



**HoGent**

Faculteit Bedrijf en Organisatie

Technische voor-en nadelen van Puppet en Ansible. Verloop en redenen van een omschakeling.

Thomas Detemmerman

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

Promotor:  
Harm De Weirdt  
Co-promotor:  
Tom De Wispelaere

Instelling: VRT

Academiejaar: 2016-2017

Tweede examenperiode



Faculteit Bedrijf en Organisatie

Technische voor-en nadelen van Puppet en Ansible. Verloop en redenen van een omschakeling.

Thomas Detemmerman

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

Promotor:  
Harm De Weirdt  
Co-promotor:  
Tom De Wispelaere

Instelling: VRT

Academiejaar: 2016-2017

Tweede examenperiode



# Samenvatting

TO DO



## Voorwoord





# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>9</b>
1.1	<b>Stand van zaken</b>	<b>9</b>
1.1.1	Profiel van Puppet	11
1.1.2	Profiel van Ansible	11
1.2	<b>Opzet van deze bachelorproef</b>	<b>12</b>
1.3	<b>Probleemstelling en Onderzoeksvragen</b>	<b>12</b>
1.3.1	Wat zijn de redenen van een omschakeling?	12
1.3.2	Wat zijn de technische voor-en nadelen van Puppet en Ansible?	12
1.3.3	Wat is het verloop van een dergelijk transitperiode?	13
<b>2</b>	<b>Methodologie</b>	<b>15</b>
2.1	<b>Redenen</b>	<b>15</b>

<b>2.2</b>	<b>Technische voor-en nadelen van Puppet en Ansible</b>	<b>15</b>
2.2.1	Infrastructuur met Vagrant . . . . .	15
<b>3</b>	<b>Conclusie</b> . . . . .	<b>19</b>
	<b>Bibliografie</b> . . . . .	<b>19</b>

# 1. Inleiding

## 1.1 Stand van zaken

Bedrijven kunnen tegenwoordig niet zonder IT-infrastructuur. Deze infrastructuur kan zeer uitgebreid en complex zijn. Bovendien moet ze ook nog schalen naarmate het bedrijf groeit. Als systeembeheerder heb je diverse taken zoals incident management, het volgen van de laatste technologische trends of maatregelen treffen tegen cyberdreigingen. Het opzetten en configureren van de zoveelste identieke server is een groot tijd- en geldverlies. Daarom werden configuration management tools in het leven geroepen. De eerst bekende tool was Puppet. Deze technologie stelde ons in staat om configuraties van servers als het ware te programmeren. Eens de gewenste configuratie geprogrammeerd is, kunnen extra gelijkaardige servers veel sneller opgezet worden. De systeembeheerders van weleer zijn nu meer een meer DevOp's geworden. Dit zijn dus mensen die zich niet enkel bezig houden met systeembeheer maar ook met software-ontwikkeling. Ze ontwikkelen als het ware de configuraties van servers. Puppet is daar altijd al marktleider in geweest. Dit is ook te zien op grafiek 1.1. Maar daar komt nu verandering in. Er is de laatste jaren meer concurrentie op de markt gekomen waaronder relatief bekenden zoals Salt en Chef. Echter, één van deze nieuwe CMT 's<sup>1</sup> doet het opvallend beter op gebied van populariteit en dat is Ansible inc. Zoals op de grafiek te zien is heeft Ansible in 2015 de leiding genomen. Het was bovendien ook in dat jaar dat Ansible werd vernoemd door multinationals waaronder Gartner, die over Ansible schreef in een artikel over 'Cool Vendors in DevOps' (**coolvendors**). Verder was het Red Hat die aankondigde dat er een akkoord was om Ansible over te nemen (**redhatovername**). Volgens grafiek 1.1 van Debian laat Ansible voorlopig zijn concurrenten ver achter zich. Maar wie zijn Puppet en Ansible nu eigenlijk?

---

<sup>1</sup> Configuration management tool



Figuur 1.1: Deze grafiek toont het aantal keer dat een bepaald softwarepakket geïnstalleerd is op een Debian distributie. (**popcon**)

### 1.1.1 Profiel van Puppet

Puppet werd ontwikkeld in 2005 met als doel het automatiseren van data centers (**puppetfaq**). Om dit te kunnen verwezelijken maakt puppet gebruik van het server/client model. De server wordt hierbij de puppetmaster genoemd. Dit kunnen er één of meerdere zijn. De client wordt de puppetagent genoemd. Zowel op de master als op de agent dient puppet geïnstalleerd te zijn om te kunnen functioneren. Tussen de master en de client bestaat er een vertrouwensrelatie die onderhouden wordt door certificaten. Het is de puppetmaster die verantwoordelijk is voor het verlenen van deze certificaten. Al deze communicatie verloopt bovendien via het HTTPS-protocol. Pas als dit alles in orde is, kan Puppet aan de configuraties van de clients beginnen. De code die je schrijft wordt een manifest genoemd. Wanneer een puppetagent wil controleren of hij nog up-to-date is, zal hij een catalogus aanvragen bij de puppetmaster. Een dergelijke catalogus is in feite een manifest dat de puppetmaster compileerd. Deze catalogus is bovendien uniek voor elke puppetagent. Dit komt omdat er bij het compileren van het manifest naar de catalogus rekening gehouden wordt met diverse parameters zoals de functie van de server of de distributie van het besturingssysteem dat op die server draait (**puppetlanguagecatalog**). Eens de puppetagent zijn persoonlijke catalogus ontvangen heeft, zal deze voor zichzelf controleren of er verschillen zijn tussen zijn huidige configuratie en de staat die beschreven staat in de catalogus. Indien er afwijkingen zijn, worden deze ook automatisch opgelost (**puppetdoc**).

### 1.1.2 Profiel van Ansible

Ansible is opgericht door Michael DeHaan, iemand die zeer vertrouwd was met Puppet (**ansiblefordevops**). Hij vond dat bedrijven die Puppet gebruikten, moeilijkheden ondervonden op gebied van eenvoud en automatisatie. Daarom is hij samen met Saïd Ziouani Ansible Inc. gestart. Ansible maakt geen gebruik van agenten. Dit betekent dat de ansibleserver enkel de naam en het wachtwoord dient te kennen van de servers die hij moet configureren. De code die beschrijft hoe deze servers geconfigureerd moeten worden zijn geschreven in YAML en de verzameling van al deze configuraties wordt een playbook genoemd. Wanneer Ansible een bepaalde server wenst te configureren wordt dit standaard verzorgd door het SSH-protocol. Het authenticeren kan op verschillende manieren. Er wordt aangeraden om gebruik te maken van een SSH-key, wat het eenvoudigst is, maar ook andere middelen zoals een ordinair wachtwoord of het kerberos-protocol worden ondersteund. Eens de verbinding tot stand is gebracht, verstuurt Ansible modules naar de te configureren server. Deze modules worden vervolgens uitgevoerd en weer verwijderd. Ook Ansible bezit de functionaliteit om na te gaan of de huidige configuratie in lijn is met de ontvangen modules. Om servers te configureren met Ansible bestaan er bovendien twee manieren. Ansible playbooks kunnen in principe verstuurd worden naar de servers vanaf elke computer. Voor een grotere hoeveelheid servers is dit echter niet aangeraden en bestaat er de commerciële versie waarbij de playbooks worden verstuurd vanaf een centraal punt. Dit centraal punt is voorzien van Ansible Tower en heeft een inventaris van alle servers en playbooks die onder zijn verantwoordelijkheid vallen (**ansibledoc**).

## 1.2 Opzet van deze bachelorproef

In dit onderzoek vallen kleinere CMT's zoals Chef en Salt buiten de scope en zal er bijgevolg de focus gelegd worden op Puppet en Ansible. Dit onderzoek vindt plaats op MediaIT, een afdeling binnen VRT. Momenteel wordt er gebruik gemaakt van Puppet maar deze voldoet niet aan de verwachtingen van de bussiness en daarom is er dan ook besloten om de huidige puppet-infrastructuur te vervangen door Ansible. Deze bachelorproef zal onderzoeken wat er precies foutgelopen was en waarom deze problemen zich hebben voorgedaan. Vervolgens zal er gekeken worden of Ansible deze problemen überhaupt kan oplossen. Dit rapport wil een hulp bieden aan bedrijven die dezelfde stappen overwegen zodat het op voorhand duidelijk is wat er verwacht kan worden, wat de mogelijkheden zijn en waar een CMT te kort schiet. Ansible stijgt drastisch in populariteit, zoveel is zeker, maar het is echter niet de eerste keer dat er een hype ontstond rond een onderwerp dat vervolgens een gigantische teleurstelling veroorzaakte bij een community. Inmiddels heeft Ansible zich uiteraard wel al kunnen bewijzen door positieve analyses te krijgen van RedHat en Gartner. Maar is het zo dat Ansible beter is dan Puppet die inmiddels al 12 jaar ervaring heeft?

## 1.3 Probleemstelling en Onderzoeksvragen

De overschakeling van Puppet naar Ansible is geen kleine stap en kan mogelijk voor veel complicaties zorgen. Daarom weet men best op voorhand wat er te wachten staat en zullen er in dit onderzoek verschillende relevante zaken onderzocht worden die kunnen worden opgedeeld in de volgende drie grote categorieën.

### 1.3.1 Wat zijn de redenen van een omschakeling?

Het is belangrijk te weten wat de drijfveren waren voor de beslissing om Puppet te vervangen door Ansible en dat is precies waar deze eerste categorie toe dient. Om een profiel van de situatie op te kunnen stellen zal een interview plaatsvinden met de verantwoordelijken binnen de VRT om zo te achterhalen waar Puppet te kort schoot en waarom men denkt dat Ansible hier een oplossing biedt. Als bedrijven hun situatie herkennen in dit profiel, is het geadviseerd om te overwegen of een overstap ook voor hen al dan niet aan te raden is.

### 1.3.2 Wat zijn de technische voor-en nadelen van Puppet en Ansible?

In deze tweede categorie zal er een vergelijkende studie plaatsvinden waarbij technische aspecten zoals performantie, schaalbaarheid en veiligheid vergeleken worden.

Ten eerste wordt de performantie onderzocht. Hieronder wordt verstaan de tijd die nodig is tot het bekomen van een consistente staat en deze zal onderzocht worden in twee situaties. Bij de eerste is er namelijk nog geen configuratie aanwezig en dient alles nog geïnstalleerd

en geconfigureerd te worden. Bij de tweede situatie is er wel al een configuratie aanwezig en is het de bedoeling dat de CMT enkel de nodige aanpassingen doorvoert en niet alles opnieuw configureerd.

Ten tweede is er de schaalbaarheid. Onder schaalbaarheid wordt verstaan: het vermogen om grote vraag te verwerken zonder kwaliteit te verliezen (**informit**). We zullen monitoren hoe Ansible en Puppet hun resources verdelen bij een toenemende drukte, hier onder de vorm van meer servers en uitgebreidere configuraties.

Er wordt afgesloten met een analyse over de veiligheid. Hierbij zal er een literatuurstudie plaatsvinden met onderzoek naar welke veiligheidsproblemen reeds gekend zijn en wat de impact hiervan is op een bedrijfsnetwerk. CMT's hebben namelijk administrator rechten tot verschillende servers die ze dienen te configureren. Wanneer de server waarop een CMT draait besmet is, kunnen de gevolgen catastrofaal zijn.

### 1.3.3 Wat is het verloop van een dergelijk transitperiode?

Problemen die bij de vervanging van Puppet door Ansible optreden, zullen gerapporteerd worden en er zal onderzocht worden waarom deze optraden. Al dan niet gevonden oplossingen zullen beschreven en uitgelegd worden zodat andere bedrijven zich goed bewust zijn van wat er te wachten staat en hoe ze eventueel sommige voorvallen best kunnen oplossen. Welke incidenten zich zullen voordoen, valt uiteraard moeilijk te voorspellen.





## 2. Methodologie

### 2.1 Redenen

TO DO (ook nog andere titel)

### 2.2 Technische voor-en nadelen van Puppet en Ansible

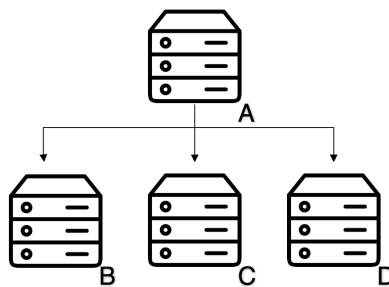
#### 2.2.1 Infrastructuur met Vagrant

Zowel Puppet als Ansible hebben een gelijkaardige infrastructuur, maar verschillen in de details. Om de resultaten van deze vergelijkende proef zo betrouwbaar mogelijk te maken is getracht de verschillen tussen beide opstellingen zo minimaal mogelijk te houden. Beide opstellingen zijn dan ook gebaseerd op de infrastructuur die te zien is op afbeelding 2.1.

In het geval van Puppet zal server A uit afbeelding 2.1 de PuppetMaster zijn. Het is deze server die de puppet manifesten bewaart en compileert naar catalogussen. In het geval van Ansible zal op server A Ansible Tower geïnstalleerd zijn en verder bevat deze server uiteraard het Ansible equivalent van manifesten, playbooks genaamd.

Server B t.e.m. D stellen de clients voor. Zij ontvangen een bepaalde configuratie van server A om vervolgens de nodige services te installeren en te configureren.

Om deze opstelling te kunnen verwezelijken en om gelijkheid tussen beide infrastructuren te garanderen wordt er gebruik gemaakt van Vagrant. Deze technologie maakt het mogelijk om op een eenvoudige manier herhaaldelijk dezelfde servers op te kunnen zetten. Ver-



Figuur 2.1: Infrastructuur

volgens worden deze servers voorbereid met de nodige configuraties zoals het installeren van de monitoringstool. Het is vanzelfsprekend dat de monitoringstool al geïnstalleerd is alvorens Puppet en Ansible de configuraties van de servers overnemen, want het is juist dit dat gemeten dient te worden, de impact van Puppet en Ansible op deze servers en het netwerk. Daarom is er ook beslist om de configuratie van de monitoringstool ook door Vagrant te laten gebeuren, zodat de monitoringstool reeds operationeel is als Ansible en Puppet in werking treden. De vagrantfile om dit te verwezenlijken ziet er als volgt uit.

#### Vagrantfile (Puppet infrastructuur)

```

1 AmountOfVM = 2
2 VmGroupName = "PPnode"
3 VagrantBox = "centos/7"
4
5 #Master
6 Vagrant.configure("2") do |config|
7   config.vm.provider "virtualbox" do |v|
8     v.memory = 5000
9     v.cpus = 2
10  end
11   config.vm.define "#{VmGroupName} Master" do |machine|
12     machine.vm.box = "#{VagrantBox}"
13     machine.vm.hostname = "#{VmGroupName} Master"
14     machine.vm.network "private_network", ip: "192.168.101.20"
15     config.vm.provision "shell",
16       inline: "systemctl restart network"
17     config.vm.provision "shell",
18       inline: "sudo sh /vagrant/puppetconfiguration/
19               installpuppetserver.sh"
19   end
20
21 #Clients
22 (1..AmountOfVM).each do |machine_id|
23   config.vm.provider "virtualbox" do |v|
24     v.memory = 500
25     v.cpus = 1
26   end
27   config.vm.define "#{VmGroupName}#{machine_id}" do |machine|
28     machine.vm.box = "#{VagrantBox}"
29     machine.vm.hostname = "#{VmGroupName}#{machine_id}"
  
```

```

30     machine.vm.network "private_network", ip: "192.168.101.#{20+
        machine_id}"
31     config.vm.provision "shell",
32       inline: "systemctl restart network"
33     config.vm.provision "shell",
34       inline: "sudo sh /vagrant/monitoring/newrelicinfra.sh"
35     config.vm.provision "shell",
36       inline: "sudo sh /vagrant/puppetconfiguration/
        installpuppetagent.sh"
37   end
38 end
39 end

```

Een woordje uitleg bij deze Vagrantfile:

Lijn 1 en 2 zijn variabelen. 'AmountOfVM' stelt het aantal client VM's voor. In totaal zullen er dus altijd AmountOfVM + 1 virtuele machines aangemaakt worden. Deze '+ 1' is afkomstig van de master welke standaard aangemaakt wordt.

Lijn 6 t.e.m.19 betreffen de master VM, net als server A uit afbeelding 2.1. De reden van lijn 14 en 15 is om het probleem op te lossen waarbij de interface, die aangemaakt wordt op lijn 13, niet automatisch gestart wordt. Lijn 17 en 18 starten het script dat de nodige software installeert om deze server als PuppetMaster te laten opereren.

Lijn 22 t.e.m 37 betreffen de clients, net als server B, C, D uit afbeelding 2.1. Dankzij de variable 'AmountOfVM' kunnen deze dynamisch bijgebouwd worden. Lijn 22 t.e.m 25 zorgen ervoor dat alle VM's dezelfde resources ter beschikking krijgen. Deze resources zijn bewust laag gehouden zodat fluctuaties in de monitoringstool beter waargenomen kunnen worden. Lijn 33 en 34 zorgen voor de opstart van het script newrelicinfra.sh. Dit script installeert New Relic, de gekozen monitoringstool. Hiermee zijn we in staat om data van de VM te analyseren vanop het online platform van New Relic. Lijn 35 en 36 starten het script genaamd installpuppetagent.sh. Dit script bestaat uit de volgende stappen:

### installpuppetagent.sh

```

1  #!/bin/bash
2  rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
3  yum install puppet -y
4  yes | cp /vagrant/puppetconfiguration/hosts /etc/hosts
5  yes | cp /vagrant/puppetconfiguration/puppet-agent.conf /etc/puppet/
    puppet.conf
6  systemctl start puppet
7  puppet agent -t

```

Hierbij wordt er eerst de repository van Puppet toegevoegd om vervolgens Puppet te installeren. Hierna wordt de bestaande 'hosts-file' vervangen door een nieuwe. Dit is vanwege de afwezigheid van een DNS-server in onze opstelling. Ook wordt de Puppet.conf aangepast. Hierbij voegen we de FQDN van de PuppetMaster toe. Om te eindigen starten we de service en doen we een aanvraag bij de master voor een certificaat. Dit is nodig omdat zonder certificaat er geen vertrouwensrelatie bestaat tussen de master en de client. Bijgevolg kan er dan ook geen catalogus verstuurd worden naar deze client.

Ondanks het feit er getracht is geweest om de verschillen zo klein mogelijk te houden, blijven het verschillende technologieën. Hieronder is een lijst opgesomd van de lijnen die niet overeenkomen met Puppet.

**Vagrantfile** (Ansible infrastructuur)

```
2 VmGroupName ="Anode"
11 machine.vm.box ='ansible/tower'
14 machine.vm.network "private_network", ip: "192.168.100.20"
17 #NIET AANWEZIG
18 #NIET AANWEZIG
29 machine.vm.network "private_network", ip: "192.168.100.#{20+machine_id}"
35 # NIET AANWEZIG
36 #NIET AANWEZIG
```

De afwezigheid van lijn 17 en 18 bij Ansible is te wijten aan het feit dat de installatie van Ansible Tower vervat zit in de vagrantbox op lijn 11. Verder is het opmerkelijk dat ook lijn 35 en 36 niet aanwezig zijn. Dit is dan ook een eerste belangrijk verschil dat al in deze vroege fase duidelijk wordt. Namelijk de afwezigheid van het script `installpuppetagent.sh`. Dit script zorgt voor de installatie van de puppetagent op de clients. Aangezien Ansible geen gebruik maakt van agenten is een dergelijk script ook niet aan de orde.

### 3. Conclusie

TO DO



## Lijst van figuren

- 1.1 Deze grafiek toont het aantal keer dat een bepaald softwarepakket geïnstalleerd is op een Debian distributie. (**popcon**) ..... 10
- 2.1 Infrastructuur ..... 16





## Lijst van tabellen