

Een vergelijkende studie van moderne configuration management tools

Onderzoeksvoorstel Bachelorproef

Jens Uyttersprot

Samenvatting

De nood aan het verwerken van steeds grotere hoeveelheden data heeft geleid tot de ontwikkeling van configuration management tools. Deze automation services vereenvoudigen het beheer van server clusters en zorgen voor een continuous delivery process. Moderne netwerkinfrastructuren zijn hierdoor steeds beschikbaar, schaalbaar en transparant.

Er bestaat reeds een groot aanbod aan verschillende configuration management tools, die hebben vaak een verschillend doeleinde. In mijn onderzoek wil ik de verschillende tools bestuderen, toepassen en testen. Hiervoor selecteer ik de grootste spelers op vlak van configuration management; zijnde Puppet, Ansible en Chef. Daarna bestudeer ik hun verschillen en weeg ik hun voor- en nadelen tegen elkaar af op vlak van performantie, gebruiksvriendelijkheid, schaalbaarheid, redundantie enzovoorts. Uiteindelijk trek ik hieruit mijn conclusies en beschrijf ik de verschillende tools naargelang hun nut en waarde.

Uit mijn onderzoek verwacht ik dat Puppet als performantste en schaalbaarste toolset uit de bus zal komen, mede omdat die al het langst op de markt en dus al het sterkst ontwikkeld is. Ansible daarentegen zal volgens mij het meest gebruiksvriendelijk zijn, maar zal minder toepasbaar zijn op een groot aantal servers tegelijkertijd. Tevens zal Chef een mooie alternatief zijn die zich ergens tussenin bevindt, althans volgens mij.

Technologie staat niet stil en ontwikkelaars zullen steeds nieuwere en betere configuration management tools blijven ontwikkelen. Als u bekijkt waar we nu staan tegenover twintig jaar geleden op vlak van automatisering, gebruiksgemak en de omvang van datacenters ziet de toekomst er zeer rooskleurig uit voor dit domein.

Sleutelwoorden

Systeem- en netwerkbeheer. Configuration Management — DevOps — Continuous Integration

Contact: jens.uyttersprot.v4158@student.hogent.be;

Inhoudsopgave

1	Introductie	1
2	State-of-the-art	2
2.1	Literatuurstudie	2
2.2	Online seminars	2
2.3	Hands-on approach	2
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2
	Referenties	3

1. Introductie

Tijdens mijn opleiding heb ik kennis gemaakt met automatisatie technieken voor servers. Bij 'Enterprise Linux' hebben we Ansible playbooks leren toepassen en tijdens de lessen 'Windows Server' heb ik me verdiept in PowerShell. Hoewel deze lessen interessant waren, had ik het gevoel dat ik slechts een klein deeltje van het geheel had ontdekt. Na wat

opzoekingswerk heb ik de wondere wereld van configuration management ontdekt. Ik was onmiddellijk gefascineerd over de mogelijkheid om clusters van servers makkelijk te beheren. In korte tijd ben ik meer te weten gekomen over onderwerpen zoals 'DevOps', 'Configuration Management' en 'Continuous Delivery'. Momenteel bestaat er een groot scala aan automation tools, toch is het niet steeds duidelijk welke tool het meest geschikt is in welke omgeving. In mijn studie wil ik de verschillen onderzoeken, de voor- en nadelen van deze tools tegen elkaar afwegen en hun performantie testen. Hierbij wil ik ook te weten komen welke tool het meest gebruiksgemak biedt, welke service het best toepasbaar is op een groot aantal servers en welke het best presteert op vlak van redundantie.

Concreet luiden mijn onderzoeksvragen:

- Wat zijn de verschillen tussen Puppet, Ansible en Chef?
- Welke tool scoort het best op vlak van performantie?
- Welke van deze tools biedt het meest gebruiksgemak?
- In welke mate houden deze tools rekening met schaalbaarheid?
- Hoe kan redundantie verzekerd worden bij de verschillende tools?

2. State-of-the-art

2.1 Literatuurstudie

Ten eerste ben ik begonnen met de documentatie van de te onderzoeken tools (Puppet, Ansible en Chef) te doorlopen. Dit nam veel tijd in beslag, maar het is de beste basis om mee te starten. Vervolgens ben ik online artikels gaan lezen die de toepassingen en de voor- en nadelen van elke tool omschrijven. Nadat ik een mooie basis van kennis had verworven over deze tools ben ik uiteindelijk relevante onderzoeken gaan bekijken. Zo kwam ik uit op de bachelorproef van Thomas Detemmerman (Detemmerman, 2017), maar in zijn werk onderzoekt hij enkel de voor- en nadelen voor een omschakeling van Puppet naar Ansible specifiek voor de infrastructuur bij de VRT. Daarin heb ik al kort kennis gemaakt met de verschillen tussen Puppet en Ansible. Vooral het boek *DevOps for Networking* (Armstrong, 2016) heeft mij veel inzicht in het praktisch verloop van het continuous delivery process bijgebracht. Dit boek beschrijft het algemene nut van DevOps en hoe Ansible helpt bij de automatisatie van het manuele, repetitieve gedeelte van de job.

2.2 Online seminars

Omdat ik een visueel geheugen heb, heb ik ook veel kennis verworven aan de hand van YouTube videos over 'Configuration Management', 'DevOps' en 'Infrastructure Automation'. Tevens heb ik de interactieve online seminars 'What is Puppet?' (Edureka!, 2017) en 'Puppet Tutorial for Beginners' (Edureka!, 2016) bijgewoond van 'Edureka!'. Omdat ik het gevoel had minder aandacht te hebben geschonken aan Chef ben ik op zoek gegaan naar extra informatie omtrent deze tool. Op de officiële website van Chef ben ik terecht gekomen op een zeer interessante sectie genaamd webinars. Deze educatieve filmpjes laten je op een speelse maar toch leerrijke manier kennis maken met de diverse mogelijkheden van Chef. Omdat deze videos zeer lang duurden heb ik er twee bekeken namelijk 'Chef Automate: Scaling Up your Automation' (Elizondo & Inamov, 2016) en 'Understanding the Chef Server' (Popovetsky & Delano, 2016).

2.3 Hands-on approach

Wanneer ik het gevoel had al een grondige kennis te hebben verworven over de verschillende tools wou ik wat voeling creëren door een tool daadwerkelijk te gebruiken. Hiervoor heb ik, geleerd uit de seminars, in virtuele machines een master-slave verbinding opgesteld door middel van het authenticatie proces en vervolgens een simpel manifest voor Puppet in de master toegepast om automatisch mariadb op alle slaves te installeren. Bovendien heb ik ook nog eens teruggeblikt op de Ansible playbooks die ik geconfigureerd heb in de lessen 'Enterprise Linux'. Een grondige uitwerking van verschillende experimenten op alle verschillende tools gebeurt in de bachelorproef zelf.

3. Methodologie

Na het uitdiepen van mijn kennis over deze tools door middel van een zeer uitgebreide literatuurstudie zal ik de handen uit de mouwen steken. Concreet ga ik een aantal experimenten uitvoeren. De Linux testomgeving zal bestaan uit verschillende virtuele machines waarop CentOS draait. Tijdens het eerste experiment zal ik automatisch een LAMP-stack laten opzetten door elke tool op een beperkt aantal virtuele machines. Bij tweede experiment voer ik hetzelfde uit maar op een groot aantal virtuele machines om de schaalbaarheid van de tools op de proef te stellen. Als derde en laatste experiment wil ik de redundantie mogelijkheden testen van de tools door een back-up master in te stellen en te zien hoe snel deze wordt geactiveerd na de uitval van de primaire master. Uiteindelijk zal ik meten welke tool het meest performant is, welke het gemakkelijkst te configureren is en hoe ze evolueren op een groter aantal agents. Tevens zal ik de individuele verschillen en hun prestaties documenteren en catalogeren op basis van beste gebruiksomgeving. Naast de experimenten zal ik ook een vragenlijst opstellen en naar diverse IT-bedrijven van verschillende groottes sturen om te weten te komen welke tools zij gebruiken in hun omgeving en de reden daarachter.

4. Verwachte resultaten

Ik verwacht uit de experimenten dat Puppet de meest performante en meest schaalbare tool zal zijn die het best kan omgaan met divergerende netwerkopstellingen. Bij zeer grote netwerken zal deze tool het meest efficiënt en dus het snelst te werk gaan. Daarentegen verwacht ik dat Ansible sneller zal werken bij kleinere opstellingen door zijn agentless eigenschap. Dankzij deze eigenschap en het feit dat de playbooks zijn geschreven in de zeer simpele YAML taal, zal Ansible als meest gebruiksvriendelijke tool beschouwd worden. Chef zal volgens mij het best kunnen omgaan met redundantie omdat zijn structuur het toelaat om meerdere niveaus van masters toe te wijzen. Door deze gelaagde regeling zal er een betere load balancing zijn van de servers. Deze tool is volgens mij de grootste 'concurrent' van Puppet. Chef is vermoed ik ook tamelijk performant, schaalbaar en redelijk gebruiksvriendelijk door zijn makkelijk te begrijpen fout uitvoer. De reden dat deze tool een geduchte, opkomende tegenstander is komt door het feit dat Facebook hiervan gebruik maakt en dit gigantisch platform meewerkt aan de dagelijkse verbetering van Chef.

5. Verwachte conclusies

Ik ben er bijna zeker van dat Puppet als beste en meest gebruikte tool uit de bus zal komen. De reden daarvoor is dat Puppet al het langst op de markt is en al veel tijd gekregen heeft voor ontwikkeling en om fouttolerantie te optimaliseren. Volgens mij zal Chef hier kort op volgen omdat deze tool zeer snel verbeterd wordt met behulp van professionele en marktleidende platformen. Op vlak van kleinere server clusters zal Ansible het best presteren. Door zijn opstelling

zonder agents en zijn makkelijke taal is de instapdrempel tot deze tool het kleinst. Ik ben zeer nieuwsgierig en hoop dat ik deze configuration management tools mag uitwerken en tegen elkaar afwegen in mijn bachelorproef.

Referenties

- Armstrong, S. (2016). *DevOps for Networking*. Packt.
- Detemmerman, T. (2017). *Technische voor-en nadelen van Puppet en Ansible. Verloop en redenen van een omschakeling*. Gent : s.n.
- Edureka! (2016). Puppet Tutorial for Beginners Part -1 — Puppet DevOps Tutorial — DevOps Tools — Edureka. Verkregen van <https://www.youtube.com/watch?v=0yVJhb2VkVk>
- Edureka! (2017). What is Puppet — Puppet Tutorial for Beginners — Puppet Configuration Management Tutorial — Edureka. Verkregen van https://www.youtube.com/watch?v=PL_J5Gj3GAQ
- Elizondo, A. & Inamov, B. (2016). Chef Automate: Scaling up your automation. Verkregen van <https://www.youtube.com/watch?v=ldY7KEOxCkM>
- Popovetsky, I. & Delano, S. (2016). Understanding the Chef Server. Verkregen van <https://www.youtube.com/watch?v=22GtVMHJDsI>