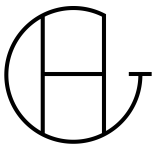

HoGent

Faculteit Bedrijf en Organisatie

Automatische opzet van Cisco IOS netwerkapparatuur in combinatie met Ansible

Gianni Stubbe

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Bert Van Vreckem

Instelling: —

Academiejaar: 2015-2016

Derde examenperiode

Faculteit Bedrijf en Organisatie

Automatische opzet van Cisco IOS netwerkapparatuur in combinatie met Ansible

Gianni Stubbe

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Bert Van Vreckem

Instelling: —

Academiejaar: 2015-2016

Derde examenperiode

Samenvatting

Internet is tegenwoordig niet meer weg te denken maar om het op te zetten is er vaak toch heel wat werk voor nodig voor een netwerkbeheerder. Zeker bij grotere bedrijven kan het configureren een tijdrovende taak zijn die vaak heel repetitief en gelijkaardig is. De laatste jaren is er veel vooruitgang geboekt bij het automatiseren van servers met behulp van verschillende frameworks en tools. Het volgende onderdeel die aan de beurt is zijn verschillende componenten uit het netwerksegment zoals switches, routers en wifi-controllers.

Meer specifiek zal er geconcentreerd worden op het automatiseren met de Ansible architectuur. Deze is uitgebreid in de les gezien bij Meneer Van Vreckem en ook in het werkveld kom je dit al maar vaker tegen. Het zou dus interessant zijn om netwerkapparatuur automatisch te kunnen configureren met dezelfde kennis die de systeembeheerders al hebben.

In het document wordt er even onderzoek gedaan naar bestaande automatisatie methoden en waarom er voor Ansible gekozen is. Daarna zal er een basis configuratie gemaakt worden met Cisco apparatuur bestaande uit 1 router en 2 switches. Deze wordt getest en de criteria die moeten voldoen worden genoteerd. Hierna wordt een "Playbook"gemaakt in Ansible en wordt deze toegepast op de apparatuur. Deze zal aan de hand van vooropgestelde resultaten vergeleken worden met de nieuwe resultaten behaald met Ansible.

Voorwoord

Inhoudsopgave

1 Inleiding 9

1.1 Stand van zaken 10

1.1.1 Wat is Ansible 10

1.1.2 Wat is Napalm 12

1.1.3 Overige mogelijkheden 12

1.2 Probleemstelling en Onderzoeksvragen 13

1.3 Opzet van deze bachelorproef 13

2 Methodologie 15

2.1 Onderzoek naar mogelijkheden 15

2.2 Uitwerking basis configuratie 16

2.3 Automatiseren van configuratie 16

3 Configuratie zonder automatisatie 17

3.1 Voorbeeld configuratie 17

3.1.1 Initiële configuratie 18

3.1.2 Minimale configuratie 18

3.1.3 Complete configuratie 20

4 Conclusie 23

Bibliografie 23

1. Inleiding

Het configureren van netwerkapparatuur kan dus heel wat tijd in beslag nemen. Hoewel er vaak bij nieuwere apparatuur mogelijkheden zijn om configuratie te automatiseren, zoals bijvoorbeeld met Cisco's eigen NX-OS die draait op zijn Nexus apparatuur, is het toch vaak vervelend dat er geen eenduidige manier is om dit te doen. (**ciscoNxosPDF**) Ieder merk heeft een eigen implementatie. Hiervoor zijn er sinds de laatste tijd enkele mogelijkheden voor opgedoken. Waar er dus meer onderzoek naar gedaan zal worden.(**Vandooren2015**)

Vorig jaar heeft Dries Vandooren een gelijkaardig onderzoek uitgevoerd naar automatisering voor netwerkapparatuur met Ansible. In de tussentijd is er al heel wat extra vooruitgang geboekt en is het dan ook interessant om te kijken in hoeverre dit gebruikt kan worden voor in productie.

Er zijn drie platformen die interessant lijken te onderzoeken. De eerste twee zijn nieuw in vergelijking met de bachelorproef die vorig jaar is afgeleverd, de andere dient als referentie. De platformen zijn:

- Anible's officiële release van Network Tools (beschikbaar vanaf Ansible 2.0)
- Napalm: Network Automation and Programmability Abstraction Layer with Multivendor support.
- Networklore's implementatie van Ansible voor Cisco IOS apparaten via het SNMP protocol.

1.1 Stand van zaken

Cisco IOS is een NOS (of Network Operating System). Dit is een van de oudste besturingssystemen die nog dagelijks gebruikt wordt op netwerkapparatuur. Het is gekend om heel robuust te zijn en is dus vaak ook als basis gezien voor de iets geavanceerdere netwerkkennis. IOS staat voor Internetworking Operating Systems en gaat al langer mee dan 1990. Hoewel dit voor informaticastandaarden al een afgeschreven iets zou zijn krijgen apparaten van 2006 nog altijd security updates en is voor een bedrijfsnetwerk dan ook nog uiterst geschikt. (**historyOfCiscoCli**)

Het enige waarbij deze apparatuur wel duidelijk toont dat deze toch wel wat verouderd begint te worden is dus automatisatie en provisioning. Bij nieuwere apparatuur worden er vaak ook al verschillende besturingssystemen meegeleverd die vaak heel wat meer automatisatie ingebouwd hebben. Een voorbeeld hiervaan is het laatste nieuwe besturingssysteem van Cisco genaamd NX-OS die op de productlijn met naam 'Nexus' draait. Deze wordt door hen beschreven als 'Open, Progammable, Agile'. Dit lost natuurlijk nog altijd het probleem niet op dat we hebben met onze oudere maar wel nog functionele apparatuur. (**ciscoNxos**)

De huidige uitvoering voor het configureren bestaat vaak uit twee methodes. De eerste is het handmatig uitvoeren van ieder commando. De tweede is een copy te nemen van de running config of startup config en deze te plakken in het terminal venster van een gelijk apparaat. Dit kan je ook doen door de config op te slaan op een TFTP server en deze dan vaan daaruit laten ophalen door een nog te configureren apparaat.

1.1.1 Wat is Ansible

Dit is waar Ansible tevoorschijn komt. Ansible is iets die de laatste jaren een heel snelle opmars aan het maken is. Maar wat is het nu precies? Ansible is een project die het automatiseren van servers over ssh mogelijk maakt. Dit klinkt niet heel spectaculair maar dat is ook niet het enige onderdeel ervan. Ansible maakt het mogelijk om op een heel gestructureerde manier te bepalen wat op een server gedaan wordt, geïnstalleerd wordt, naar gekopieerd wordt etc. Ook wordt alles in een heel leesbare vorm opgeslagen waardoor het voor mensen die geen kennis van Ansible hebben het toch eenvoudig is om het te verstaan wat er allemaal gebeurt.

Ansible is dus nog iets vrij recent, het is iets dat sinds 2012 in ontwikkeling is, startende bij een Red Hat Emerging Technologies groep in 2006. Hier werkten werknemers vrij aan wat zelf interessant leek voor de klant. Hieruit kwam 'Cobbler' voort, dit had als doel het beheren van datacenters eenvoudiger te maken. Dit werd vrij snel groot en kreeg mogelijkheden om bijvoorbeeld DHCP en DNS te beheren in combinatie met dit pakket. Dit werd zo groot dat het gebruikt werd om de grootste DNS netwerken te beheren, heel wat grotere gameservers zoals Modern Warfare van Call of Duty en dan nog kritische

zaken zoals financiële onderdelen op Wall Street.

Toen kwamen configuratie managementtools zoals Puppet en Chef op de proppen. Deze waren in combinatie met Cobbler een sterke combinatie. Cobbler stelde het basissysteem in en dan werd Puppet gebruikt om de virtuele machines op het apparaat te configureren. Een van de grotere problemen bij deze combinatie was dat er nog problemen waren met deze oplossing. Met Puppet was het bijvoorbeeld niet mogelijk om een server te herstarten.

Om dit probleem op te lossen werd Func in het leven geroepen. Het concept was vrij eenvoudig: een centrale server die andere servers aanspreekt om bepaalde taken uit te voeren. Dit is vrij gelijkaardig aan een 'Master/slave' configuratie waarbij de 'slaves' luisteren naar de 'master'. Hoewel het niet een heel groot succes werd is het toch gebruikt geweest door Tumblr en zijn er 2 spin-offs uit voortgekomen die elk ook hun onvolledigheid had.

Deze periode voor de opkomst en grotere populariteit van de term 'DevOps'. Hierbij werden heel vaak Cobbler, Puppet en Func geassocieerd. Daarna werd de betekenis uitgebreid naar de algemene cultuur rond het automatiseren van infrastructuur. Zelf is de oprichter van Ansible nog een lange tijd na deze periode een Puppet voorstander geweest maar het bleek al snel dat er een eenvoudigere taal nodig was om een groter publiek te kunnen bereiken.

Door alles samen te zetten kwam de nood voor Ansible. Zoals de bedenker het zelf zegt ' it was mostly to build the tool that you could not use for six months, come back to, and still remember.' **historyAnsible** Dus rond februari 2012 werd de fundatie gelegd voor het volledige project, omdat de ontwikkelaar al heel wat kennissen had binnen de DevOps wereld sloeg het vrij snel aan. Samen me de community wordt alles onderhouden en aangevuld. (**historyAnsible**)

Nu een viertal jaar later is Ansible uitgegroeid tot een vaste waarde voor het automatiseren van infrastructuur. In de tussentijd is er nog de toevoeging van onder andere Ansible Tower (de betaalde versie van Ansible voor grotere bedrijven). Dit biedt de mogelijkheid om een overzicht te houden van je hele serverinfrastructuur en maakt het bijhouden van Ansible playbooks eenvoudig en overzichtelijk.

Dan sinds kort de laatste belangrijke toevoeging, vooral ook voor dit onderzoek. De toevoeging van de network modules in de Ansible Core vanaf versie 2.0 die januari 2016 gelanceerd is. Deze bevat de mogelijkheid om veel netwerkapparatuur van verschillende merken aan te spreken via een Ansible playbook. (**ansible2**)

1.1.2 Wat is Napalm

Napalm staat voor 'Network Automation and Programmability Abstraction Layer with Multivendor support'. In andere woorden iets heel gelijkaardig aan wat Ansible probeert te bereiken met hun network modules. Net als ansible is alles ook open source terug te vinden op een Github pagina waardoor ook hier community kan aan meewerken. Het nadeel aan deze optie is dat deze minder aanhangers zal hebben omdat het iets volledig nieuw op zich is speciaal voor netwerkapparatuur.(**napalmGithub**)

De eerste commits op Github dateren van begin 2015 waardoor het dus wel al een stuk langer in ontwikkeling is dan de Ansible network modules. Deze wordt nog altijd actief gesupport waardoor het wel een goede oplossing kan zijn voor een huidig netwerk te automatiseren. Support voor verschillende apparaten is goed en heel uitgebreid gedocumenteerd met welk netwerk besturingssysteem wat precies mogelijk is. Hieruit is vrij snel duidelijk dat IOS het grootste deel van de mogelijkheden ondersteund.(**napalmSupport**)

1.1.3 Overige mogelijkheden

Als laatste bestaan er nog enkele kleinere opties om het opzetten van een netwerk met Cisco IOS apparatuur te vereenvoudigen.

Een van de oudere methodes om het opzetten van een switch eenvoudiger te maken is het kopiëren van de huidige configuratie van een netwerk apparaat naar een gelijke die gelijkaardig geconfigureerd moet zijn. Op die manier moet je toch al heel wat minder werk doen. Het nadeel hiermee is dat je welk nog elk apparaat afzonderlijk moet configureren al is het dan wel niet de gehele configuratie. Het voordeel van deze methode (tegenover het gewoon handmatig configureren) kan zijn dat als je bijvoorbeeld 10 switches hebt die elk dezelfde configuratie moeten hebben. Het enige verschil zal dan de ip adressering zijn wat dan uiteindelijk een kleinere wijziging is. Het nadeel is dat dit nog vrij handmatig is waardoor een foutje er sneller door sluipt.

Een laatste die nog vermeld zal worden is de 'ansible-cisco-snmp' module van Networklore. Deze is vorig jaar uitgebreid besproken geweest in het onderzoek van Dries Vandooren en wordt daarom deze keer ook nog vermeld om het verschil met de stand van zaken van vandaag te kunnen vergelijken met een jaar geleden. Wat wel op te merken valt aan de module is dat ongeveer sinds de network modules van Ansible zelf uitgekomen zijn er geen commits meer toegevoegd zijn aan de repository op Github.(**networkloreSnmpAnsible**)

1.2 Probleemstelling en Onderzoeksvragen

Tegenwoordig wordt nog heel veel netwerkbeheer gedaan aan de hand van manueel wijzigingen doorvoeren en configuraties uitschrijven. Dit kan heel tijdrovend zijn en zorgt dus voor veel overhead. Tijd is iets die veel IT'ers te kort komen en door het huidig werk te automatiseren zou er dus weer meer tijd vrijkomen voor andere taken.

Een voorbeeld uit het echte leven zou het volgende kunnen zijn. In België heb je een van de grootste Lanparties genaamd 'Frag-O-Matic'. Deze hebben jaarlijks 2 edities waar ongeveer een 1000 tal gamers op afkomen, het opzetten van dergelijke structuur is dan ook een heel werk. In het netwerkteam alleen al zitten een 15 tal enthousiastelingen die naast hun vaak full time job of school nog heel wat avonden tijd vrij maken om het netwerk voor de volgende editie op punt te stellen. Alles wordt uitgetekend, uitgewerkt en uiteindelijk getest. Zelf al hebben zij hun grootste deel van de configuratie op voorhand klaar zou het handiger zijn moest er een automatisch manier zijn om alles in 1 keer te provisionen en zo het werk van dit team te verlichten.

Op die manier zou het per editie in plaats van alles opnieuw te kopiëren zijn eerder de play books eens doorlopen en aanvallen waar nodig. Een testopstelling kan dan ook op een korte tijd opgezet worden om de werking van bepaalde onderdelen van het netwerk te testen.

Mijn onderzoeksvragen sluiten dan ook vooral aan op de mogelijkheid van het gebruik van Ansible voor het automatiseren.

- Is het mogelijk om met Ansible een netwerk te provisionen op Cisco IOS apparatuur.
- In hoeverre is de implementatie volwassen om dit voor het dagelijks leven te gebruiken en dit te laten integreren bij bedrijven.
- Zijn er dingen die niet mogelijk zijn met Ansible.
- Wat is de minimumvereiste voor de apparatuur om aan de slag te kunnen gaan en playbooks te pushen vanaf Ansible.

1.3 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 3 wordt de basisinstelling van het Cisco apparaat vastgelegd en wat het te bereiken resultaat moet zijn.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Methodologie

2.1 Onderzoek naar mogelijkheden

Eerst en vooral werd er gezocht naar wat de mogelijkheden zijn de dag van vandaag in vergelijking met een jaar geleden. Omdat dit vooral rond Ansible draait is er dus eerst naar dat onderdeel gekeken. Het is belangrijk dat het onderzoek toekomstgericht is en dus ook met de laatste mogelijkheden werkt. Omdat er heel veel vooruitgang geboekt wordt binnen deze sector is het dus cruciaal dat dit documentatie gebruikt die zo up-to-date mogelijk is.

Hiervoor werd er gezocht op verschillende fora, er werd gekeken naar de blogposts van Ansible zelf en heel vaak wordt er ook gekeken op Github repositories. Deze zijn vaak een heel goede indicatie van in hoeverre een project nog actief ondersteund wordt en deze nog verder ontwikkeld wordt. Iets die meer dan een jaar geen commit meer heeft ontvangen kan je zo goed als niet nuttig beschouwen in vergelijking met de projecten die de laatste jaar op gang zijn gekomen.

Heel vaak werd er ook door documentatie gegaan om te kijken wat mogelijk was met welke tools om te zien of dit voldoet aan wat we willen bereiken. Goede documentatie is ook een belangrijke factor om het project goed te kunnen gebruiken in het verdere proces. Ook maakt dit ook heel veel uit om de keuze voor een bepaald pakket eenvoudiger te maken.

2.2 Uitwerking basis configuratie

Om het configureren op de test te stellen moet er eerst een basis configuratie voor handen zijn. Dit is hoe het eindresultaat zou moeten zijn na het uitvoeren van gelijk welke automatisatie methode. Als dit niet zo is dan kunnen we beslissen dat het gebruik van een bepaalde techniek niet geschikt is voor dagelijks gebruik of implementatie in een bedrijfsinfrastructuur.

De basisconfiguratie zal zich richten op Cisco IOS apparatuur meer bepaald wat voor handen is. Hoofdzakelijk zal er gewerkt worden met een 2960 series switch die voor dit onderzoek ten allen tijde beschikbaar was. De configuratie zal vooral inwerken op enkel basiscommando's die in het dagelijks leven het meest gebruikt worden. Hieronder valt bijvoorbeeld VLAN tagging en bijvoorbeeld ip adressering.

2.3 Automatiseren van configuratie

Als laatste hebben we dan nog het belangrijkste aspect van dit onderzoek, dit zijnde het automatiseren van de configuratie die bereikt is met de bovenstaande methode. Er zal gestart worden van een bepaalde basisconfiguratie, dit het liefst zo dicht mogelijk bij een factory reset. Dus er wordt ook gekeken naar hoeveel werk er op voorhand nodig is om alles werkende te kunnen krijgen via een automatisch platform zoals Ansible. Ook moet er rekening gehouden worden met de configuratie in combinatie met Ansible. Hoeverre kan het automatiseren voor problemen zorgen bijvoorbeeld als Ansible een ip adres verandert en zichzelf zo buitensluit bijvoorbeeld. Dit probleem heeft dan ook te maken met dat Ansible voor netwerkapparatuur over SSH loopt die een geldige ip adressering nodig heeft om te kunnen werken.

Net zoals het configureren van een server zal er ook eerst gekeken worden naar de stappen die manueel nodig zijn om het resultaat te behalen. Daarna zullen deze stappen overgezet worden in 'Tasks' zoals dit binnen Ansible noemt. Het voordeel vergeleken met een server role is dat de stappen bij een switch en router veel eenduidiger zullen zijn. Het installeren van een DNS server op Linux is net iets ingewikkelder dan de commando's laten uitvoeren binnen Ansible die nodig zijn om een Cisco apparaat te configureren.

3. Configuratie zonder automatisatie

Voor de testen zal er gebruik gemaakt worden van een Cisco IOS switch geproduceerd in 2006. Deze is een 2960 Series switch en meer bepaald de C2960-24LT. Deze heeft 8 poorten met POE en 2 Gigabit interfaces. De versie die op de switch geïnstalleerd staat is 'Version 12.2(44r)SE1'.

Bij de configuratie van de switch zijn er drie grote lijnen die we kunnen zien: eerst en vooral het instellen van de basisinstellingen. Dit bestaat uit het configureren van een wachtwoord, hostname en het instellen van standaardwaarden zoals een 'no ip domain-lookup'. Daarna wordt er een configuratie aangemaakt met enkele vlans. Deze krijgen een ip adres en een beschrijving. Als laatste onderdeel hebben we dan nog het instellen van een VTP (VLAN Trunk Protocol) server op het apparaat. Dit allemaal samen zou al goed moeten aantonen in hoeverre het mogelijk is om deze zaken te automatiseren.

3.1 Voorbeeld configuratie

Voor de configuratie van de Switch is er het voorbeeld gevolgd van een oefening uit de CCNA cursus in combinatie met een blogpost die beschrijft hoe je SSH openzet op cisco apparatuur. De SSH connectie is nodig om vanaf Ansible scripts te kunnen laten uitvoeren. De opzet van een switch wordt door het benodigde onderdeel dan ook verdeeld in 2 onderdelen. Het eerste zijn de commando's om van de initiële configuratie naar de minimale configuratie te raken. Dan daarna hebben we het tweede onderdeel met commando's die nodig zijn om de complete configuratie van de apparatuur af te maken. Hierdoor hebben we als resultaat 3 verschillende configuraties in bijlage: Initial Config, Minimal Config en Complete Config.

3.1.1 Initiële configuratie

De initiële configuratie is deze die beschikbaar is zonder enige configuratie op de switch. Op de huidige switch stond er al wat configuratie dus werd deze teruggezet naar 'Factory Defaults'. Hiervoor zijn twee commando's nodig. Het ene commando verwijdert de huidige configuratie en de tweede verwijdert de vlan configuratie (indien deze aanwezig is) die op het moment aanwezig is op de switch. (**resetSwitch**)

```
switch1#write erase
Erasing the nvram filesystem will remove all configuration files!
    Continue? [confirm]
[OK]
Erase of nvram: complete
switch1#delete flash:vlan.dat
Delete filename [vlan.dat]?
Delete flash:vlan.dat? [confirm]
switch1#reload
System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm]
```

Na het uitvoeren van de commando's wordt nog even gevraagd om de reload van de switch te bevestigen (wat we ook doen). Als er gevraagd zou worden om configuratie wijzigingen op te slaan dan kiezen we voor 'no'. Hierna komt de switch in de staat terecht waarvan we uitgaan hoe deze in gebruik wordt genomen door gebruikers. Na het heropstarten zal deze dan ook vragen of we de initiële setup willen doorlopen.

3.1.2 Minimale configuratie

Omdat het nodig is om vanaf de switch op afstand te kunnen aanspreken is er nood aan ssh in combinatie met een management ip adres. Omdat we verder werken op de CCNA oefening is er gekozen om een ip adres met ip '172.17.1.2' in te stellen op vlan1. Verder wordt het crypto commando gebruikt om een RSA key aan te maken met een sterkte van 2048 bits omdat dit tegenwoordig grotendeels de standaard is en er zelf al vaak geopteerd wordt om keys van 4096 bits te nemen. Verder is het ook nodig om een gebruikersnaam en wachtwoord in te stellen om gebruik te kunnen maken van remote ssh logins. Dan pas zal het mogelijk zijn om een connectie te maken over ssh. Als laatste wordt hier ook nog de 'password-encryption' service geactiveerd om de beveiliging te optimaliseren en de wachtwoorden niet in clear text in de configuratie te laten staan.

```
Would you like to enter the initial configuration dialog? [yes/no]: no
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Switch(config)#ip default-gateway 172.17.1.1
Switch(config)#int vlan1
Switch(config-if)#ip address 172.17.1.2 255.255.255.0
Switch(config-if)#no shut
Switch(config-if)#exit
Switch(config)#ip domain-name website.be
Switch(config)#crypto key generate rsa
The name for the keys will be: Switch.website.be
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...[OK]

Switch(config)#username switchadmin password cisco
Switch(config)#enable secret class
Switch(config)#no ip domain-lookup
Switch(config)#line console 0
Switch(config-line)#logging synchronous
Switch(config-line)#login local
Switch(config-line)#line vty 0 4
Switch(config-line)#transport input ssh
Switch(config-line)#login local
Switch(config-line)#password cisco
Switch(config-line)#exit
Switch(config)#service password-encryption
Switch(config)#end
Switch#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Switch#reload
Proceed with reload? [confirm]
```

Na dat we deze configuratie hebben ingeladen is het mogelijk om via ssh connectie te maken met de switch en vanaf daaruit de configuratie af te werken. In ons voorbeeld weten we dat we poort 1-23 zullen veranderen van configuratie dus daarom connecteer je voor ssh best via ssh omdat er anders disconnects kunnen voorkomen tijdens het wijzigen van de configuratie op de switch. Als we poort 24 op het management vlan houden is er geen probleem. Daarom dat we ook hier al een ip adres instellen voor de default gateway en vlan1.

3.1.3 Complete configuratie

Als laatste hebben we dan nog de volledige configuratie. Dit zijn de commando's die uitgevoerd moeten worden om de volledige correcte configuratie te verkrijgen. Het is de bedoeling dat ieder automatisatie methode dezelfde De onderstaande commando's moeten uitgevoerd worden om alles werkende te hebben zoals het hoort. De uiteindelijke volledige configuratie zijn onderaan beschikbaar in de bijlagen.

```
Username: switchadmin
Password:
Switch>en
Password:
Switch#
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname switch1
switch1(config)#vtp mode server
Device mode already VTP SERVER.
switch1(config)#vtp domain Lab5
Changing VTP domain name from NULL to Lab5
switch1(config)#vtp password cisco
Setting device VLAN database password to cisco
switch1(config)#int range fa0/1-5
switch1(config-if-range)#switchport mode trunk
switch1(config-if-range)#switchport trunk native vlan 99
switch1(config-if-range)#no shutdown
switch1(config-if-range)#exit
switch1(config)#vlan 99
switch1(config-vlan)#name management
switch1(config-vlan)#vlan 10
switch1(config-vlan)#name faculty-staff
switch1(config-vlan)#vlan 20
switch1(config-vlan)#name students
switch1(config-vlan)#vlan 30
switch1(config-vlan)#name guest
switch1(config-vlan)#exit
switch1(config)#int vlan99
switch1(config-if)#ip address 172.17.99.11 255.255.255.0
switch1(config-if)#exit
switch1(config)#int range fa0/6-10
switch1(config-if-range)#switchport access vlan 30
switch1(config-if-range)#int range fa0/11-17
switch1(config-if-range)#switchport access vlan 10
switch1(config-if-range)#int range fa0/18-23
switch1(config-if-range)#switchport access vlan 20
switch1(config-if-range)#exit
```

```
switch1(config)#spanning-tree vlan 99 priority 4096
switch1(config)#end
*Mar  1 00:02:09.679: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG:
    VTP domain name changed to Lab5.
switch1(config)#end
*Mar  1 00:02:11.642: %LINEPROTO-5-UPDOWN: Line protocol on
    Interface Vlan99, changed state to down
switch1(config)#end
switch1#
*Mar  1 00:02:16.985: %SYS-5-CONFIG_I: Configured from
    console by switchadmin on console
switch1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
switch1#reload
Proceed with reload? [confirm]
```

Na het uitvoeren van bovenstaande commando's zijn er enkele vlans ingesteld op de switch en heeft deze de vtp server functionaliteit gekregen.

Bijlage: cisco excercise example.pdf
Bijlage: -does not exist yet- final configuration

3.2 'Automatisatie' of vereenvoudiging via ingebouwde mogelijkheden

Voor de configuratie van een Cisco apparaat kan vaak wel nog een basis methode gebruikt worden om alles te automatiseren. Wel moeten we hier bij zeggen dat deze methode veel grondigere moet manueel getest worden op kleine foutjes. Ook dit kunnen we nog opdelen in twee methodes.

3.3 Commando's in terminal venster kopiëren

Deze methode is vrij eenvoudig en is bij voorgaande voorbeelden al gebruikt geweest omdat het gelijk staat aan het letterlijk uitvoeren van bepaalde commando's. Ieder commando die je normaal zou uitvoeren staat in de configuratie in de correcte volgorde. Let op hierbij moet je ook rekening houden dat je een enter of dergelijke ook tussen je lijnen moet plaatsen om bepaalde zaken te confirmen.
Een voorbeeld hiervan vind je onderaan in de bijlagen. Dit is het template die hier gebruikt is geweest om de basis configuratie op te zetten waarmee Ansible zal vergeleken worden. Voor een apparaat is dit nog vrij haalbaar maar als je meerdere apparaten hebt in een netwerk waarbij je een voor een moet connecteren om dan een configuratie te plakken die

elke keer een klein beetje anders is. Dan is het nog hopen dat je een bepaald item niet vergeet te veranderen of fout verandert. Hier is dus heel veel plaats voor 'human error'. Voor de gebruikte configuratie op punt stond is de switch toch wel een paar keer gewist geweest om terug van vooraf te beginnen.

4. Conclusie

Lijst van figuren

Lijst van tabellen