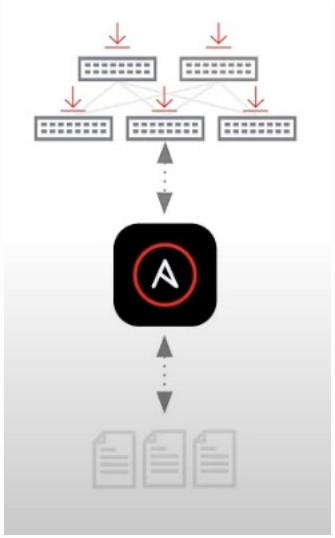
- Day 2-Challenge: Wir haben eine bestehende Infrastruktur und wollen diese weiterentwickeln. Wir ändern die Struktur im Laufe der Zeit, fügen neue Dienste hinzu, entfernen Dienste; im Allgemeinen entwickeln wir das Aussehen unserer Infrastruktur weiter.

Beides versucht man mit Infrastructure-as-Code (IaC) und dazugehörigen Automatisierungstools zu lösen.

Anwendungsfall #1: Sicherung und Wiederherstellung von Gerätekonfigurationen

Automatisiertes Abfragen aller Cisco oder Checkpoint-Geräte-Konfigurationen

- Backup für den (Hardware-)Fehlerfall
- Wiederherstellung von Konfigurationen per Tastendruck
- Rollback zu einem früheren Zustand
- Automatisierungslösungen bringen von sich aus Tools zum Informationssammeln mit.
- Automatisierungstools bringen oft Tools mit, um Netzwerkbetriebssystem-agnostisch Daten abzufragen

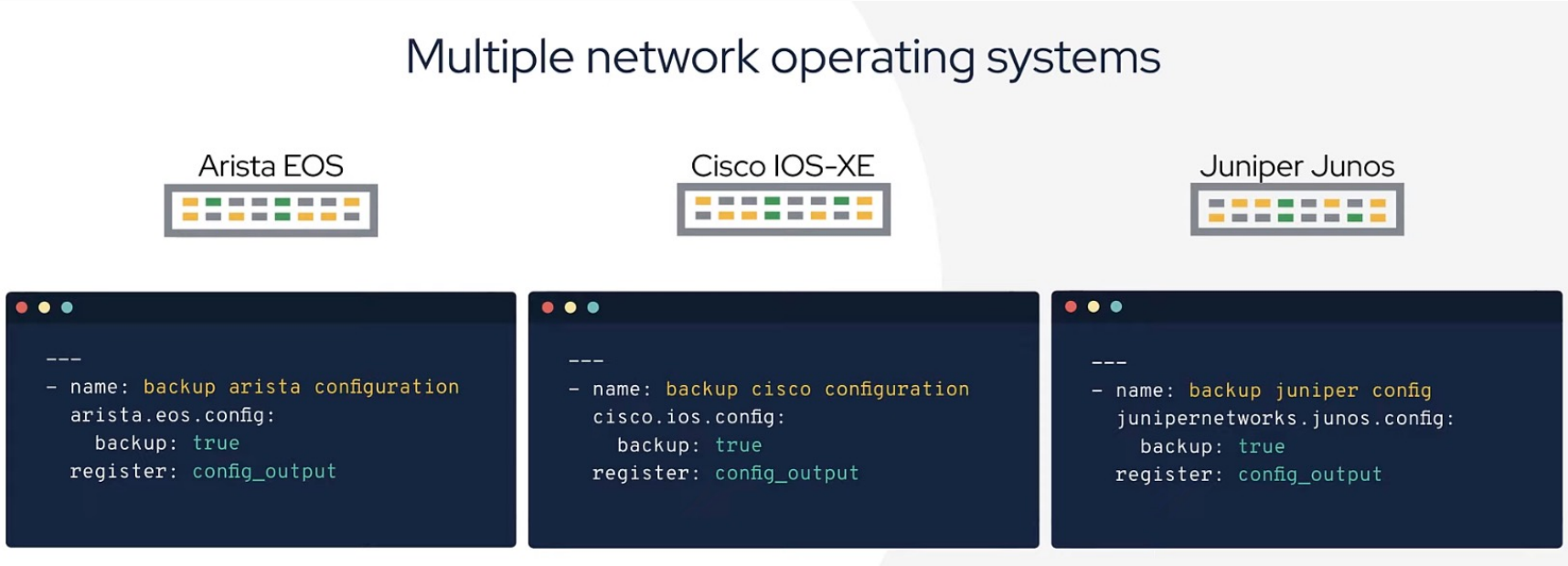


Vorteile:

1. Ergibt schreibgeschützte Kopie der aktuellen Konfiguration
2. Absicherung für den Fehlerfall
3. Schnelle Wiederherstellung der Infrastruktur

```yaml
---
- name: network backup
  hosts: all
  gather_facts: false
  tasks:
    - name: multi platform backup
      include_role:
        name: network.toolkit.backup
```

Anwendungsfall #1: Sicherung und Wiederherstellung von Gerätekonfigurationen

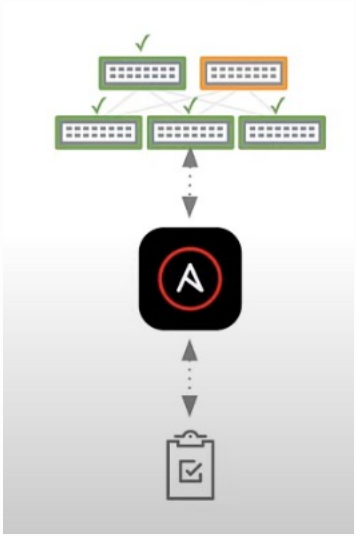


Anwendungsfall #2: Hersteller-unabhängige Konfiguration

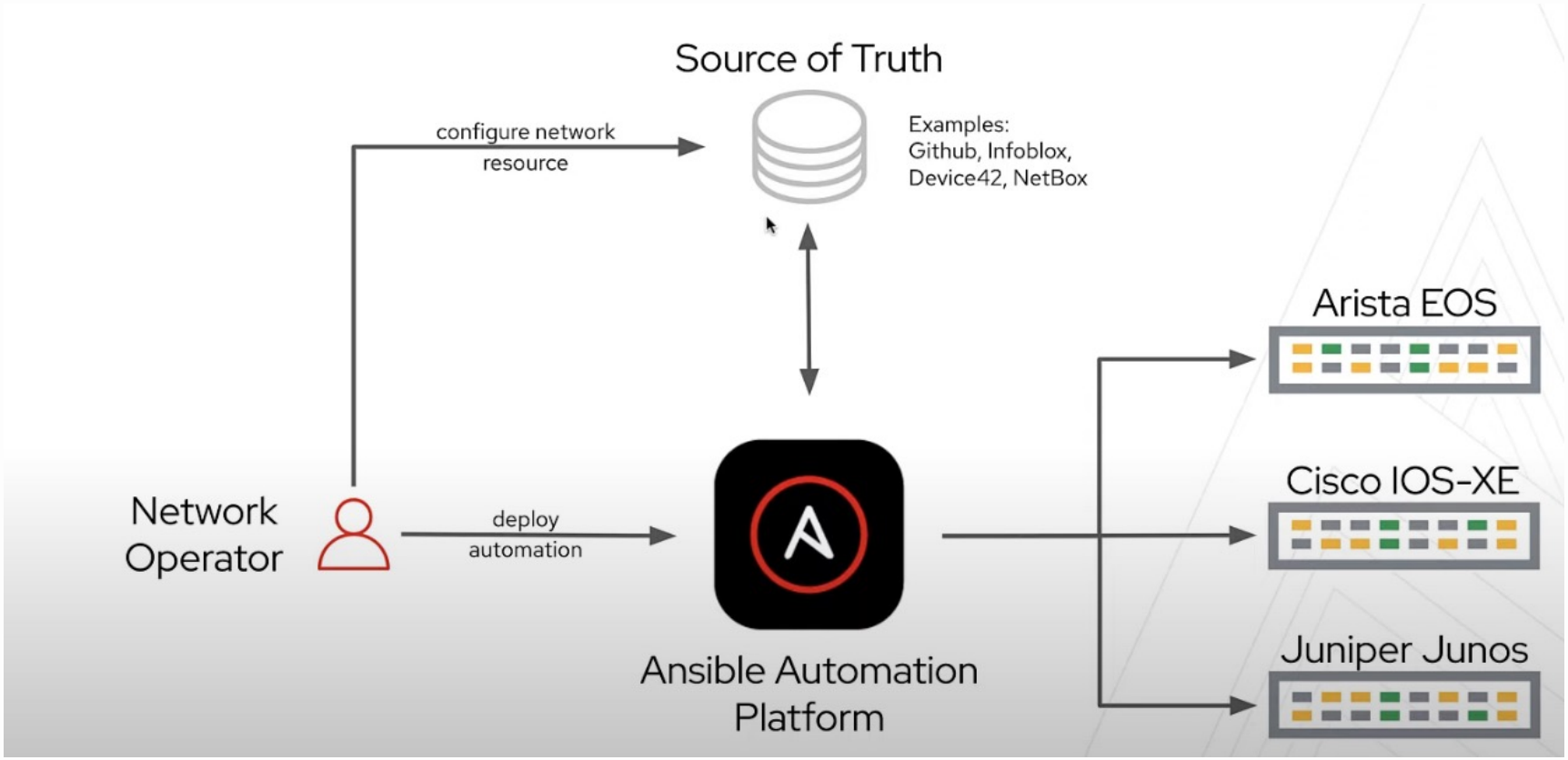
Durchführung einer Konfigurationsaufgabe auf heterogener Hardware

- Aufschreiben der gewünschten Konfiguration in einem strukturierten Format, zB. yaml
- Einlesen mit Automatisierungstool mit generalisiertem Ressource-Module
- Umwandlung in native Konfigurationsbefehle für bestimmte Hardware (falls möglich)
- Ausführung der nativen Konfigurationsbefehle auf der Hardware

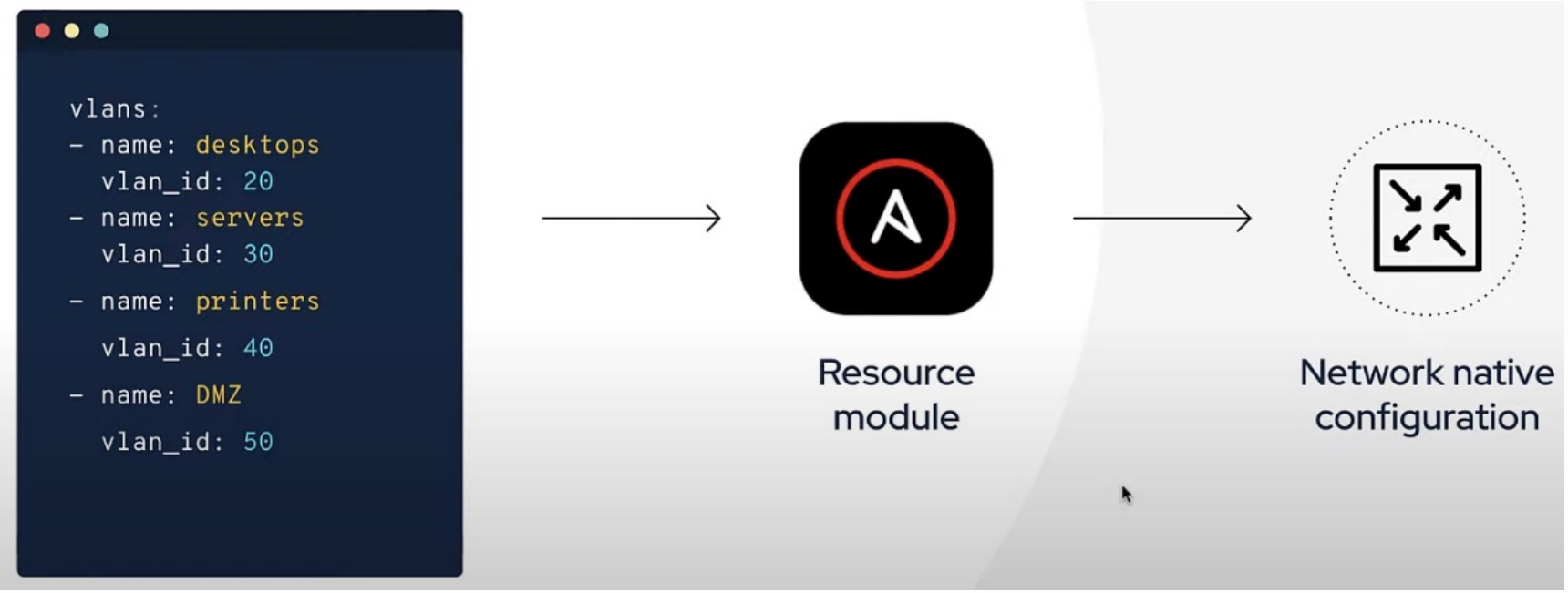
Vorteile:
- Generalisierbarkeit von Anweisungen
- Zeitersparnis
- Herstellerunabhängigkeit



Anwendungsfall #2: Hersteller-unabhängige Konfiguration



Anwendungsfall #2: Hersteller-unabhängige Konfiguration



Anwendungsfall #2: Hersteller-unabhängige Konfiguration

```
vlans:
- name: desktops
  vlan_id: 20
- name: servers
  vlan_id: 30
- name: printers
  vlan_id: 40
- name: DMZ
  vlan_id: 50
```

```
- name: add VLAN configuration
  arista.eos.vlans:
    config: "{{ vlans }}"
    state: merged
```

State:		
	Merged	- add/increment
	Replaced	- template/diff
	Overridden	- force/policy
	Deleted	- destroy/remediate

Beliebte Automatisierungstools

- **Erste moderne Automatisierungstool: Puppet**
  - 2005 als Open Source eingeführt und 2011 von Puppet Labs als Puppet Enterprise kommerzialisiert

- **Die beliebtesten Automatisierungstools sind *Ansible*, *Puppet*, *Chef*.**
  Sie teilen die folgenden Eigenschaften:
  - Relativ leicht zu erlernen
  - Verfügbar in Open-Source-Versionen
  - Plugins und Adapter ermöglichen es, viele Arten von Ressourcen direkt oder indirekt zu steuern

- Es gibt auch viele andere Lösungen. Anbieter privater und öffentlicher Clouds empfehlen häufig ihre eigenen Tools zur Verwendung auf ihren Plattformen, wie das HEAT-Projekt von OpenStack, CloudFormation, SaltStack und Terraform von AWS.

Beliebte Automatisierungstools

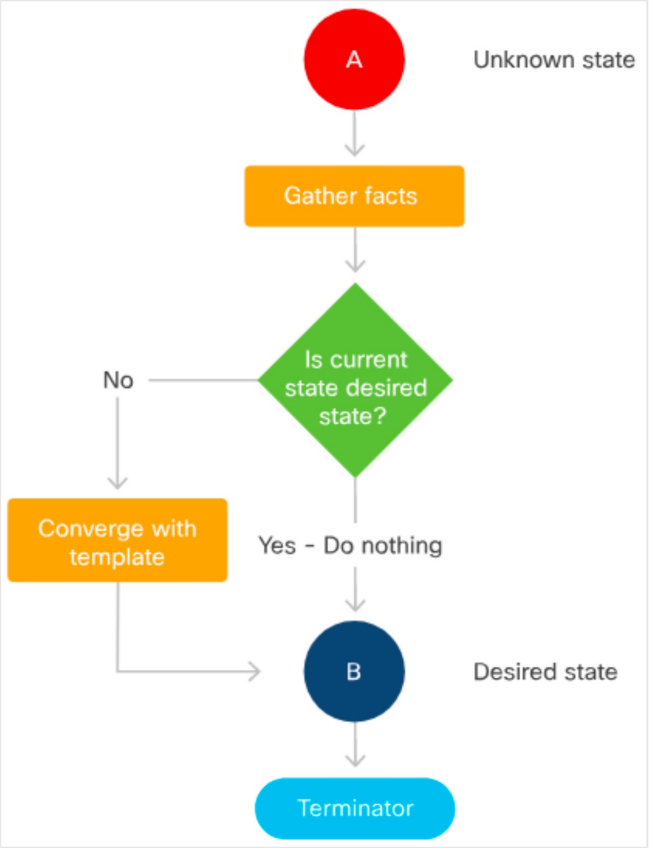
- Automatisierung ist nicht nur Softwareverteilung, sondern auch Orchestrierung von Infrastruktur
- Man unterscheidet Tools zur
  1. Konfiguration und Administration von Computern und Netzwerkgeräten
  2. Tools zur Automatisierung der Infrastruktur

Automatisierung der Konfiguration und Administration von Computern

Automatisierung der Infrastruktur

Ablauf und Idee der Automatisierungstools

- Definition eines Endzustands für ein System
- Automatisierungstool kennt und verwaltet Ist-Zustand
- Automatisierungstool entscheidet über Änderungen und führt diese ggf. durch, z. B. Installation des Webservers



Kernkonzepte

- **Push-Prinzip:** Ein Tool ist ein Push-basiertes Konfigurationsmanagement-Tool. Es gibt keinen Client. Der Haupt-/Zentralserver überträgt die Konfigurationsinformationen an die Knoten. Tools steuern die Knoten, wenn die Änderungen auf dem Server vorgenommen werden. Es ist also der Hauptserver, der die Kommunikation initiiert, nicht die Knoten.

- **Agentless:** Ein Tool verwendet keine Agenten oder Tools auf dem Zielknoten. Es wird immer eine Remote-Verbindung an den Zielknoten über Netzwerk erstellt.

- **Idempotent:** Wiederholte Ausführung führt zum gleichen Endergebnis

- **Deklarativ und Imperativ:** Playbooks geben den gewünschten Zielzustand des Systems vor. Wie dieser erreicht wird, entscheidet das Werkzeug.

Gründe für eine Automatisierung

Produktivität

Schneller auf Änderungen reagieren
- Optimierung von Routinetätigkeiten
- Änderungen automatisch testen und bereitstellen
- Automatisieren von repetitiven Tasks

Compliance

Einhalten von Richtlinien und Vorschriften durch Prozesse
- Konfigurationen an zentraler Stelle
- Automatisierte Tests von Änderungen
- Nachvollziehbarkeit
- Verifizierbarkeit: Vergewissern, dass Änderungen ordnungsgemäss durchgeführt wurden

Sicherheit

Kurze Updatezyklen mit schnellen Änderungen
- Automatisches Sammeln von Informationen zu Netzwerggeräten
- Erstellen und verwalten von Inventories

Verfügbarkeit

Erhöhte Verfügbarkeit durch schnelles, automatisiertes Deployment
- Reduzieren von Fehlern im Vergleich zu manueller Konfiguration
- Automatische Neuinstallation/-konfiguration bei Hardwarefehlern
- Im besten Fall: Skalierung, um Anforderungen gerecht zu werden.

IT Infrastructure Automation

- Automatisierung ist die Optimierung von manuellen Prozessen.
- Dadurch erhalten Unternehmen Transparenz und Kontrolle über Prozesse.
- Inhalte
  - Gründe, Beispiele und Kernkonzepte der Automatisierung
  - Konfiguration und Administration vs. Infrastrukturautomatisierung