

Applied Network Automation

Adviesrapport v1.1

Titel	Adviesrapport
Versie	v1.1
Project	Applied Network Automation
Auteur(s)	Martijn Remmen
Datum	15 juni 2022
Opdrachtgever	KEMBIT Services

Versiebeheer

1.0 Martijn Remmen (2022-06-13) — Initiële versie

1.1 Martijn Remmen (2022-06-15) — Inleiding bijgewerkt

Inhoudsopgave

1	Inleiding	3
2	Adviezen	4
2.1	Installeren van WSL op de beheerserver	4
2.2	Ontwikkelen van nieuwe playbooks met Ansible CLI	4
2.3	Investeren in developmentkennis	4
2.4	Produceren en onderhouden van integraties en tooling	5
2.5	Nieuwe inrichting van TOPdesk Asset Management	5
2.6	Rekening houden met de kwaliteiten van Ansible	5

1 Inleiding

Dit document beschrijft diverse adviezen die worden gemaakt richting de opdrachtgever van het project. De adviezen betreffen zaken die de opdrachtgever kan verwerkelijken met als doelstelling om automatisering te bevorderen in de huidige werkwijze van het Expert Team Networking. Deze adviezen zijn gebaseerd en onderbouwd door het uitgevoerde onderzoek.

2 Adviezen

Op basis van het uitgevoerde onderzoek wordt een advies uitgebracht naar de opdrachtgever met betrekking tot de vervolgstappen om netwerkautomatisering te kunnen bevorderen in de dagelijkse werkzaamheden van het Expert Team Networking.

2.1 Installeren van WSL op de beheerserver

Installeren van de 'WSL' (Windows Subsystem for Linux) rol op de windows beheerserver maakt het mogelijk om gebruik te maken van Linux op de beheerserver. Dit maakt het mogelijk om Ansible hierop te installeren en te gebruiken. Hierdoor kan de beheerserver gebruikt worden om de dagelijkse werkzaamheden te automatiseren met Ansible.

Er zijn de volgende alternatieven naast het installeren van WSL op de beheerserver:

- Een andere beheerserver in gebruik nemen die gebaseerd is op Linux.
- Het beheermodel aanpassen zodat werknemers de klantomgevingen kunnen benaderen met hun lokale computer.

2.2 Ontwikkelen van nieuwe playbooks met Ansible CLI

Wanneer een nieuw playbook gemaakt wordt kan deze beter gemaakt worden met de Ansible CLI op de beheerserver. Op deze manier kan de engineer het playbook op een snellere iteratieve manier ontwikkelen (in vergelijking met het maken van een nieuw playbook met AWX). Dit zorgt ervoor dat er minder tijd benodigd is om het playbook te maken en daardoor toegankelijker is voor de engineer.

AWX past hier alsnog goed bij. De playbooks kunnen namelijk getest en gemaakt worden op de beheerserver. Wanneer deze playbooks zodanig stabiel en vaak getest zijn kunnen deze automatisch worden uitgevoerd door AWX. Dit maakt CI/CD principes mogelijk op de langere termijn.

2.3 Investeren in developmentkennis

Bij het toepassen en realiseren van automatisering is niet alleen kennis van het te automatiseren proces nodig maar ook kennis over onder andere versiebeheer, ontwikkelen en testen van code. Deze kennis is belangrijk voor het creëren van betrouwbare, stabiele en functionele software.

Het Expert Team Networking heeft uitgebreide kennis over de processen die potentieel geautomatiseerd kunnen worden. Echter bezit het team minder kennis en ervaring over softwareontwikkeling en hiermee geassocieerde werkwijzes en tools. Dit kan ervoor zorgen dat het zeer lastig is voor het team om deze werkwijzes en tools effectief toe te kunnen passen. Daarnaast bestaat een grotere kans op het introduceren van technical debt door het maken/implementeren van suboptimale ontwerpkeuzes en code die moet worden gecorrigeerd bij het toevoegen van functionaliteit.

Het risico van niet investeren in developmentkennis is dat geen gestructureerd developmentproces wordt nagestreefd bij het ontwikkelen en beheren van systemen en code die de automatisering mogelijk maakt. Dit kan leiden tot onbetrouwbare systemen en/of code.

2.4 Produceren en onderhouden van integraties en tooling

KEMBIT heeft als nadeel dat er geen bestaande integraties zijn tussen Ansible en TOPdesk. Dit maakt het niet mogelijk om Ansible volledig geautomatiseerd uit te kunnen voeren en een stuk minder gemakkelijk om Ansible in de dagelijkse werkzaamheden te kunnen gebruiken.

Om dit op te lossen wordt geadviseerd om verder te evalueren welke integraties er nodig zijn, en deze te implementeren waar nodig. Om te garanderen dat deze integraties wordt tevens geadviseerd om hier tests op uit te voeren bijvoorbeeld met Continuous Integration.

2.5 Nieuwe inrichting van TOPdesk Asset Management

Gedurende het onderzoek is vastgesteld dat de inrichting van TOPdesk Asset Management de tool nog niet tot zijn volledige potentie benut. Dit veroorzaakt een verminderde gebruikerservaring en maakt het minder makkelijk om integraties te realiseren met de software.

Door een nieuwe inrichting van TOPdesk Asset Management te maken zal het makkelijker worden om koppelingen te maken en de software te gebruiken.

De software is ingericht met veel verschillende templates voor hetzelfde type apparaat. Onderling is geen verschil, alleen de naam. Het betreft de volgende templates:

- Switch 8 Poorten
- Switch 12 Poorten
- Switch 24 Poorten
- Switch 48 Poorten
- Switch Core
- Switch Distributie
- Switch Acces

Het enige dat verschilt, is de naam van de templates. In deze naam is informatie opgenomen over het apparaat dat belangrijk is voor het team. Dit zijn de hoeveelheid poorten en de plaats in de hiërarchie van het netwerk. Om dit verder te optimaliseren kunnen al deze templates worden vervangen door slechts één template met de twee eigenschappen toegevoegd als velden. Dit kan er als volgt uitzien (bestaande velden zijn niet mee opgenomen, alleen de voorgestelde wijziging):

- Switch
 - Aantal poorten (8, 12, 24, 48)
 - Hiërarchie (Core, Distributie, Acces)

2.6 Rekening houden met de kwaliteiten van Ansible

Ansible is een zeer effectieve tool voor het automatiseren van configuration management op netwerkkapapparaat. Dit komt door de toegankelijke, declaratieve configuratietaal en uitgebreide modules voor netwerkkapapparaat.

Echter is de tool niet geschikt voor gebruik als general purpose programmeertaal. Dit komt door (opzettelijke) limitaties in de domain specific language van Ansible. Deze hebben als doel om de taal zo eenvoudig en toegankelijk mogelijk te maken, ook zonder programmeerkennis.

Op het moment dat de DSL gebruikt wordt voor een usecase die op de limieten ligt van de functionaliteiten die hierin worden blootgesteld, kan dit leiden tot een juist zeer complexe implementatie.

Ook is het lastig om playbooks te testen. Het niet uitvoeren van tests kan ervoor zorgen dat er regressies optreden bij het maken van wijzigingen die niet opgemerkt worden. Over het algemeen kan het testen de stabiliteit van de software verbeteren door het voorkomen van onverwacht gedrag.

Het advies is om bij het overwegen van een nieuwe automatisering eerst de voor- en nadelen van Ansible in acht te nemen. Zeker wanneer de usecase niks te maken heeft met configuration management is het wellicht beter om een andere tool te kiezen zoals een PowerShell script.