



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

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus.

Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Voorwoord

Inhoudsopgave

1	Inleiding	9
1.1	Stand van zaken	9
1.1.1	Profiel van Puppet	11
1.1.2	Profiel van Ansible	11
1.2	Opzet van deze bachelorproef	11
1.3	Probleemstelling en Onderzoeksvragen	12
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
2	Methodologie	15
2.1	Redenen	15
2.2	Technische voor-en nadelen van Puppet en Ansible	15

3	Conclusie	19
	Bibliografie	20

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 en ook het volgen van de laatste technologische trends en bedreigingen. 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's¹ 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?

¹ 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 servers 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 wordt door de puppetmaster gecompileerd en is bovendien voor elke puppetagent persoonlijk. Dit komt omdat er bij het compileren van het manifest naar de catalogus rekening gehouden wordt met diverse zaken 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, stuurt Ansible modules naar de te configureren server. Deze modules worden vervolgens uitgevoerd en weer verwijderd. (**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 business 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 drie volgende 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 de tweede categorie zal er een vergelijkende studie plaatsvinden waarbij technische aspecten zoals performantie, schaalbaarheid in veiligheid vergeleken worden.

Ten eerste wordt er performantie onderzocht. Hieronder wordt verstaan de tijd die nodig is tot het bekomen van een consistente staat. Dit wordt onderzocht in de volgende twee situaties. Bij de eerste is er 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 configureert.

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 in de eerste plaats een literatuurstudie plaatsvinden met onderzoek naar welke veiligheidsproblemen reeds gekend zijn en wat de impact hiervan is op onze systemen. CMT's hebben immers administrator rechten tot alle servers die ze dienen te configureren. Wanneer de server waarom de 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. Welke incidenten zich zullen voordoen, valt uiteraard moeilijk te voorspellen.

2. Methodologie

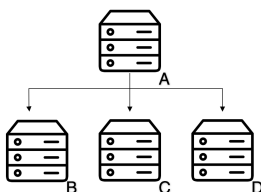
2.1 Redenen

2.2 Technische voor-en nadelen van Puppet en Ansible

Zowel Puppet als Ansible hebben een gelijkaardige infrastructuur, maar verschillen in de details. Om de resultaten van de 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 manifests bewaart en compileert naar catalogs. Bij Ansible zal deze server AnsibleTower geïnstalleerd hebben en verder bevat deze server uiteraard het ansible equivalent van manifests, 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.



Figuur 2.1: Infrastructuur

Om deze opstelling te kunnen verwezelijken en om gelijkheid tussen beide infrastructuren te garanderen werd er gebruik gemaakt van Vagrant. Deze technologie staat ons in staat om eenvoudig servers op te zetten. Vervolgens werden deze servers voorbereid met de nodige configuraties zoals het installeren van de monitoringstool. Het is vanzelfsprekend dat de monitoringstool al operationeel is alvorens Puppet en Ansible de configuraties overnemen. Want het is juist dit dat gemonitord dient te worden. Daarom is beslist om de configuratie van de monitoringstool door Vagrant uit te laten voeren.

Vagrantfile Puppet infrastructuur

```

1 AmountOfVM = 2
2 VmGroupName = "PPnode"
3
4 #Master
5 Vagrant.configure("2") do |config|
6   config.vm.provider "virtualbox" do |v|
7     v.memory = 5000
8     v.cpus = 2
9   end
10  config.vm.define "#{VmGroupName} Master" do |machine|
11    machine.vm.box = "bertvv/centos72"
12    machine.vm.hostname = "#{VmGroupName} Master"
13    machine.vm.network "private_network", ip: "192.168.101.20"
14    config.vm.provision "shell",
15      inline: "systemctl restart network"
16  end
17
18 #Clients
19 (1..AmountOfVM).each do |machine_id|
20   config.vm.provider "virtualbox" do |v|
21     v.memory = 500
22     v.cpus = 1
23   end
24   config.vm.define "#{VmGroupName}#{machine_id}" do |machine|
25     machine.vm.box = "bertvv/centos72"
26     machine.vm.hostname = "#{VmGroupName}#{machine_id}"
27     machine.vm.network "private_network", ip: "192.168.101.#{20+
28       machine_id}"
29     config.vm.provision "shell",
30       inline: "systemctl restart network"
31     config.vm.provision "shell",
32       inline: "sudo sh /vagrant/monitoring/newrelicinfra.sh"
33     config.vm.provision "shell",
34       inline: "sudo sh /vagrant/puppetagent/installpuppetagent.sh"
35   end
36 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 machine aangemaakt worden. Deze '+ 1' is afkomstig van de master welke standaard aangemaakt wordt.

Lijn 4 t.e.m.16 betreffen de master VM, oftewel 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 18 t.e.m.35 betreffen de clients, oftewel server B, C, D uit afbeelding 2.1. Dankzij de variabele 'AmountOfVM' kunnen deze dynamisch bijgebouwd worden. Lijn 20 t.e.m.22 zorgen ervoor dat alle VM's dezelfde resources ter beschikking krijgen. Deze resources zijn bewust laag gehouden zodat fluctuaties in de monitoringstool beter zouden opvallen. Lijn 30 en 31 zorgen voor de opstart van het script newrelicinfra.sh. Dit script installeert New Relic, de gekozen monitoringstool. Hiermee zijn we in staat om data te analyseren vanop het online platform van New Relic. Lijn 32 en 33 starten het script genaamd installpuppetagent.sh. Hiermee wordt puppet geïnstalleerd, gestart om vervolgens automatisch een certificaat aan te vragen bij de PuppetMaster. Dit komt omdat alleen gecertificeerde PuppetClients configuraties van de PuppetMaster kunnen ontvangen.

Ondanks er getracht is het zo gelijk mogelijk te houden zijn er wel enkele kleine verschillen, het blijven immers verschillende technologieën.

```
2 VmGroupName =/ "Anode/"
11 machine.vm.box = /ansible/tower"
13 machine.vm.network "private_network", ip: "192.168.100.20"
27 machine.vm.network "private_network", ip: "192.168.100.#{20+machine_id}"
32 # NIET AANWEZIG
33 #NIET AANWEZIG
```

Een eerste belangrijk verschil wordt al in deze vroege fase duidelijk. Namelijk de afwezigheid van installpuppetagent.sh in het geval van de Ansibleinfrastructuur. Opvallend is ook dat hier geen alternatief script voor nodig is. Ansible maakt namelijk geen gebruik van agents. Het installpuppetagent.sh bestaat uit de volgende stappen:

```
1 #!/bin/bash
2 rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
3 yum install puppet -y
4
5 yes | cp /vagrant/puppetagent/hosts /etc/hosts
6
7 yes | cp /vagrant/puppetagent/puppet.conf /etc/puppet/puppet.conf
8
9 systemctl start puppet
10
11 puppet agent -t
```

Hierbij wordt er eerste de repository van Puppet toegevoegd om het vervolgens te installeren. Hierna plaatsen we een nieuwe hosts file op de server aangezien 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.

3. Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit

lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetur libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem.

Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.

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 15

Lijst van tabellen