



Faculteit Bedrijf en Organisatie

Onderzoek naar samenwerking tussen Cloud-Init en Ansible

Maarten De Smedt

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

Promotor:  
Lotte Van Steenberghe  
Co-promotor:  
Simon Lepla

Instelling: Be-Mobile

Academiejaar: 2018-2019

Tweede examenperiode



Faculteit Bedrijf en Organisatie

Onderzoek naar samenwerking tussen Cloud-Init en Ansible

Maarten De Smedt

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

Promotor:  
Lotte Van Steenberghe  
Co-promotor:  
Simon Lepla

Instelling: Be-Mobile

Academiejaar: 2018-2019

Tweede examenperiode



# Woord vooraf

Maken als bachelorproef helemaal gedaan is.



# Samenvatting

Maken als bachelorproef klaar is.

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.



# Inhoudsopgave

<b>1</b>	<b>Inleiding .....</b>	<b>15</b>
1.1	Context	15
1.2	Probleemstelling - Onderzoeksvraag	16
1.3	Opzet van deze bachelorproef	16
<b>2</b>	<b>Inleiding tot Ansible en cloud-init .....</b>	<b>19</b>
<b>2.1</b>	<b>Ansible</b>	<b>19</b>
2.1.1	Architectuur .....	19
2.1.2	Playbook - Roles .....	20
2.1.3	Omgevingen .....	21
<b>2.2</b>	<b>Cloud-Init</b>	<b>21</b>
2.2.1	Modules .....	21
2.2.2	User Data - Cloud config .....	23

2.2.3	Omgevingen .....	24
2.2.4	Systemen .....	24
<b>3</b>	<b>Literatuurstudie .....</b>	<b>25</b>
<b>3.1</b>	<b>An introduction to server provisioning with CloudInit</b>	<b>25</b>
3.1.1	Verschil en samenwerking met Ansible volgens Viktor Petersson .....	25
3.1.2	Setup cloud config bestand .....	26
<b>3.2</b>	<b>Using Ansible to Bootstrap My Work Environment Part 4</b>	<b>28</b>
3.2.1	Ansible initial playbook .....	29
3.2.2	Cloud-init .....	29
3.2.3	Ansible follow up playbook .....	31
<b>4</b>	<b>Methodologie .....</b>	<b>39</b>
<b>5</b>	<b>Opzetten lokale testomgeving .....</b>	<b>41</b>
<b>6</b>	<b>Opzetten Hetzner Cloud testomgeving .....</b>	<b>43</b>
<b>7</b>	<b>Basisconfiguraties op de servers .....</b>	<b>45</b>
<b>8</b>	<b>Server installatie en configuratie .....</b>	<b>47</b>
<b>9</b>	<b>Instellingen aanpassen na het opstarten .....</b>	<b>49</b>
<b>10</b>	<b>Container &amp; Cluster Configuratie .....</b>	<b>51</b>
<b>11</b>	<b>Ansible &amp; cloud-init met golang .....</b>	<b>53</b>
<b>12</b>	<b>Conclusie .....</b>	<b>55</b>

<b>A</b>	<b>Onderzoeksvoorstel</b>	<b>57</b>
A.1	Introductie	57
A.2	Stand Van Zaken	58
A.3	Methodologie	58
A.4	Verwachte resultaten	59
A.5	Verwachte conclusies	59
	<b>Bibliografie</b>	<b>61</b>



## Lijst van figuren

2.1	Voorbeeld van een Ansible playbook. ....	20
2.2	Ansible Galaxy site met lijst van roles. ....	21
2.3	Voorbeeld User-Data Script. ....	23
2.4	Voorbeeld Cloud Config Data. ....	24



## Lijst van tabellen





# 1. Inleiding

In dit hoofdstuk wordt er een kort inleiding over de bachelorproef. De oorsprong van het idee en de onderzoeksvraag wordt besproken. Ook wordt er al wat basisinfo gegeven over het onderwerp.

De installatie en modificatie van software servers moet voor de gebruiker altijd makkelijker en sneller. Eens de gebruiker weet wat voor server hij wil, wil hij deze liefst zo snel mogelijk opzetten met de nodige specificaties. Of als de gebruiker een kleine aanpassing wil doen aan de server, wil hij dit zo makkelijk mogelijk aanpassen.

Om dit zo efficiënt mogelijk te doen heb je configuration management tools nodig. Dit zijn tools die gemaakt zijn om software op servers te installeren en te beheren. De meest bekende tools zijn Chef, Puppet, Salt en Ansible. In deze bachelorproef gaat het onder andere over Ansible.

## 1.1 Context

Ansible is op zich al een zeer goede configuration management tool, maar voor het bedrijf Be-Mobile nog niet goed genoeg.

Be-Mobile is een Big Data verkeersbedrijf. Be-Mobile wil verkeer revolutioneren en de mobiliteits oplossingen voor morgen en vandaag creëren. Hun hoofdzetel ligt in Melle, bij Gent.

Be-Mobile werkt met "Hetzner Cloud". Dit is hun cloud server provider. Met hetzner cloud heb je de optie om te verwijzen naar een cloudconfig file om je server te configureren.

Het cloudconfig bestand is het configuratie bestand van cloud-init.

Cloud-init is net zoals een Ansible een soort van configuration management tool maar speciaal voor cloud servers. Door middel van paramaters in te vullen in dit bestand kan je jouw server configureren.

## 1.2 Probleemstelling - Onderzoeksvraag

Het probleem is meteen heel duidelijk. Moet er worden overgestapt naar cloud-init in plaats van Ansible. In deze bachelorproef gaat dit worden onderzocht. Waar zijn Ansible en cloud-init verschillend, waar zijn ze hetzelfde en waar vullen ze mekaar aan. De echte onderzoeksvragen waar deze thesis een antwoord op hoopt te vinden is:

- Is Ansible overbodig door het gebruikt van Cloud-init?
- Zijn Ansible en cloud-init compatibel?
- Op welke manier zijn ze compatibel of overbodig?

## 1.3 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een inleiding gegeven tot Ansible en cloud-init

In Hoofdstuk 3 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 4 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 5 worden lokale testomgevingen opgezet doormiddel van Vagrant en Virtual-box.

In Hoofdstuk 6 worden testomgevingen opgezet doormiddel van Hetzner Cloud.

In Hoofdstuk 7 worden er basisconfiguraties gedaan op alle servers (aanmaken users, package installeren,...)

In Hoofdstuk 8 worden er verschillende soorten servers(file, web, dhcp, ...) gemaakt met de testomgevingen.

In Hoofdstuk 9 wordt er gekeken hoe je instellingen aanpast na het opstarten van de server.

In Hoofdstuk 10 wordt gekeken de configuratie containers en clusters wordt ondersteunt.

In Hoofdstuk 12, 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. Inleiding tot Ansible en cloud-init

Dit hoofdstuk is een inleiding tot cloud-init en Ansible. Eerst wordt Ansible besproken. Er wordt eerst wat algemene uitleg gegeven, daarna wordt iets dieper op ingegaan op de architectuur, playbooks, roles en de omgeving. Daarna wordt er voor cloud-init hetzelfde gedaan. Eerst wordt er wat algemene uitleg gegeven. Daarna wordt er dieper ingegaan op de modules, Cloud config, omgevingen en systemen.

### 2.1 Ansible

Ansible is een open source IT configuration management en deployment tool. Ansible hun grote doel is om besturingssysteem configuratie en de implementatie van software allemaal onder 1 systeem. De informatie werd gevonden met behulp van het document *Ansible In Depth* (RedHat, 2017).

Ansible staat bekend als een systeem dat makkelijk te leren is als IT administrator, ontwikkelaar of manager. Het probeert er voor te zorgen dat het makkelijk te verstaan is en makkelijk om zelf op te bouwen. Zodat nieuwe gebruikers dit makkelijk kunnen oppikken. Ze proberen uniek te zijn door dingen door val aanpassing mogelijk te geven aan gebruikers voor de expert-gebruikers. Maar toch net zo toegankelijk voor de nieuwe gebruiker.

#### 2.1.1 Architectuur

Eén van de belangrijkste verschillen tussen Ansible en andere configuratie management tools, is zijn architectuur. Ansible gaat uit van het “push” model. Ook is er geen additionele software nodig om machines bruikbaar te maken voor Ansible. Het heeft geen extra

gebruikers of referenties nodig om te draaien. Het gebruikt gewoon de informatie die de user meegeeft. Daarbij hoort ook dat Ansible geen administrator of sudo toegang nodig heeft. Ansible wordt standaard bestuurd door een remote computer.

Dit zorgt ervoor dat Ansible veiliger wordt. Door alleen de informatie die de gebruiker meegeeft te gebruiken. Iemand die wel toegang heeft tot de server kan maar niet tot de remote computer kan geen aanpassingen pushen.

### 2.1.2 Playbook - Roles

Ansible voert de automatisatie en deployment uit via playbooks. Dit zijn yaml bestanden die beschrijven hoe de automatisatie moet verlopen.

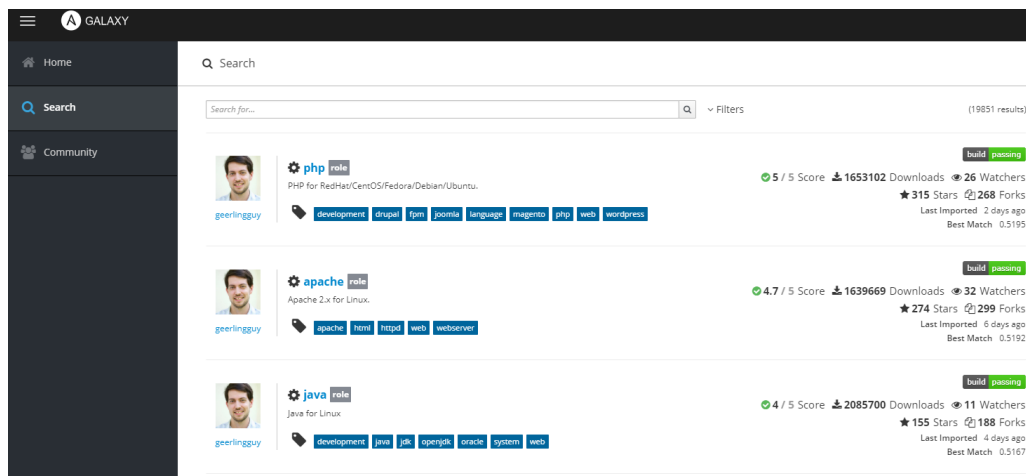
Deze playbooks bevatten verschillende “plays” die de automatisatie definiëren over verschillende hosts. Deze hosts staan bekend als de inventory. Elke “play” bevat verschillende taken die één, verschillende of alle hosts moeten uitvoeren. Elke taak roept een Ansible module aan, een klein stukje code dat een specifieke taak uitvoert. Deze kan kunnen zeer simpel zijn, een bestand op een machine zetten of een specifieke package installeren. Maar ze kunnen ook complex zijn zoals een gehele CloudFormation opstarten in Amazon EC2.

```
1 ---
2 - hosts: webservers
3   sudo: yes
4
5   vars:
6     app_name: PleaseDeployMe
7     repo_url: https://github.com/username/repo_name.git
8     repo_remote: origin
9     repo_version: master
10    webapps_dir: /deployed
11    virtualenv_root: /deployed/PleaseDeployMe/mac
12  tasks:
13
14    - name: git pull project
15      git: repo={{repo_url}} dest={{webapps_dir}}/{{app_name}} version=master
16
17      notify:
18        - restart app
19
20    - name: install things
21      pip: name=virtualenv
22
```

Figuur 2.1: Voorbeeld van een Ansible playbook.

Ansible is geschreven zodat als ze de playbook uitvoeren ook checken of deze task nog moet gedaan worden. Bijvoorbeeld als een Ansible taak is om een webserver op te starten, zal Ansible deze alleen uitvoeren als de webserver nog niet is opgestart. Dit staat bekend als idempotente. Het zorgt ervoor dat de configuratie altijd snel en efficiënt wordt uitgevoerd.

Met Ansible kunnen taken ook ingekapseld worden in een role. Dit wordt gebruikt als er een specifieke configuratie meerdere keren wordt uitgevoerd, bijvoorbeeld het opzetten van een webserver. De Ansible Galaxy site bevat veel roles die kunnen gebruikt en aangepast worden voor het gebruik in een playbook.



Figuur 2.2: Ansible Galaxy site met lijst van roles.

### 2.1.3 Omgevingen

Ansible is even makkelijk te deployen in publieke of private cloud omgevingen, als in een lokale omgeving. Voor publieke of private cloud providers kan er gekozen worden voor: Amazon Web Services, Microsoft Azure, Rackspace,... Maar er kan ook op lokale infrastructures gewerkt worden door middel van virtuele machines. Tools die hiervoor worden gebruikt zijn: VirtualBox, VMWare,...

## 2.2 Cloud-Init

Net als Ansible is cloud-init ook een soort van configuration manager en deployment tool. Op de site van cloud-init (Moser & Harlow, 2019) staat het ontstaan. Het is ontwikkeld en uitgebracht als gratis software met de open source licentie en ook de Apache Version 2.0 licentie. De makers, Scott Moser en Joshua Harlow, hebben het in python geschreven.

Cloud-init zorgt voor de customisaties tijdens het opstarten van de cloud of virtuele instanties. Deze service gebeurt heel vroeg in het boot proces. De instantie zoekt naar de *user data* (het cloud-init script) van de gebruiker en voert deze uit.

Veel van de punten die hier verder worden besproken zijn gevonden in de presentatie van (Skinner & Heldebrant, 2017).

### 2.2.1 Modules

Cloud-init heeft 6 hoofdpijlers waar het modules voor heeft, namelijk: schijf configuratie, commando's uitvoeren, gebruikers en groepen creëren, beheren van packages, content bestanden schrijven en bootstrappen van Chef en/of Puppet. Er zijn nog andere modules maar dit zijn de 6 hoofdpijlers van cloud-init en daarbij de meeste gebruikte modules. Er kunnen ook zelf modules toegevoegd worden door deze in Python te schrijven. Met behulp

van de documentatie van (cloud-init.io, 2019) wordt er per pijler uitleg gegeven.

### Schijf configuratie

Er is 1 module die wordt gebruikt voor de schijf configuratie, namelijk **Disk Setup**. Via deze module kunnen er simpele partities en bestandssystemen worden geconfigureerd. Via de *device\_aliases* richtlijnen kunnen er aliassen worden gemaakt voor de block devices. Zodat er makkelijker naar deze kan worden verwezen. Via de *disk\_setup* richtlijn wordt de partitie configuratie gedaan. De *table\_type* richtlijn wordt gebruikt om de partitie tabel mee te geven. Ten laatste is er ook de *fs\_setup* richtlijn deze wordt gebruikt om de systeembestand configuratie te doen.

### Commando's uitvoeren

Voor het uitvoeren van commando's zijn er 2 modules: **Runcmd** en **Bootcmd**. Beide bevatten maar 1 richtlijn namelijk *runcmd* en *bootcmd*. Bij *runcmd* worden de commando's die worden meegegeven elke keer uitgevoerd als het script wordt gedraaid. Bij *bootcmd* enkel alleen als de instantie wordt opgestart. Ook worden de commando's bij *bootcmd* veel vroeger in het bootproces uitgevoerd.

### Gebruikers en groepen

Ook voor Gebruikers en groepen is er slechts 1 module: **Users and Groups**. Groepen kunnen worden toegevoegd door deze aan de richtlijn *groups* toe te voegen samen eventuele configuratie per groep. Gebruikers kunnen dan weer toegevoegd worden door deze aan de richtlijn *users* toe te voegen. Ook bij de gebruikers kan er nog extra configuratie gedaan worden per gebruiker.

### Packages

Voor het beheren en configureren van packages zijn er verschillende modules: **Apt Configure**, **Apt Pipelining**, **Package Update Upgrade Install**, **Snap**, **Snappy** en **Yum Add Repo**. Er kunnen packages geïnstalleerd en geconfigureerd worden via Yum en Apt. Meestal ondersteund een besturingssysteem Yum of Apt. Voor installatie van een package moeten deze plaatsen bij de richtlijn *packages*. Cloud-init weet zelf of het met yum of apt wordt gedaan. De configuratie en het toevoegen van package repo's gebeurt via Apt Configure, Apt Pipelining en Yum Add Repo. Ook ondersteunt cloud-init de installatie van snap packages. Dit zijn gecontaineriseerde packages. Via Snappy kan deze geconfigureerd worden.

### Content bestanden

Voor het aanmaken van bestanden is er de module **Write Files**. In de richtlijn *write\_files* word de gecodeerde inhoud van het bestand gezet met het pad.



## Chef - Puppet

Ook zijn er 2 modules die Chef en Puppet installeren, configureren en starten. Deze modules zijn logischer wijs **Chef** en **Puppet**. Voor Puppet wordt de richtlijn *puppet* meegegeven. Daaronder worden alle configuraties geplaatst. Chef werkt op dezelfde manier.

### 2.2.2 User Data - Cloud config

Zoals Ansible zijn playbook heeft, heeft cloud-init zijn user data. Ook heeft cloud-init de meta-data, maar deze wordt meegegeven door het cloud platform zelf. Dit zijn bijvoorbeeld de server naam en het server id. De 2 bekendste methodes om user data mee te geven zijn User-Data Script en Cloud Config Data.

#### User-Data Script

Het User-Data script een Linux script dat de server overloopt voor dat hij opstart. Het moet altijd beginnen met *#!/* of *Content-Type: text/x-shellscript*. Als er echt wordt gebruikt gemaakt van cloud-init wordt deze methode niet gebruikt. De modules kun je via zo een script bijvoorbeeld niet aanroepen. Dit is eerder voor gebruikers die al een script hebben. In plaats van dit om te zetten naar Cloud Config Data, kunnen ze dat dan script gebruiken.

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Figuur 2.3: Voorbeeld User-Data Script.

#### Cloud Config Data

De meest gebruikte methode is Cloud Config Data. Dit is een yaml bestand met de configuratie van de server in. Een Cloud Config bestand begint altijd met *#cloud-config*. In het bestand worden de modules opgelijst die worden gebruikt en daaronder dan hun configuraties. Dit bestand is veel overzichtelijker dan een script om later aan te passen of om te beheren. Ook heeft het wat gelijkenissen met het playbook van Ansible, doordat dit beide yaml bestanden zijn.

```
#cloud-config

package_upgrade: true

users:
  - default
  - name: maarten
    passwd: $6$Y5N.ote2cyDmb0l0$EizY59wJI4hMQwyFXIRpJtk.nqlntp5b4VpABWUGwXU.
    lock_passwd: false

packages:
  - docker.io
  - docker-compose
```

Figuur 2.4: Voorbeeld Cloud Config Data.

### 2.2.3 Omgevingen

De naam van het programma zegt het al duidelijk. Cloud-init is een service gemaakt voor cloud instanties en cloud platformen. Bij cloud platformen heb je de optie om user data toe te voegen aan de server(s) die je opstart. Meestal is dat een tekst vak waar je data kan invoeren. Dit kan een User-Data script zijn of een Cloud Config bestand. Voorbeelden van cloud platformen zijn: RackSpace, Amazon Web Services, Hetzner en IBM.

Het is mogelijk om er lokaal mee te werken. Op de server in kwestie moet de package cloud-init worden geïnstalleerd en dan via die package de user data uitvoeren. Je kan deze package installeren en uitvoeren op je host systeem, maar ook in een virtuele omgeving. Programma's om hiervoor te gebruiken zijn: Hyper-V, VMWare en VirtualBox.

### 2.2.4 Systemen

Volgens de site van cloud-init (Moser & Harlow, 2019) is het oorspronkelijk gemaakt voor Ubuntu systemen. Maar door het succes van de software is het nu beschikbaar op bijna elk Linux systeem: Fedora, Redhat, CentOS,..

## 3. Literatuurstudie

In dit hoofdstuk wordt er een literatuur studie gedaan. Er zijn 2 artikels gevonden die worden besproken: **An introduction to server provisioning with CloudInit** en **Using Ansible to Bootstrap My Work Environment Part 4**. De 2 artikels waren alle 2 ook een onderdeel van het bachelorproef voorstel.

### 3.1 An introduction to server provisioning with CloudInit

Het eerste artikel dat gevonden is heet: **An introduction to server provisioning with CloudInit**. Het is geschreven door Viktor Petersson. (Petersson, 2017) beschrijft in zijn artikel de basis van cloud-init en hoe een er mee kan worden gewerkt op CloudSigma.

In deze literatuurstudie wordt er besproken wat de link met Ansible is volgens (Petersson, 2017). Een ander groot onderdeel van het artikel is hoe het Cloud config bestand wordt opgemaakt. En wat dit bestand heeft/geeft qua voordelen. Ook is er een deel waar er wordt getoond hoe er een CloudSigma server wordt opgesteld .Dit wordt worden besproken. Dit is een klein deel van het artikel en niet van nut voor deze bachelorproef/onderzoek.

#### 3.1.1 Verschil en samenwerking met Ansible volgens Viktor Petersson

(Petersson, 2017) wilt allereerst de lezer doen inzien dat cloud-init een specifieke plaats heeft in de server provisioning wereld. In (Petersson, 2017) wordt er vermeld dat Cloud-init namelijk 1 eigenschap heeft, dat vele provisioning systemen zoals een Ansible, Puppet of Chef niet hebben.

Terwijl cloud-init perfect kan gebruikt worden als een stand-alone provisioning systeem. Is het 1 van de weinig systemen die het gebruik met andere provisioning systeem ondersteunt, en zelf aanraadt volgens (Petersson, 2017). (Petersson, 2017) vermeld ook dat hij prefereert cloud-init te gebruiken met Ansible.

### 3.1.2 Setup cloud config bestand

Het volgende onderdeel van (Petersson, 2017) dat we bespreken is het opstellen van zijn cloud config bestand. Dat is het bestand waar de server configuraties wordt ingevoerd. (Petersson, 2017) legt aan de hand van verschillende functies van het cloud config bestand uit wat de voordelen van cloud-init zijn. In Hoofdstuk 2.2 is hier wel al meer over uitgelegd. Sommige delen gaan minder uitgebreid zijn omdat het al in Hoofdstuk 2.2.1 te vinden is.

#### SSH

Het eerste onderdeel van het cloud config bestand van (Petersson, 2017) zijn de SSH keys. Cloud-init is volgens (Petersson, 2017) handig om ssh keys toe te voegen aan de server. Zo kan er verbinding gemaakt worden met de server.

(Petersson, 2017) gebruikt 2 fictieve gebruikers en SSH Keys in zijn voorbeeld. Deze publieke SSH sleutels worden geïnstalleerd op de server voor de gekozen gebruiker. In het fictieve voorbeeld dus een een publieke sleutel voor user1 en user2 op de server host. Op Ubuntu Cloud Images als de gebruikers niet aanwezig zijn zullen deze worden geïnstalleerd op de standaard gebruiker ubuntu.

(Petersson, 2017) maakt verbinding met de server via het commando *ssh ubuntu@IPADRESS*. Via de toegevoegde public keys gaat dit zonder problemen.

```
#cloud-config
ssh_authorized_keys:
- ssh-rsa AAA... user1@host
- ssh-rsa AAA... user2@host
```

#### Systeem updates

Het volgende voordeel van cloud-init volgens (Petersson, 2017) is het uitvoeren van systeem updates tijdens de eerste boot. Dit is 1 van de functies die al werden besproken in Hoofdstuk 2.2.1.

In zijn voorbeeld gebruikt (Petersson, 2017) *apt\_upgrade*. Terwijl er in Hoofdstuk 2.2.1 werd gezien dat updates worden gedaan met *package\_upgrade*. *apt\_upgrade* is gewoon een alias voor *package\_upgrade*.

Ook is *apt\_update* zijn standaard waarde true als er packages worden geïnstalleerd op de server.

```
#cloud-config
```

```
apt_update: true
apt_upgrade: true
```

### Installeren packages

Het volgende dat wordt besproken is het installeren van packages. Ook dit werd al besproken in Hoofdstuk 2.2.1. Extra informatie die nog niet gekend was geeft (Petersson, 2017) niet. Zijn voorbeeld staat wel ook hieronder

```
#cloud-config
packages:
- python-pip
- fail2ban
- vim
```

### Hostname

Wat (Petersson, 2017) ook handig vindt aan cloud-init is het aanpassen van de hostname. Deze kan ook makkelijk worden aangepast.

```
#cloud-config
hostname: mynode
fqdn: mynode.example.com
manage_etc_hosts: true
```

### Commando's

Als er sprake is van een meer geavanceerde gebruiker moeten er ook commando's worden uitgevoerd. Er zijn 2 opties voor commando's uit te voeren *runcmd* en *bootcmd*. Hierover kan er ook meer informatie worden gevonden in Hoofdstuk 2.2.1.

```
#cloud-config
runcmd:
- ls -l /root
```

### Server configuration manager

Een laatste voordeel volgens (Petersson, 2017) is dat als meer geavanceerde gebruiker er ook de mogelijkheid is om een extra Server Configuration Manager te gebruiken.

Cloud-init ondersteunt samenwerking met onder andere Chef, Puppet en Salt.

### Alles samenbinden

Ten laatste toon (Petersson, 2017) hoe dit allemaal kan worden samen gegooit in bestand. Dit allemaal wordt samengevoegd in een YAML bestand en bovenaan wordt er *#cloud-*

*config* gezet. Dit werd ook vermeld in Hoofdstuk 2.2

```
#cloud-config
ssh_authorized_keys:
- ssh-rsa AAA...user1@host
- ssh-rsa AAA...user2@host

hostname: mynode
fqdn: mynode.example.com
manage_etc_hosts: true

apt_update: true
apt_upgrade: true

packages:
- python-pip
- fail2ban
- vim

runcmd:
- ls -l /root
```

## 3.2 Using Ansible to Bootstrap My Work Environment Part 4

Het tweede artikel is: **Using Ansible to Bootstrap My Work Environment Part 4**. Het is een Blogpost geschreven Scott Harney. (Harney, 2016b) beschrijft in zijn blogpost hoe hij zijn werk omgeving opstart via Ansible met behulp van cloud-init.

Het is het beste artikel dat werd gevonden waar Ansible en cloud-init beide expliciet werden gebruikt. Er werd voor het eerste een omgeving weergegeven die opgezet werd door een samenwerking van beide. In (Harney, 2016b) wordt beschreven hoe Scott Harney een EC2 instance op zet op AWS. AWS is 1 van de cloud providers die cloud-init ondersteunt. Het artikel is opgedeeld in 4 onderdelen.

'*Initial provisioning*', in dit deel wordt de EC2 instance opgestart. Dit gebeurt met een eerste Ansible playbook dat ondere ook verwijst naar het cloud-init script dat nodig is. '*Cloud-init*', in het 2de gedeelte wordt het cloud-init script opgemaakt en uitgelegd wat alles doet en betekent. '*Ansible follow up playbook*', in dit gedeelte wordt de EC2 instance geconfigureerd met Ansible. '*Launch and configure*' hier wordt de omgeving opgestart en worden sommige variabelen nog ingevuld.

Het literatuur onderzoek gaat over de eerste 3 onderdelen gaan: initial provisioning, cloud-init en Ansible follow up playbook. Hierin worden Ansible en cloud-init besproken en geconfigureerd.

### 3.2.1 Ansible initial playbook

Er werd gekozen om de initiële provisioning en post-provisioning in 2 verschillende playbooks te doen. (Harney, 2016b) vond dit het handigste en vind het onnodig om deze tot 1 playbook te maken.

Dit playbook zorgt voor het aanmaken en opstarten van de instance. Dit wordt gedaan via de Ansible role *ec2*. Via deze role en dit playbook wordt ook het cloud config bestand meegegeven. (Harney, 2016b) gaf dit mee door middel van de optie *user\_data*.

### 3.2.2 Cloud-init

(Harney, 2016b) beschreef hier hoe hij de het cloud config bestand, waar hij in het eerste playbook naar verwees, heeft geconfigureerd. Een eerste opmerking van (Harney, 2016b) was dat cloud-init niet veel goede documentatie had. (Harney, 2016b) vond dit opmerkelijk omdat het een populaire tool is. Vervolgens beschreef (Harney, 2016b) hoe zijn cloud config bestand is opgedeeld.

#### Begin

Allereerst werd de normale gebruiker van (Harney, 2016b) aangemaakt met de nodige configuraties.

```
#cloud-config
users:
- name: {{ ansible_user }}
ssh-authorized-keys:
- ssh-rsa umm. nope
groups: [ 'admin', 'adm', 'dialout', 'sudo', 'lxd',
          'plugdev', 'netdev' ]
shell: /bin/bash
sudo: ["ALL=(ALL) NOPASSWD:ALL"]
```

#### Server en host update

In het 2de deel van het bestand nam (Harney, 2016b) zijn Hostname en fqdn en werden de packages upgedate. Ook update hij de instance zijn */etc/host* en update hij de tijdzone.

```
hostname: "{{ item.hostname }}"
fqdn: "{{ item.fqdn }}"
manage_etc_hosts: true
timezone: US/Central
package_update: true
package_upgrade: true
```

## Packages

Om het laatste van zijn script uit te voeren had (Harney, 2016b) 2 packages nodig, namelijk: *python* en *awscli*.

```
packages :
- awscli
- python
```

## DNS zone bestand

Het laatste deel was het belangrijkste deel van het cloud config bestand. (Harney, 2016b) moest zijn DNS zone bestand updaten. Zodat zijn EC2 instance correct werkte. Dit deed hij door met de optie *write\_files* een script aan te maken om deze te updaten. Hij prefereerde deze methode dan het the pushen via *systemd*.

```
write_files :
- content: |
#!/bin/sh
FQDN='hostname -f '
ZONE_ID="{{ zone_id }}"
TTL=300
SELF_META_URL="http://169.254.169.254/latest/meta-data"
PUBLIC_DNS=$(curl ${SELF_META_URL}/public-hostname 2>/dev/null)

cat << EOT > /tmp/aws_r53_batch.json
{
"Comment": "Assign AWS Public DNS as a CNAME of hostname",
"Changes": [
{
"Action": "UPSERT",
"ResourceRecordSet": {
"Name": "${FQDN}.",
"Type": "CNAME",
"TTL": ${TTL},
"ResourceRecords": [
{
"Value": "${PUBLIC_DNS}"
}]]]]}
EOT

aws route53 change-resource-record-sets --hosted-zone-id ${ZONE_ID}
--change-batch file:///tmp/aws_r53_batch.json
rm -f /tmp/aws_r53_batch.json
path: /var/lib/cloud/scripts/per-boot/set_route53_dns.sh
permissions: 0755e
```



### 3.2.3 Ansible follow up playbook

Nadat de instance is opgestart en de eerste configuraties zijn uitgevoerd. Is het tijd voor het tweede playbook. Het eerste deel van het playbook ziet er zo uit.

```

- hosts: tag_Name_candyapplegrey
  gather_facts: True
  roles:
    - role: ec2
    - role: common
    - role: ansible_mystuff
    - role: openvpn_server

  tasks:
    - name: Reboot system if required
      tags: reboot
      become: yes
      command: /sbin/reboot  removes=/var/run/reboot-required
      async: 1
      poll: 0
      ignore_errors: true

    - name: waiting for {{ inventory_hostname }} to reboot
      local_action: wait_for host={{ inventory_hostname }} state=started delay=10
      become: no

```

#### EC2 rol

Dit deel van het playbook voegde de nieuwe publieke sleutel van de opgestarte host toe aan het *known hosts* bestand. Maar Ansible zal ook vragen de sleutel te accepteren bij de eerste ssh. Als deze nog niet in het *known hosts* bestand zit. Dit is het enige dat de EC2 role deed. Het wordt ook alleen aangeroepen als het IP adress gedefinieerd is.

```

name: Add the instance to known hosts
local_action: command sh -c 'ssh-keyscan -t rsa {{ ec2_ip_address }} >>'
when: ec2_ip_address is defined

```

#### Common rol

Voor de common rol werden we verwezen naar een andere artikel namelijk: (Harney, 2016a). Deze rol was zeer uitgebreid, dus heeft het zijn eigen artikel.

#### Sectie 1 common rol

Git is zeer belangrijk in deze rol dus koos de auteur om dit als allereerst te installeren met de rol. De ssh sleutels worden ook gekopieerd van uit een private git repository. Erna wordt *with\_items* gebruikt. Het is een looping optie in Ansible die hier gekozen git

bestanden kopieerde. Het laatste deel zorgde ervoor dat de auteur zijn private bitbucket repo's beschikbaar zijn ,met zijn vorige sleutels, vanop de instance.

```

- name: install git
  become: yes
  apt:
    name: git
    state: present

- name: send ssh id
  copy:
    src: /home/sharney/.ssh/id_rsa
    dest: /home/sharney/.ssh/id_rsa
    mode: 0600

- name: git configfiles
  become: no
  copy: src={{ item.src }} dest={{ item.dest }}
  with_items:
    - { src: 'dot_gitignore', dest: '/home/sharney/.gitignore' }
    - { src: 'dot_gitignore', dest: '/home/sharney/.gitignore' }
    - { src: 'dot_git', dest: '/home/sharney/.git' }

- name: bitbucket key to known_hosts
  known_hosts:
    key: "{{ lookup('pipe', 'ssh-keyscan -t rsa bitbucket.org') }}"
    name: bitbucket.org
    state: present

```

## Sectie 2 common rol

Dit deel toon (Harney, 2016a) zijn script, dat hij maakte om zijn private configuraties te doen. Dit werd ergens online gevonden, meerbepaald dankzij: (Harney, 2014).

```

- name: do configfiles repo
  become: no
  script: /home/sharney/source/ansible-mystuff/configfiles_setup.sh
  creates=/home/sharney/configfiles/.configfiles_done

- name: ssh id permissions fix
  become: no
  file: path=/home/sharney/.ssh/id_rsa mode=0600

#!/bin/sh

# do the bits to setup configflies repo
if [ ! -d $HOME/configfiles ]; then
  mkdir $HOME/configfiles

```

```

cd $HOME
git clone --no-checkout git@bitbucket.org:scott_harney/configfiles.git
git reset --hard origin/master
fi

if [ ! -e $HOME/Dropbox/.tmux.conf ]; then
ln -s $HOME/Dropbox/.tmux.conf
fi
if [ ! -e $HOME/Dropbox/.dir_colors ]; then
ln -s $HOME/Dropbox/.dir_colors
fi

touch $HOME/configfiles/.configfiles_done

```

### Sectie 3 common rol

Het volgende deel van de rol voegde een ssh github sleutel toe. Voor het clonen van publieke repos later in het in de rol. Ook werden de andere nodige packages geïnstalleerd. Wel zijn sommige packages onnodig als er geen GUI aanwezig is op de instance.

```

- name: github key to known_hosts
known_hosts:
    key: "{{ lookup('pipe', 'ssh-keyscan -t rsa github.com') }}"
    name: github.com
    state: present

- name: install packages for emacs and more
action: apt pkg={{ item }} state=installed install_recommends=yes
become: yes
with_items:
    - emacs24
    - emacs24-el
    - pandoc
    - tmux
    - zsh
    - ispell
    - vpnc
    - fonts-hack-ttf
    - ruby
    - ruby-aws-sdk
    - python-pip
    - python-pip-whl
    - virtualenv
    - curl
    - openjdk-8-jre
    - fonts-crosextra-caladea

```

```
- fonts-crosextra-carlito # substituijon fonts for ms office. probab
```

#### Sectie 4 common rol

Dit waren gewoon python pip en ruby onderdelen die hielpen bij het beheren van AWS.

```
- name: python pip install items
become: yes
pip: name={{ item }} state=present
with_items:
  - powerline-status
  - awscli
  - saws
```

```
- name: aws sdk v1 for ruby for awscli
become: yes
gem: name=aws-sdk-v1 state=present
```

#### Sectie 5 common rol

Dit deel zorgde voor de installatie van *spacemacs*. *spacemacs* is een linux tekst editor met de power en uitbreidbaarheid van *Emacs* en de werking van *vi/vim*.

```
name: check for spacemacs already imported
stat: path=/home/sharney/.emacs.d/spacemacs.mk
register: spacemacs_import
```

```
- name: mv .emacs.d out of the way for spacemacs if spacemacs not yet d
command: mv /home/sharney/.emacs.d /tmp
when: spacemacs_import.stat.exists == False
```

```
- name: spacemacs
git: repo=https://github.com/syl20bnr/spacemacs dest=/home/sharney/.ema
when: spacemacs_import.stat.exists == False
```

```
- name: for spacemacs org-protocol-capture-html
git: repo=https://github.com/alphapapa/org-protocol-capture-html.git de
```

```
- name: fix .emacs.d/private via source copy
copy: src=/home/sharney/.emacs.d/private/private dest=/home/sharney/.em
when: spacemacs_import.stat.exists == False
# note: relying on the fact that the host running ansible has checked o
```

#### Sectie 6 common rol

Hier zette (Harney, 2016a) zijn tijdzone. Ook worden hier bestanden gesynchroniseerd die in zijn Dropbox stonden.

```
- name: gen en_US.UTF-8 locale
```

```

become: yes
locale_gen: name=en_US.UTF-8 state=present

- name: set default locale to en_US.UTF-8
become: yes
command: /usr/sbin/update-locale LANG=en_US.UTF-8

- name: symlink for dir_colors
file: src=/home/sharney/Dropbox/.dir_colors path=/home/sharney/.dir_col

- name: symlink for .tmux.conf
file: src=/home/sharney/Dropbox/.tmux.conf path=/home/sharney/.tmux.con

```

### Sectie 7 common rol

Het laatste deel configureerde de vpn. Ook worden de files van zijn website gepulled. Ten laatste is de customisatie van *zsh*.

```

- name: copy vpnc config
become: yes
copy:
  src: vpnc
  dest: /etc
  owner: root
  group: yes
  mode: 0700

- name: git clone scottharney.com
git: repo=git@bitbucket.org:scott_harney/scottharney.com.git dest=/home

- name: install ohmyzsh via git
git: repo=https://github.com/robbyrussell/oh-my-zsh.git dest=/home/sha

- name: fix .zshrc
copy: src=/home/sharney/.zshrc dest=/home/sharney/.zshrc

```

### Ansible bestanden rol

Deze rol pulled alle playbooks van (Harney, 2016b) zijn git repo *ansible\_mystuff* en zet het op de server. Zo kan hij vanop deze instance aan zijn playbooks werken.

```

- name: install ansible
become: yes
apt:
  name: ansible
  state: present

```

```

- name: ansible-mystuff repo for deployment of util hosts
  git: repo=git@bitbucket.org:scott_harney/ansible-mystuff.git
  dest=~/.source/ansible-mystuff

```

## OpenVPN rol

Via deze kan (Harney, 2016b) zijn instance gebruiken op plaatsen die geen veilige verbinding zijn (bijvoorbeeld een café). Ook hij aan bestanden geraken van EC2 instances die geen internet gateway hebben. Deze instance wordt door deze rol een ssh host en een gateway naar (Harney, 2016b) zijn VPC. Omdat het een aparte rol is kunnen deze functie later hergebruikt worden.

### Sectie 1 OpenVPN rol

Dit installeerde de openvpn package en kopieerde de al aanwezige configuraties naar de instance.

```

- name: install openvpn
  become: yes
  apt:
    name: openvpn
    state: present

- name: copy openvpn configuration data
  become: yes
  copy: src=openvpn dest=/etc

```

### Sectie 2 OpenVPN rol

Hier wordt ip forwarding aangezet en een *iptables* is nu aanwezig voor het VPN verkeer.

```

- name: set up ip forwarding
  become: yes
  sysctl: name="net.ipv4.ip_forward" value=1 sysctl_set=yes state=present

- name: update /etc/rc.local
  become: yes
  copy: src=rc.local dest=/etc/rc.local mode=0755

- name: do iptables
  become: yes
  iptables: state=present table=nat chain=POSTROUTING source=192.168.3.0/24

```

### Sectie 3 OpenVPN rol

Dit was volgens (Harney, 2016b) het moeilijkste deel van zijn playbook. Voor dit deel heeft een lange tijd moeten troubleshooten. Het probleem dat hij ondervond was dat er een bug

was die `openvpn` reports niet goed bij hield in Xenial. Zo heeft (Harney, 2016b) zijn bug kunnen oplossen.

```
– name: fix debian bug in openvpn service https://bugs.launchpad.net/ub
become: yes
copy: src=openvpn@.service dest=/lib/systemd/system/
```

## Sectie 4 OpenVPN rol

Er is ee een `systemd` module in Ansible 2.2 maar (Harney, 2016b) koos er voor om dit toch via de command line te doen. Dit is iets dat hij later zou kunnen aanpassen naar de module van Ansible. Maar momenteel werkt dit goed genoeg

```
– name: restart systemd daemon after updating unit config
become: yes
command: systemctl daemon-reload
```

```
– name: enable openvpn services
become: yes
command: systemctl enable openvpn.service
```

```
– name: restart openvpn services
become: yes
command: systemctl restart openvpn.service/
```





## 4. Methodologie

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Maecenas non massa. Vestibulum pharetra nulla at lorem. Duis quis quam id lacus dapibus interdum. Nulla lorem. Donec ut ante quis dolor bibendum condimentum. Etiam egestas

tortor vitae lacus. Praesent cursus. Mauris bibendum pede at elit. Morbi et felis a lectus interdum facilisis. Sed suscipit gravida turpis. Nulla at lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent nonummy luctus nibh. Proin turpis nunc, congue eu, egestas ut, fringilla at, tellus. In hac habitasse platea dictumst.

Vivamus eu tellus sed tellus consequat suscipit. Nam orci orci, malesuada id, gravida nec, ultricies vitae, erat. Donec risus turpis, luctus sit amet, interdum quis, porta sed, ipsum. Suspendisse condimentum, tortor at egestas posuere, neque metus tempor orci, et tincidunt urna nunc a purus. Sed facilisis blandit tellus. Nunc risus sem, suscipit nec, eleifend quis, cursus quis, libero. Curabitur et dolor. Sed vitae sem. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas ante. Duis ullamcorper enim. Donec tristique enim eu leo. Nullam molestie elit eu dolor. Nullam bibendum, turpis vitae tristique gravida, quam sapien tempor lectus, quis pretium tellus purus ac quam. Nulla facilisi.

## **5. Opzetten lokale testomgeving**



## **6. Opzetten Hetzner Cloud testomgeving**



## 7. Basisconfiguraties op de servers

MSS OOK SCHIJF CONFIGURATIE + MSS INLEIDING WAT AANPASSEN BIJ  
HOOFSTUKKEN





## **8. Server installatie en configuratie**



## **9. Instellingen aanpassen na het opstarten**



## 10. Container & Cluster Configuratie



## 11. 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 consetetur 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.



# A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

## A.1 Introductie

Voor de installatie van servers wordt al jaren Ansible gebruikt. Ansible is een universele 'taal' die taken voor servers automatiseert door middel van hun playbook. Zo een Ansible playbook is een georganiseerde unie van scripts dat het werk voor de server configuratie definieert.

Cloud-Init is momenteel een van de industrie standaarden voor het opbouwen van cloud servers, het maakt gebruik van cloud images. Dat zijn besturingssysteem sjablonen en elke instantie begint als een identieke kloon van elke andere instantie. De gebruikersgegevens geven elke cloud instantie haar persoonlijkheid. Doormiddel van cloud-init worden deze gegevens op de instantie toegepast.

Dit zijn allebei provisioning systemen. Op een verschillende manier doen ze in theorie hetzelfde. Ze brengen de server allebei naar de gewenste toestand van de gebruiker. Doordat deze op verschillende manieren werken, wordt er in de praktijk in combinatie met beide gewerkt. In dat geval gebruik je eerste cloud-init om de server naar een gewenste toestand te brengen waar na Ansible het kan overnemen.

Maar is dit de perfecte samenwerking? Het bedrijf Be-mobile is opzoek naar het antwoord. In dit onderzoek bestuderen we waar deze mekaar aanvullen en hoe ze dan op een perfect performante manier werken. Natuurlijk is het goed mogelijk dat deze mekaar niet aanvullen

en dan is het onderzoek waar en wanneer je voor wat moet kiezen en waarom dit mekaar overbodig maakt.

In dit onderzoek zullen we dit trachten te ontdekken. De vragen waar het, de antwoorden op wil vinden zijn:

- Vullen Ansible en Cloud-Init mekaar aan, of maken ze elkaar overbodig?
- Ook hoe ze mekaar aanvullen en/of hoe ze mekaar overbodig maken?

## A.2 Stand Van Zaken

Ansible en Cloud-Init zijn geen onbekende voor mekaar dus er is wel een basis gevonden waarop er verder kan gewerkt worden. Maar zijn niet zo bekend dat de ultieme samenwerking al gevonden is. In het onderzoeken van de 2 systemen zijn er 5 artikels gevonden die een samenwerking beschrijven of afraden.

In de artikels **"An introduction to server provisioning with CloudInit"**(Petersson, g.d.), **"Using Ansible to Bootstrap My Work Environment Part 4"** (Harney, 2016b) en **"Customizing Cloud Assembly Deployments with Cloud-Init"** (Arkland, 2018) is er een vaak voorkomend fenomeen gevonden. Meestal gebruiken de gebruikers/auteurs eerst cloud-init om de machine op te starten, daarna ansible voor de verdere specifieke eigenschappen.

Het artikel geschreven door Petersson (g.d.) is wel interessant doordat hij ook andere systemen buiten cloud-init heeft getest, waaronder puppet en chef. Maar toch verkiest hij om Ansible te gebruiken. Dit betekent dat een samenwerking met ansible optimaler is.

De artikels **"Zero Touch Provisioning of Infoblox Grid on OpenStack using Ansible"**(Aditya, 2018) en **"Automated Ansible AWX Installation"**(„Automated Ansible AWX Installation", g.d.) beschrijven dan weer hoe beide elke als een apart systeem kunnen worden gebruikt zonder de hulp van de ander. Maar toch zijn er een paar gelijkenissen zo gebruikt „Automated Ansible AWX Installation" (g.d.) in "Automated Ansible AWX Installation" notities van een Ansible installatie om het met cloud-init te installeren.

Er is duidelijk al bekend dat er samengewerkt kan worden, maar ook soms weer niet. Maar criteria om te bepalen wanneer welke het best is, is nog niet bekend.

## A.3 Methodologie

Dit onderzoek zal gevoerd worden door virtuele server omgevingen op te zetten met Cloud-init en/of Ansible. De programma's die hierbij zullen worden gebruikt zijn VirtualBox, Vagrant en Atom.

Virtualbox is een programma om virtuele machines op te draaien. Met Vagrant kan je van

in je shell met een simpel commando virtuele machines op starten. Atom is dan weer een teksteditor om de configuratie goed in neer te pennen. Normaal is deze het vermelden niet waar maar door de overzichtelijke manier van werken die het aanbiedt is deze toch veel beter dan een gewone Notepad.

Met Ansible zal er voor verschillende servers een testomgeving worden opgezet. Dit zal dan ook gebeuren met cloud-init. Deze resultaten en omgevingen kunnen dan worden vergeleken met elkaar. De technische eigenschappen/resultaten worden dan onder de loep genomen: de performantie, de snelheid van het opstarten,... Ook zal er worden gekeken naar de config files van beiden en kan er worden gekeken welke daar per omgeving “beter” is. M.a.w. welke er op een kortere overzichtelijkere manier hetgeen kan bekomen. Be-Mobile kan ook nog verdere criteria bepalen waardoor we een keuze gaan maken. Daarna worden ook combinaties van de twee aangemaakt en dan kunnen we deze vergelijken met de originele servers.

Ook kan het zijn dat er vragen worden gesteld aan medewerkers van het bedrijf Be-Mobile om te bekijken wat hun mening over beide is.

## A.4 Verwachte resultaten

De verwachtingen zijn dat een combinatie van cloud-init en Ansible meestal het beste zal zijn. Dat er voor elke server tot een bepaald moment cloud-init of ansible zal worden gebruikt waarna ansible of cloud-init het zal overnemen. Ook zijn de verwachtingen dat er voor elke omgeving een andere oplossing zal zijn. Er zal veel afhangen van het type besturingssysteem dat draait en van de servers om te kunnen bepalen welke het best wordt gekozen.

De verwachtingen zijn ook dat niet bij alles een combinatie zal worden gebruikt. Ook kan het misschien zijn dat het beter is dat je één van beide kiest voor een bepaalde server omgeving.

## A.5 Verwachte conclusies

De verwachte conclusie is dat elke server omgeving een andere uitkomst zal hebben. Er veel zal afhangen van welke server je precies wilt draaien op welk systeem. Dan kan er worden gekozen voor een bepaalde combinatie van beide of misschien één van beide. De conclusie zal niet zijn welke van de twee beter is. Maar eerder hoe ze het best gebruikt worden per serveromgeving.



## Bibliografie

- Aditya. (2018). Zero Touch Provisioning of Infoblox Grid on OpenStack using Ansible. *Info Blox*. Verkregen van <https://community.infoblox.com/t5/Community-Blog/Zero-Touch-Provisioning-of-Infoblox-Grid-on-OpenStack-using/ba-p/14910>
- Arkland, C. D. (2018). Customizing Cloud Assembly Deployments with Cloud-Init. *Blog VMware*. Verkregen van <https://blogs.vmware.com/management/2018/11/customizing-cloud-assembly-deployments-with-cloud-init.html>
- Automated Ansible AWX Installation. (g.d.). *devendor tech*. Verkregen van [https://www.devendortech.com/articles/AWX\\_Tower.html](https://www.devendortech.com/articles/AWX_Tower.html)
- cloud-init.io. (2019). *cloud-init-docs-modules*. cloud-init. Verkregen van <https://cloudinit.readthedocs.io/en/latest/topics/modules.html>
- Harney, S. (2014). How To Use Git to Manage your User Configuration Files on a Linux VPS. Verkregen van <https://www.digitalocean.com/community/tutorials/how-to-use-git-to-manage-your-user-configuration-files-on-a-linux-vps>
- Harney, S. (2016a). Using Ansible to Bootstrap My Work Environment Part 3. Verkregen van <https://www.scottharney.com/using-ansible-to-bootstrap-my-work-environment-part-3/>
- Harney, S. (2016b). Using Ansible to Bootstrap My Work Environment Part 4. Verkregen van <https://www.scottharney.com/using-ansible-to-bootstrap-my-work-environment-part-4/>
- Moser, S. & Harlow, J. (2019). cloud-init.io. Verkregen van <https://cloud-init.io/>
- Petersson, V. (g.d.). An introduction to server provisioning with CloudInit. *Cloudsigma*. Verkregen van <https://www.cloudsigma.com/an-introduction-to-server-provisioning-with-cloudinit/>
- Petersson, V. (2017). An introduction to server provisioning with CloudInit. Verkregen van <https://www.cloudsigma.com/an-introduction-to-server-provisioning-with-cloudinit/>

- RedHat. (2017). *ANSIBLE IN DEPTH*. Red Hat Inc. Verkregen van <https://www.ansible.com/hubfs/pdfs/Ansible-InDepth-WhitePaper.pdf>
- Skinner, M. & Heldebrant, M. (2017). *Cloud-Init*. RedHat. Verkregen van <https://people.redhat.com/mskinner/rhug/q3.2014/cloud-init.pdf>