Automatisierung für Netze

September 2019

jl@jenslink.net

/me

- Freelancer (ich bin käuflich)
- Linux seit es das auf 35 Disketten gab
- Cisco Netzwerke seit 2002
- IPv6 seit 10 Jahren oder so
- DNS, DNSSEC
- Monitoring
- Automatisierung

Bullshit Bingo

- DevOps, DevOps, DevOps
- IMHO: Nur weil man als Admin Config Management verwendet ist das noch lange kein DevOPS
- Aber es verkauft sich gut ;-)
- Juniper hat 2 eigene Zertifizierungen

Wenn wir über Netze reden

- Software Hersteller: ca. 30 Komponenten
- Hochschule: ca 300 Komponenten
- Versicherung: ca. 3000 Komponenten
- und es geht größer

Warum nicht von Hand?

- Konfiguration nicht einheitlich
- Dinge werden vergessen (ntp server, logging, ACLs)
- Änderungen werden nicht gemacht

Was ändert sich so alles

- NTP-Server
- VLAN Zuordnungen
- SNMP Communitys bzw. Username + Passwort
- Username / Passwort fürs Login
- ....

Fragen:

- Wann habt ihr zuletzt die Passwörter auf den Switchen gewechselt?
- Oder die SNMP Authentifizierung?
- Und auf wie vielen Switchen hat immer noch die IP des Vor-Vor-Vorgänger des Monitoringsystems freigegeben?

Aber da gibt es doch sicher was von Ratioˆ WCisco

- Netzwerkmanagement Software in kommerziell gibt es, aber kostet €€.€€€ oder auch €€€.€€€
- Viele Hersteller haben ihre eigene Lösung, wenn man mehr als einen Hersteller hat, braucht man für jeden eine eigene Software
- Software ist komplex, nicht einfach zu bedienen, hat obskure Fehler, kein richtiges Logging und ist langsam
- Wird of gekauft und dann nie genutzt

In der realen Welt

- Configs aus Word / Excel
- Excel als IPAM
- Netzwerkadmins die nicht scripten können / wollen (“Ich bin doch kein Softwareentwickler”)
- Andere Abteilungen mit denen man reden müsste
- Webinterface vom Router / Switch fürs Monitoring

Warum Ansible?

- Es ist relativ einfach und schnell zu lernen
- Nutzt ssh, kein Agent auf dem “Client”
- sehr beliebt im Netzwerkumfeld

Aber Puppet kann doch auch Cisco / Juniper!

- Sicher? Puppet geht nur auf bestimmten Typen und nur für einige wenige Funktionen

Wie?

- Entweder nur für einzelne Änderungen
- komplettes Setup aus dem Config Management

=> Hängt von der Umgebung ab

Ein neues Gerät von Hand

- Auspacken
- Inventarisieren
- IPs raussuchen
- (DNS Einträge machen)
- (reverse DNS Einträge machen)
- (neues Image)
- Konfiguration per copy'n'paste / von Hand
- (Monitoring)

Ein neues Gerät - Wie man es auch machen kann

- Auspacken
- Inventarisieren (DCIM)
- Erster Boot ZTP
  - neues Image
  - Basis Config, generiert aus dem DCIM / IPAM
- DNS Einträge automatisch
- Einbindung ins Monitoring automatisch
- Andere Configs automatisch

Ein neues VLAN von Hand

- anlegen auf alle Switchen (wenn man VTP und co. nicht mag)
- passendes VLAN und IP Prefix im IPAM suchen
- ggf. DHCPv4/v6)Config
- Layer-3 Config
- reverse DNS

Ein neues VLAN automatisch

- Den passenden Ansible Code ausführen, Daten kommen aus dem IPAM

ZTP

- Zero Touch Provisioning
- Im Prinzip DHCP + TFTP / HTTP
- Heißt nicht unbedingt so (Cisco IOS: AutoInstall)
- neues Image + Basis Config
- Man will keine Devices ohne

ZTP Juniper

- Unterschiedlich für EX/MX und SRX
- Etwas fummelig

Werkzeugkasten

- IPAM / DCIM - Schaut euch mal Netbox an
- zentrales Git Repository
  - Review für kritische Änderungen
  - automatische Tests
- Monitoring, DNS, . . . ggf. mit API

https://github.com/digitalocean/netbox

Ansible und Cisco - SNMP Config

```
- name: configure SNMP
  ios_config:
    lines: hostname {{ inventory_hostname }}
       provider: "{{ cli }}"
    - name: configure interface settings
      ios_config:
      lines:
          - snmp-server community foobar RO 1
          - name: load new acl into device
            ios_config:
            lines:
              - 10 permit ip host 192.0.2.1
              - 30 deny any log
              parents: ip access-list extended test
              before: no ip access-list extended test
              match: exact
              provider: "{{ cli }}"
```

Ansible und Cisco - User Config

```
- name: configure user
  ios_config:
    lines: hostname {{ inventory_hostname }}
       provider: "{{ cli }}"
    - name: configure interface settings
      ios_config:
      lines:
        - username foo secret bar
```

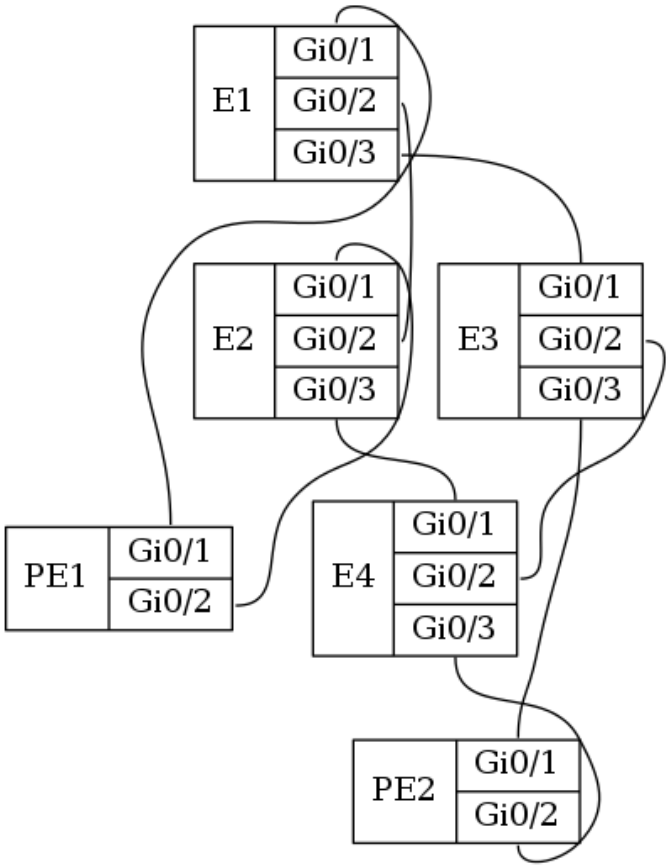
Ansible und JunOS

```
- name: Change configuration using junos_config module
  hosts: all
  connection: local
  gather_facts: no
  tasks:

  - name: Change description on interfaces based on a list
    junos_config:
      lines:
        - "set interfaces {{ item.name }} description \"{{
      comment: "Update description of interface {{ item.nam
      provider: "{{ credential }}"
with_items: "{{ interfaces }}"
```

Mehr als Config Management

- Informationen von Geräten sammeln
- Troubleshooting Playbooks
- Netzwerkdoku



Noch mehr Abstraktion: NAPALM

NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support) is a Python library that implements a set of functions to interact with different router vendor devices using a unified API.

https://github.com/napalm-automation/napalm

NAPALM

- Für SALT und Ansible
- Zugriff auf die Remote Devices per NetConf, Expect, ...

NAPALM Devices

- IOS / IOS-XR / NXOS
- JunOS
- FortiOS
- IBM
- ...

Resourcen:

- JuniperAnsibleDayOne
- https://blog.ipspace.net/
- https://www.dravetech.com/presos/network_automation_tutorial.html

Fragen?