

# Onderzoek naar samenwerking tussen Cloud-Init en Ansible

## Onderzoeksvoorstel Bachelorproef

Maarten De Smedt<sup>1</sup>

### Samenvatting

Bij bedrijven wordt voor de installatie van servers al jaren provisioning systemen gebruikt zoals Ansible en Cloud-init, te samen of apart. Ansible maakt gebruik van hun playbooks om de server juiste te configureren. Cloud-Init doet dat doormiddel van hun cloud-images. Maar waar moet je overstappen naar de ander, hoe gebruik je ze het best samen en maken ze mekaar misschien overbodig? Allemaal vragen die nog onbeantwoord zijn. Aangezien veel bedrijven en gewone gebruikers gebruiken maken van beide is het belangrijk dat daar een duidelijk antwoord op word gevonden. In dit document kan u verder lezen over: wat deze 2 technologieën precies en waar ze verschillen en gelijk zijn, 5 verschillende literatuur studies/onderzoeken over dit onderwerp, hoe ik dit wil onderzoeken, de verwachte resultaten en conclusies. Ik verwacht niet dat onderzoek één resultaat zal hebben maar eerder verschillende resultaten voor verschillende type omgevingen. De verwachte conclusie zal dus eerder zijn waar je welke het beste gebruikt en waar je best gaat combineren. In de toekomst zal er door dit onderzoek hopelijk meer duidelijkheid zijn tussen de samenwerking(of misschien is die er niet) van Cloud-init en Ansible.

### Sleutelwoorden

Systeembeheer — Cloud-Init — Ansible

### Co-promotor

Simon Lepla<sup>2</sup> (Be-Mobile)

**Contact:** <sup>1</sup> maarten.desmedt.v5415@student.hogent.be; <sup>2</sup> simon.lepla@be-mobile.com;

## Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>1</b>
<b>2</b>	<b>Literatuurstudie</b>	<b>2</b>
2.1	Introduction to server provisioning with CloudInit	2
2.2	Using Ansible to Bootstrap My Work Environment Part 4	2
2.3	Customizing Cloud Assembly Deployments with Cloud-Init	2
2.4	Zero Touch Provisioning of Infoblox Grid on OpenStack using Ansible	2
2.5	Automated Ansible AWX Installation	2
<b>3</b>	<b>Methodologie</b>	<b>2</b>
<b>4</b>	<b>Verwachte resultaten</b>	<b>3</b>
<b>5</b>	<b>Verwachte conclusies</b>	<b>3</b>
	<b>Referenties</b>	<b>3</b>

## 1. Introductie

Voor de installatie van servers wordt al jaren Ansible gebruikt. Ansible is een universele 'taal' die taken voor servers au-

tomatiseert doormiddel 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 op zoek 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?

## 2. Literatuurstudie

### 2.1 Introduction to server provisioning with CloudInit

Het allereerste artikel dat gevonden werd is An introduction to server provisioning with CloudInit. Dit artikel werd gevonden op de site CloudSigma en is geschreven door Petersson (g.d.). Dit artikel gaat over hoe je een cloud-init server opzet. Hierin in beschrijft hij hoe je een cloud-init server opzet met Ubuntu.

In dit artikel beschrijft Petersson (g.d.) cloud-init als een provisioning systeem, maar niet zomaar een provisioning systeem. Petersson (g.d.) raadt aan om cloud-init te gebruiken in samenwerking met een ander provisioning systeem. Hierin geeft hij tal van voorbeelden zoals Chef, puppet, Salt. Maar ook Ansible de auteur zelf prefereert zelf om cloud-init samen met ansible te gebruiken.

Verder staat er in "An introduction to server provisioning with CloudInit" hoe je SSH sleutels, systeem updates en packages installeert op de server. Ook hoe je de host-name moet veranderen en commands moet runnen op de eerste boot, allemaal vrij standaard.

Over de samenwerking met Ansible staat er dat dit aan te raden is voor de meer "verfijnde" gebruikers. Dit is een zeer complex onderwerp door de vele variabelen die er zijn.

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

Het tweede artikel dat werd gevonden is "Using Ansible to Bootstrap My Work Environment Part 4". Het is een geschreven door Harney (2016). In dit experiment gebruikt de auteur eerste cloud-init voor de eerste basis provisioning en daarna ansible om de meer geavanceerde taken uit te voeren. Hij wilt EC2 installeren op zijn server.

In het artikel gebruikt Harney (2016) cloud-init vooral om de vroege provision onderdelen uit te voeren. Met cloud-init doet hij de eerste basis dingen die er moeten gebeuren op de server. Hij maakt zijn user aan en installeert de nodige packages en installeert ook python2. In vroeger experimenten pushte hij deze naar systemd. Maar werken via cloud-init vindt de auteur handiger.

Een laatste opmerking die hij over cloud-init heeft is dat er iets te weinig documentatie over te vinden was in zijn opinie. Hierna gebruikte hij Ansible om EC2 te installeren. Dit deed hij door een role met EC2 toe te voegen. Hierna voegde hij de juiste parameters toe.

In dit artikel zien we een eerste keer hoe ansible en cloud-init kunnen samenwerken.

### 2.3 Customizing Cloud Assembly Deployments with Cloud-Init

Het derde artikel dat gekozen is "Customizing Cloud Assembly Deployments with Cloud-Init". Het artikel is geschreven door Arkland (2018).

In het artikel zeggen ze dat automatisering van servers (virtuele machines) iets fantastisch is, maar dat er nog verbetering mogelijk is. En daar komt cloud-init in het spel. Door dit te gebruiken kan je op een snelle efficiënte manier vm's op zetten volgens Arkland (2018).

Ansible komt in dit artikel eerder te pas als een extra provisioning systeem dat je kunt gebruiken voor verdere aanpassingen.

Verder in het artikel zie je hoe Arkland (2018) een cloud-init server opzet en hoe je de best beheert.

### 2.4 Zero Touch Provisioning of Infoblox Grid on OpenStack using Ansible

Het vierde artikel dat gevonden werd is "Zero Touch Provisioning of Infoblox Grid on OpenStack using Ansible". Het is een geschreven door Aditya (2018).

In tegenstelling ook tot vorige artikels toont dit aan hoe ansible cloud-init overbodig maakt. In het artikel spreekt Aditya (2018) over hoe een ansible taak namelijk "Deploying infoblox grid" ervoor zorgt dat bepaalde dingen niet meer nodig zijn. Waaronder heet license and network initialization dat gedaan werd via cloud-init. Door ansible te gebruiken is dit nu overbodig.

### 2.5 Automated Ansible AWX Installation

Het vijfde en laatste artikel dat er werd gevonden is „Automated Ansible AWX Installation" (g.d.). Het beschrijft een automatische installatie van AWX op een ubuntu machine via cloud-init. Het volgt een installatie van AWX met ansible maar de auteur past deze toe met cloud-init.

Voor die rede is dit artikel zeer interessant. Er zijn al artikels gepasseerd die één beschrijven, die een samenwerking beschrijven of een overbodigheid van één beschrijven. Maar dit beschrijft hoe je met installatie notities van ansible cloud-init kunt gebruiken. Dit toont toch wel wat gelijkenissen aan in configuratie.

## 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. Met beter wordt bedoeld dat de één misschien overzichtelijker is of dat er minder code moet worden geschreven om iets aan te passen/toe te voegen. Zo kunnen er dan combinaties van de twee worden 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.

#### 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.

#### 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.

#### Referenties

- 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>

Harney, S. (2016). Using Ansible to Bootstrap My Work Environment Part 4. *Blog*. Verkregen van <https://www.scottharney.com/using-ansible-to-bootstrap-my-work-environment-part-4/>

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/>