UPPSALA
UNIVERSITET

TVE 14 013 maj
Examensarbete 15 hp
Juni 2014

A Study of Configuration Management Systems

Solutions for Deployment and Configuration of Software in a Cloud Environment

Kim Torberntsson
Ylva Rydin

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

A Study of Configuration Management Systems

Kim Torberntsson & Ylva Rydin

The amount of servers needed in the world is constantly increasing along with the end users constantly increasing demand for large scale network solutions and IT- services. With virtualisation and cloud computing it has become much easier to create and deploy large numbers of virtual servers, but the servers still need configuration. This is precisely what configuration management systems (CMS) can help with. Cloud providers often want to include built in support for these systems to simplify for their customers. The purpose of the study was to compare the different CMS available and to recommend the ones that deserve to be supported by a cloud provider. A general study, based on parameters defined in the report, was made on twenty different CMS. The five top solutions were Puppet, Chef, CFEngine, Salt and Ansible, where Salt and Ansible are more lightweight solutions and Puppet, Chef and CFEngine are more advanced. They were included in a comparison study based on opinions and reviews from the Internet. The recommendation for a hypothetical cloud provider is to include support for several CMS if possible. If not, it is recommended to support one of the lightweight ones, preferably Salt, and one of the more advanced, preferably Puppet. If it is only possible to include one, it is recommended to support Puppet, since it provides the most comprehensive solution.

Handledare: Joacim Halén
Ämnesgranskare: Lars Jansson, Patrik Karlsson och Alisa Cabrajic
Examinator: Martin Sjödin
ISSN: 1401-5757, TVE 14 013 maj

Popular Science Summary in Swedish

Verktygen som möjliggör moderna IT-lösningar

När vi loggar in på banken, använder sociala medier eller tittar på film över internet utförs mycket arbete som man kanske inte alltid tänker på. Bilder och filmer ska skickas i rätt kvalité, meddelanden ska komma till rätt person och pengar hamna på rätt konto. Detta arbete utförs av en speciell sorts datorer, som kallas för servrar. Vi har inte bara höga krav på att allt ska bli rätt, utan också på att det skall gå extremt fort. För varje applikation som används behöver flera servrar samarbeta för att ge den service som användarna förväntar sig. Idag finns det otroligt många servrar i världen och antalet ökar hela tiden. Till exempel hade Facebook 2012 cirka 180 000 servrar igång för att driva deras system.[1]

Precis som på en vanlig dator behöver olika programvaror installeras och uppdateras på servrarna. Om underhållet skulle göras manuellt skulle det ta väldigt lång tid och bli väldigt dyrt. Stora och komplicerade applikationer som Facebook och Youtube skulle dessutom vara näst intill omöjliga att bygga. Lösningen på detta problem är konfigurationshanteringsverktyg, på engelska kallat configuration management systems (CMS). Dessa verktyg kan automatiskt installera och uppdatera mjukvara på servrar och dessutom ge en bild av hur ett större system sitter ihop.

I rapporten har vi gjort en jämförelse mellan olika CMS till ett företag som erbjuder en lösning för att hyra ut servrar. I en sådan lösning kan dessa mjukvaror vara förinstallerade på servrarna som hyrs ut, så att kunden slipper installationen. Målet med undersökningen var att utreda vilken eller vilka mjukvaror som bör stödjas i en sådan lösning. Först undersöktes 20 olika mjukvaror utifrån olika kriterier, exempelvis hur bra de löste de relevanta uppgifterna och om tillgång till professionell hjälp och aktiva diskussionsforum fanns. Sedan gjordes en jämförelsestudie på de fem mest relevanta mjukvarorna från undersökningen. De som ingick var Puppet, Chef, CFEngine, Ansible och Salt.

Alla fem CMS i jämförelsestudien är bra mjukvaror, men de har lite olika infallsvinklar till uppgifterna och passar olika bra för olika system. Ansible och Salt har utvecklats för att vara enkla att använda och lära sig medan Chef, Puppet och CFEngine är mer komplexa. Denna komplexitet ger dock större flexibilitet för mer avancerade system. Vår slutsats var därför att det bästa vore att erbjuda fler än ett CMS så att fler typer av användare kan täckas. Om möjligheten att erbjuda stöd för flera CMS är begränsad rekommenderar vi att ett av de enklare och ett av de mer komplexa systemen erbjuds, förslagsvis Salt och Puppet. Om bara ett system kan implementeras föreslår vi Puppet, då det är mycket allsidigt och passar ett brett spektrum av system, utan att vara för komplicerat eller svårt att lära sig.

---

[1] Fahrenbacker, Katie. Gigaom. Facebook's number of servers soar to an estimated 180K. 2012-08-13. https://gigaom.com/2012/08/13/facebooks-number-of-servers-soar-to-an-estimated-180k/ (Retrieved 2014-05-14)

Table of Contents

Introduction

Within this section the background for the project is explained, the main concepts of configuration management systems (CMS) are introduced, the purpose of the project is declared and the methods used during the project are explained.

Background

Within this section the main theory required for the understanding of CMS is introduced along with an introduction to virtualisation. The section about virtualisation is included since it is needed for the understanding of CMS.

Virtualisation

The purpose of this project is to compare different CMS and make a recommendation of the most useful software. To understand what these systems are used for and why they are important, we first need to discuss the concepts virtualisation and cloud computing.

Historically one physical computer has had only one operating system installed on it. You could say that the software and the hardware were inseparable. To avoid problems servers were often set up to have one server per application. This means that for every new application a new server had to be configured and managed.[1]

When buying servers you had to make sure that their capacity was designed so that they could withstand the maximal workload. They also had to last for several years, which meant that the servers had some extra capacity, called headroom. Since there often existed a physical server for each application, the machines were not using all of their capabilities, making the electricity bill and the cost for purchasing and maintaining the hardware higher than it could have been with virtualisation. Before virtualisation, it was common that the servers were only using a small fraction of their capabilities.[2]

A more modern solution than the traditional solution described earlier is virtualisation and virtual machines. The reason to use this solution is that it solves many of the problems that occur with the traditional systems, where the hardware and operating systems are more fundamentally linked. With virtualisation, instead of installing the operating system directly on the hardware, there is a layer in between called a hypervisor. Then the operating systems that are to be used in the system can be installed on the hypervisor. These virtual operating systems are called virtual machines and emulate real or hypothetical physical computers. This means that there can exist several operating systems that function independently of each other on one physical computer. [3] A picture that describes this can be seen in figure 1.

---

[1] Portnoy, Matthew. *Virtualization Essentials*. Indianapolis: John Wiley & Sons, 2012. E-book. p.4.
[2] Portnoy, Matthew. *Virtualization Essentials*. p. 3-9.
[3] Ibid p. 19-21

Since the hardware and the software of the virtual machines have the hypervisor layer in between them, it is much easier to migrate a virtual machine to a new physical computer with a hypervisor installed. If for example a physical server is close to 100% utilised, virtual machines can be transferred to a new physical server on the fly, without the end users being aware of the change. This is a very useful property, which improves demand loading and reliability. With virtualisation, the previously describes headroom is no longer needed, since the virtual machines can be transferred quickly when needed. This makes it possible to build more efficient systems.[4]

Virtualisation is one of the concepts that has made cloud computing possible. Cloud computing allows businesses to have their computer infrastructures and applications hosted in an external environment. The actual hardware will be located in data centres managed by a host who makes sure the machines run properly.[5]

For the businesses it will appear as if they have access to the physical machines, even though they are actually virtual machines accessed from the Internet. This gives the businesses opportunity to expand rapidly without having to invest in expensive hardware by purchasing more computer recourses from the host. Traditionally if a company needed to expand its server capabilities, they would have had to buy, make room for and install new hardware, which was time demanding and expensive. Purchasing new hardware was also a risk, since it is hard to foresee exactly how much computer power is needed later on. In a cloud service it is possible to pay only for the computing recourses that are actually being used and expand without making big investments, which makes it possible for start-ups to expand without tanking huge economical risks.

Since a virtualised cloud environment is elastic it is also possible to scale up a business during a short period of time, when the system is heavily loaded and only pay for the extended system while it is being used. Traditionally you would have to scale your server solution after your peeks. Virtualisation and cloud computing has made it possible to make use of the world's computing recourses in a more efficient way.[6]

[4] Knight, William. High noon for virtualisation. *Infosecurity* 6 no. 2 (2009): 42-45. doi: 10.1016/S1754-4548(09)70036-1
[5] Zaigham, Mahmood. *Cloud Computing*. Indianapolis: Wiley Publishing, Inc, 2010. E-book. p.10.
[6] Zaigham, Mahmood. *Cloud Computing*. p. 108-110.

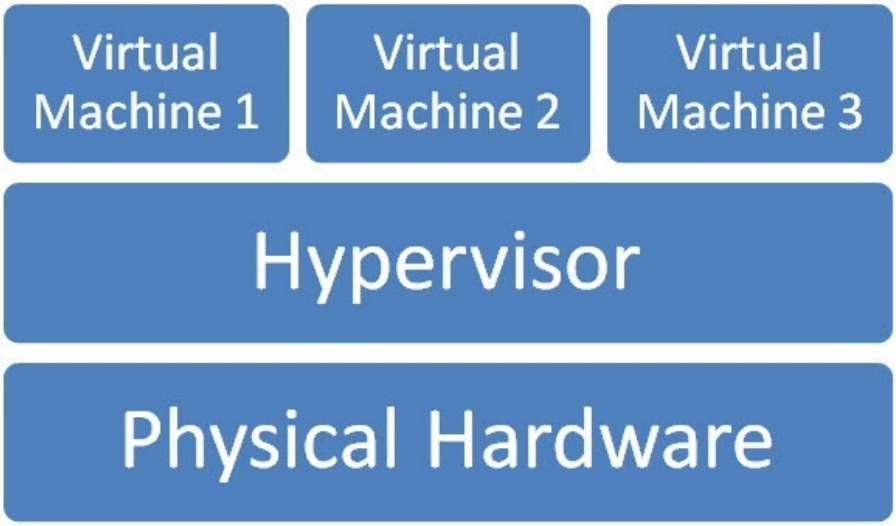

Figure 1: Picture of how hypervisors and virtual machines are installed onto physical servers.[7]

Configuration Management

After a virtual cloud environment with virtual machines has been set up, the software required by the system should also be set up. This means installation, configuration and maintenance for the virtual system. While the software needed could be installed and configured by hand, there is a risk that this may lead to unnecessary complexity in the system, making maintenance difficult and expensive. This can also lead to problems when trying to expand the service, or even developing and testing the system.

CMS, such as for example Puppet and Chef, was designed to solve these tasks in a consistent and reliable way, making it easier to configure and install software, as well as making the system more prepared for expansion or modification. These tools simplify the task of managing large and complex compute deployments and keeping the system up to date. They also make it possible to keep operation costs low. The main tasks of a CMS are to automatically install and upgrade software on the machines within the system. They can also keep track of the state of the system and provide an overview of the software installed.

The two main models for managing the configuration of a system used by the CMS are the standalone model and the master-agent model. Which model that is preferable depends on the managed system and on the preferences of the users.

In a master-agent model there is a master server, connected to a database that keeps track of the system and the software within it. Its main job is to provide the necessary configuration information to all the clients within the system. A client is a machine, virtual or real, that should be managed by the master server. The clients report their current states to the master, so that the state of the whole system can be tracked and viewed. An administrator can connect to the master from a workstation or a laptop and make changes to the master. In this model the

[7] Picture from: Dominey, Andy. 1E Blogs – Insight and debate on IT Efficiency. The Bad Kind Of Sprawl. 2010-03-15. http://blogs.1e.com/2010/03/15/the-bad-kind-of-sprawl/ (Retrieved 2014-05-14)

configurations can be implemented by either a push or pull model. A picture of the master-agent model can be seen in figure 2.

In a pull model the agents installed on the clients connect to the master periodically asking for new configurations. In a push configuration the master has the ability to deploy configuration information to the clients anytime. Some systems supports a combination of push and pull, where the clients connect periodically just like a pull model. However, in addition to that, the master can connect to the clients to provide the needed configuration as in a push model.

In the standalone model no master is used and the clients are connected to one or more workstations within the environment. Changes of the system's configuration information can be pushed from the workstations to the clients. It is possible to use a variant of the standalone model where it becomes more similar to a pull model. The idea is to push configuration information from a workstation to a central repository, for example available on GitHub, a popular software uploading service. Then the clients that should be managed uses simple scheduling tools to connect to the repository, from which they pull and apply the latest configurations on a regular basis.[8] A picture of the standalone model can be seen in figure 3.

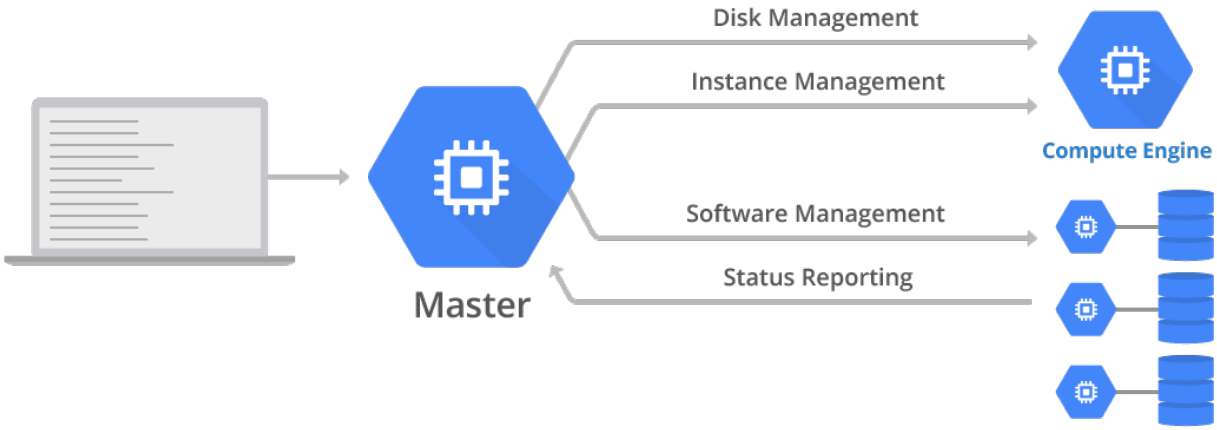

Figure 2: Schematic picture of the master-agent model.[9]

[8] Google. Google Cloud Platform Compute Engine Management with Puppet, Chef, Salt, and Ansible. https://cloud.google.com/developers/articles/google-compute-engine-management-puppet-chef-salt-ansible (Retrieved 2014-05-14)
[9] Picture from: Google. Google Cloud Platform Compute Engine Management with Puppet, Chef, Salt, and Ansible. https://cloud.google.com/developers/articles/google-compute-engine-management-puppet-chef-salt-ansible (Retrieved 2014-05-14)

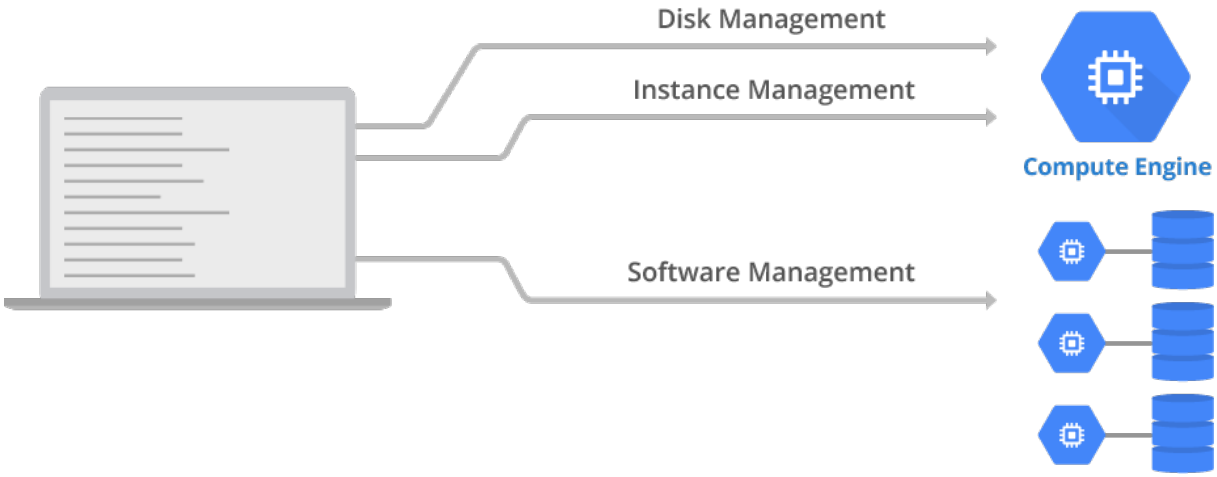

Figure 3: Schematic picture of the standalone model.[10]

Purpose

Cloud providers such as for example Amazon often want to offer servers that are pre-configured with different software. One kind of software that can be pre-installed on the servers is CMS. The purpose of our study is to compare different CMS alternatives for a company that considers selling hosted computer solutions. A recommendation for the CMS that we think should be supported in the hosted solution will be presented.

Method

The goal of the study is to compare different CMS and to recommend the software that we find to be best suited for our scenario. First a general study was made on twenty different software that offer configuration management capabilities. The study was based on the parameters describes in section Parameters for the study. In addition to the general study a comparison study was made on only the five that we found most useful.

We quickly discovered that there was not much information about the CMS available from scientific papers. Therefore the information gathering process for the general study was done by reading information available on the Internet, mainly from the official webpages of the different CMS. For the comparison study we also used opinions found from articles and blog post. Since we wanted to include opinions from users of the software, we decided to include these opinions, even if they are not always from reliable sources.

Parameters for the study

The most important parameter is if the software can accomplish the configuration management task our supervisor was interested in. The programming language and licenses is a parameter that can be important to consider for users. All the systems

[10] Picture from: Google. Google Cloud Platform Compute Engine Management with Puppet, Chef, Salt, and Ansible. https://cloud.google.com/developers/articles/google-compute-engine-management-puppet-chef-salt-ansible (Retrieved 2014-05-14)

in this study have an open source version available under different licenses. Open source means that the software is free and that it is possible to access the source code. However, it is not meaningful to evaluate software based on the licenses or the programming language, since it is subjective whether a certain license or programming language is good or bad. The platform support is important since it affects how flexible the CMS is and how many different systems are possible to manage with the software. We evaluated the support for Windows, Linux and Unix-based operating systems. Another important aspect is how much software that has to be installed in order for the system to work, such as for example agent software on the clients. This is important since it affects how quickly it is possible to set up the system.

A commercial version with additional features and the possibility to get professional support is important, especially for companies that want to be able to quickly implement the software in their systems. Active communities and forums show how popular and widely used the CMS are. They also provide the possibility to discuss and ask questions, which can be very important for the learning process of the new software. Another important parameter is if there are any big companies using the software, since it shows how popular the software is and the usefulness for larger deployments. We also added how long it has been since the last update as a parameter to evaluate how up to date the software is.

For the comparison study of the five most interesting options, we decided not to search for answer for a certain set of parameters. Instead we tried to conclude the opinions of different users from articles and blog posts available on the Internet. We decided to put focus on the different aspects that distinguishes the CMS from each other, instead of writing about the same parameters for each of them.

The following parameters were evaluated in the general study.

- If it does the configuration management tasks that we are interested in, such as automatic update and installation.
- The programming language when using the software.
- The programming language used for the development of the CMS.
- How much software (such as for example agents) that has to be installed onto the clients and the master server.
- The license it is distributed under.
- The platform support.
- If a commercial version exists and the possibilities to get support.
- Access to active communities, forums and other community features.
- Popularity and if any well established companies are using it.
- How long it has been since the latest release.

Results

This section contains the general and the comparison study of the CMS that were among the five most interesting ones in the general study. A summary of the results

of the general study can be seen in the Table 1 and a summary of the comparison study can be seen in Table 2. The section Results for the CMS that were not among the five most interesting ones can be found in the appendix.

Table 1: Summary of the general study.

	Functionality according to setup	Active community	Commercial support	Latest update	Windows support	Unix support	Mac OSX support	Linux support	Popularity on the market	Comment on further studies
Ansible	Yes	Yes	Yes	apr 2014	No	Yes	Support by independent packages	Yes	Several big companies	Fulfills most of our criteria. Since it does not use daemons we will do a more in-depth study on it
BCFG2	Yes	Yes	No	feb 2014	No	Yes	Yes	Yes	No information of companies currently using the software	Fulfills a lot of our criteria but other similar software are better. No further studies
Cdist	Yes	Has an active mailing list	Support available from small company	apr 2014	No	Yes	Yes	Yes	No information of companies currently using the software	It fulfills some of our criteria but comes from a small developer and does not seem to have a lot of users. No further studies.
CFEngine	Yes	Yes	Yes	dec 2013	Yes	Yes	Yes	Yes	Several big companies	Since it fulfills our parameters, we will do a more in-depth study on it.
Chef	Yes	Yes	Yes	apr 2014	Yes	Yes	Yes	Yes	Several big companies	Since it fulfills our parameters, we will do a more in-depth study on it.
Isconf	Complicated update routines	No	No	2006	No information available	Yes	No information available	No information available	No information of companies currently using the software	Complicated update routines lack of platform support, dated software. No further studies.
Juju	Can be run together with other configuration management software	Yes	Yes	jun 2013	No	No	No	Bound to ubuntu	No information of companies currently using the software	Does not focus on configuration management. No further studies.
LCFG	Yes	Has an active mailing list	No	weekly updates	No	Yes	No	Yes	No information of companies currently using the software	Does not seem to bee a modern software since it has been replaced and there is no information about companies using it. No further studies
OCS Inventory	Has software inventory as main focus and lacks automation functionality	Yes	Yes	feb 2014	Yes	Yes	Yes	Yes	No information of companies currently using the software	The software is mainly designed for hardware inventory. No further studies.
OPSI	Yes	Yes	Yes	feb 2014	Yes	No	No	No	Has some users within Europe	Does only manage windows clients which is limiting. No further studies.
PIKT	No	No	No	2007	No	No	No	Yes	Claims to have big customers but gives no company names	Not relevant to our study.
Puppet	Yes	Yes	Yes	feb 2014	Yes	Yes	Yes	Yes	Several big companies	Since it fulfills our parameters, we will do a more in-depth study on it.
Quattor	Yes	Has an active mailing list	No	mar 2014	No	Yes	No	Yes	No information of companies currently using the software	It was developed for a project that now has switched to puppet. No further studies.
Radmind	Can only manage Mac computers	Has an active mailing list	No	2010	No	No	Yes	No	No information of companies currently using the software	Can only manage Mac computers. No further studies.
Rex	Yes	Yes		apr 2014	No	Yes	No	Yes	No information of companies currently using the software	Not very well documented and lack of big companies using it. No further studies.
Rundeck	Can be run together with other configuration management software	Yes	Yes	april 2014	No	Yes	Yes	Yes	There is information about one company using the software	Does not focus on configuration management. No further studies.
Spacewalk/ Satellite	Yes	Yes	Yed	Satellite oct 2014 / Spacewalk mars 2013	No	No	No	Yes	Yes	Has a lot of the require functionality but does only support a few linux based platforms. No further studies.
Salt	Yes	Yes	Yes	april 2014	Yes	Yes	Yes	Yes	Several big companies	Since it fulfills our parameters, we will do a more in-depth study on it.
Smartfrog	Yes	Mailing list used during 2013	No	2007	Yes	Yes	Yes	Yes	No information of companies currently using the software	Limited platform support and issues with domain specific language. Therefore no further studies.
STAF	Built for test environments	Yes	No	2014	Yes	Yes	Yes	Yes	No information of companies currently using the software	Focuses on build environments. No further studies.

Table 2: Summary of the comparison study.

	Pros	Cons	Annual price example for 100 nodes with Enterprise version
Ansible	• Simple architecture • Easy to learn	• No windows support • Might not work well with large or complex systems • GUI not as developed as Puppet and Chef's	Ansible Tower $3,000 or Ansible Tower Enterprise $10,000 for 100 managed machines
CFEngine	• Very fast and efficient	• Hard to learn and use • Other alternatives gets more attention	$10,000 ($5,000 for non profit educational institutions)
Chef	• Flexible and feature rich	• Requires Ruby knowledge • Steep learning curve	$7,200
Puppet	• Well developed GUI • Offers capable solution • Most established on the market	• Can be hard to configure	$10,500
Salt	• Fast and scalable • Easy to learn and use • All functionality available in open scours version	• GUI only released in pre-alpha version	$15,000

Ansible

General study

Development on Ansible started in 2012, which makes it one of the later CMS that has been released. The project was started by Michael DeHaan, who wanted to simplify Puppet and by doing that bring it to the masses. On Github, Ansible has gotten a lot of attention and is the CMS that is most starred and forked.[11] The latest release was made available on Github on April 18 2014.[12]

Ansible does not use daemons and therefore there is no need to install anything on the remote machines. Ansible only has to be installed on the one machine that should control the other machines. Ansible configures the system by a standalone model with a control machine that can connect to the servers using SSH. In order to run Ansible from a machine it is required that python 2.6 is installed, but that is the only requirement. Windows is not supported as a control machine though. For the nodes it is required that python 2.4 or later is installed, which means that OSX, Linux and Unix is supported.[13] It is currently not possible to manage Windows remote hosts, but according to Michael DeHaan, who founded Ansible, it is something that they want to implement later on.[14] Ansible is written in Python and is one of the top ten python projects on Github.[15]

Ansible is open source, but there exists several commercial versions where the price depends on what kind of service the customer is interested in. They offer a product called Guru, which is a support plan for those who want to use the command line

[11] DeHaan, Michael. Ansible Blog. The Origins of Ansible. 2013-12-08. http://www.ansible.com/blog/2013/12/08/the-origins-of-ansible (Retrieved 2014-04-23)
[12] GitHub. Ansible. 2014. https://github.com/ansible/ansible/releases (Retrieved 2014-04-23)
[13] Ansible. Ansible Docs Installation. 2014. http://docs.ansible.com/intro_installation.html (Retrieved 2014-04-23)
[14] DeHaan, Michael. Re: [ansible-project] Re: ansible on windows - on the roadmap?. Ansible Project Forum. 2013-12-13. https://groups.google.com/forum/#!topic/ansible-project/ayJ3nd0YTIc (Retrieved 2014-04-24)
[15] Ansible. Ansible. 2014. http://www.ansible.com/home (Retrieved 2014-04-23)

tool, which costs 99$ per user and month.[16] They also offer a product called Ansible Tower, which is a Web UI that gives you the possibility to have role-based access control. It also gives the possibility to monitor and administer the deployments in more detail.[17] Ansible Tower is free for up to 10 managed machines. It costs $3,000 per year for 100 managed machines and include technical support. They also offer a program called Enterprise Tower, which includes everything Ansible Tower offers, but adds phone support and a service license agreement. For 100 managed machines the price is $10,000 annually. [18]

Ansibles' configuration management language is written in scripts called playbooks. They are written in the YAML-format, which is a very simple format, and describe the state that the system should be in.[19] Ansible is goal oriented, which means that the user does not have to write the code that changes the system. Instead the desired system is expressed in the playbooks and then Ansible can transform the system into the state that is described.[20] Each playbook consists of one or more plays in a list. The goal of each play is to map groups of hosts into a defined state. The tasks in a play are executed one at a time and in order against all the defined hosts. Ansible uses so called tasks, which are calls to Ansible modules, to make this possible.[21] Modules control system resources, such as services or packages. All modules return JSON format data. This means that it is possible to write new modules in any language as long as it returns data in the JSON format. In Ansible, modules are idempotent, which means that they will not make changes to the system if it is not necessary. That means that if the system already is in the right state, the module will not do anything. That makes it safe to run a playbook several times.[22]

Ansible has been downloaded over one million times and is being used by some established companies like GoPro and Evernote.[23] There is an active forum and a mailing list available.[24] Ansible has released a beta of Ansible Galaxy, which is a hub for user made content. It is possible to download and review other users Playbooks and to upload Playbooks for other Ansible users.[25]

Comparison study

[16] Ansible. Ansible Pricing. 2014. http://www.ansible.com/pricing (Retrieved 2014-04-23)
[17] Ansible. Ansible Tower. 2014. http://www.ansible.com/tower (Retrieved 2014-04-29)
[18] Ansible. Ansible Pricing. 2014. http://www.ansible.com/pricing (Retrieved 2014-04-29)
[19] Ansible. Ansible Docs Intro to Playbooks. 2014. http://docs.ansible.com/playbooks_intro.html (Retrieved 2014-03-23)
[20] Ansible. Configuration Management. 2014. http://www.ansible.com/configuration-management (Retrieved 2014-03-24)
[21] Ansible. Ansible Docs Intro to Playbooks. 2014. http://docs.ansible.com/playbooks_intro.html (Retrieved 2014-03-23)
[22] Ansible. Ansible Docs About Modules. 2014. http://docs.ansible.com/modules.html (Retrieved 2014-04-28)
[23] Ansible. Ansible. 2014. http://www.ansible.com/home (Retrieved 2014-04-24)
[24] Ansible. Ansible Project Forum. 2014. https://groups.google.com/forum/#!forum/ansible-project (Retrieved 2014-04-25)
[25] Ansible. Ansible Galaxy. 2013. https://galaxy.ansible.com (Retrieved 2014-04-28)

Perhaps the most defining quality of Ansible is its simplicity. Michael DeHaan begun development on Ansible as a reaction to Puppet, which he had used but thought was to complex. And there are a few voices on the Internet that embraces Ansible's simpler approach to configuration management. Mark Phillips is one of these persons as he writes on his blog.[26] For over six years he considered Puppet to be his preferred CMS. He has worked with Puppet in some established infrastructures, used it for quite a lot of clients and has been generally pleased with it.

But when he tested out Ansible he was so impressed by how simple it was that he now states that he cannot think about doing configuration management with anyone else. He liked that no daemons were used, that it was built around Python instead of Ruby, that it used a push instead of a pull model and the fact that the connections were made using SSH, which lets him skip the complex SSL certificate management that he previously had to do. In fact, he claims that he has not yet found anything with Ansible that is problematic, at least not with the systems that he currently uses. He claims that Ansible is brilliant for small to medium enterprises and for startups.

For larger systems though, he points out that Ansible might not be as simple and efficient. Michael DeHaan, founder of Ansible, claims in a comment for this blog post on 16 October 2013 that there actually are a few projects with around 5000 nodes that use Ansible. However, other CMS, such as for example CFEngine, has a much longer history of users with very big systems. Even if DeHaan claims that there should be no problem with managing bigger systems, there are fewer examples of large systems using Ansible. And while this can at least partly be explained by the fact that Ansible is newer, the fact that the competition offers more proven solution remains.

According to Paul Venezia, who wrote a comparison of Puppet, Chef, Ansible and Salt for ComputerWorld, there is a difference in what audience these four tools are designed for.[27] Ansible and Salt are more focused on system administrators and therefore focuses on simple setup, intuitive interfaces and direct usability. Puppet and Chef are more interesting for developers or development-oriented companies. This means that while Ansible is well adapted for the needs of a system administrator, it may not be equally well suited for complicated systems under development.

Another problem with Ansible that Venezia points out in the article is that the graphical user interface (GUI) Tower (previously called AWX) is not directly tied to the command line interface. This means that in order for the GUI to have the latest information it has to be synchronized, which can be done on a regular basis.

---

[26] Phillips, Mark. Probably. Puppet vs Chef vs Ansible. 2013-10-01. http://probably.co.uk/puppet-vs-chef-vs-ansible.html (Retrieved 2014-05-05)

[27] Venezia, Paul. Review: Puppet vs. Chef vs. Ansible vs. Salt. *InfoWorld*. 2013-11-21. http://www.infoworld.com/d/data-center/review-puppet-vs-chef-vs-ansible-vs-salt-231308 (Retrieved 2014-05-15)

Steve Hall has written an article for the website ScriptRock.[28] It compares Ansible to Puppet and he writes that he shares the opinion with Venezia regarding the GUI available for Ansible. He points out that it is not as developed as Puppet's version. In another article published on ScriptRock comparing Ansible to Salt, Steve Hall claims that some consider Ansible's setup with no master and just pure SSH to be more secure than master-agent models.[29] He also mentions that while Ansible uses SSH for the connections, Salt uses zeroMQ that does not include native encryption. This means that Salt has to include another AES (Advanced Encryption Standard). Jens Rantin agrees with Hall on this matter as he writes on a blog post where he compares Ansible with Salt on March 17 2014.[30] He points out that Salt has had security problems before.

CFEngine

General study

Mark Burgess started the CFEngine project already in 1993 at Oslo University. The purpose of the project was initially to automate the management of a few workstations at the Department of Theoretical Physics. In 2008 Mark Burgess left the University and founded a company to be able to develop CFEngine further and finance future work in the area.[31] CFEngine has been developed since then and is still getting updates and improvements.[32]

The desired state of the system that is to be configured and managed by CFEngine is expressed as promises written in the CFEngine policy language. A collection of promises make a policy that defines how the system should be set up.[33] CFEngine is written in C, uses a DSL and has agents that run on each host. CFEngine's security is built on SSH. The system is built based on the desired state and CFEngine can also maintain the system over time. CFEngine compares the system in real time with the desired state and can automatically make changes if the system is in the wrong state. A pull based approach is used and once a policy has been deployed, the clients keep all the discovered facts locally. This minimizes the strain on the network since the client can make decisions without connecting to the master. This leads to a system that can function even if the network becomes temporarily unavailable.[34]

---

[28] Hall, Steve. ScriptRock. Ansible vs Puppet. 2014-02-27. https://www.scriptrock.com/articles/ansible-puppet/ (Retrieved 2014-05-06)
[29] Hall, Steve. ScriptRock. Ansible vs. Salt. 2014-01-10. http://www.scriptrock.com/articles/ansible-vs-salt (Retrieved 2014-05-20)
[30] Rantil, Jens. Jens Rantil's Blog. Salt vs. Ansible. 2014-03-17. http://jensrantil.github.io/salt-vs-ansible.html (Retrieved 2014-05-07)

[31] Burgess, Mark. homepage mark burgess. Biographical Information. http://markburgess.org/bio.html (Retrieved 2014-04-23)
[32] GitHub. CfEngine core. 2014. https://github.com/cfengine/core/releases (Retrieved 2014-04-23)
[33] CFEngine. CFEngine Docs 3.5. 2014. https://cfengine.com/docs/3.5/index.html (Retrieved 2014-04-23)
[34] CFEngine. What is CFEngine?. 2014. https://cfengine.com/what-is-cfengine (Retrieved 2014-04-23)

CFEngine supports the most popular operating systems, such as the common Linux distributions, Unix, Windows and Mac OSX.[35] The software is available both as an open source version and an enterprise version that is free for up to 25 servers.[36] The open source version, which is called CFEngine Community, is licensed under the GNU GPL v.3 license.[37]

The enterprise version provides professional support and implementation consultation. There are also some additional features within the enterprise version such as reporting and monitoring functionality and a GUI. Another Enterprise feature is the possibility to create role based user access control. The CFEngine Enterprise also provides support for specialized platforms and a version control feature.[38] The price for 100 servers with CFEngine Enterprise is $10,000 but for non-profit educational institutions the price is $5,000 for 100 nodes. A node is network-connected devise, such as a server or a workstation.[39]

There is an official forum for the community where different problems and topics can be discussed.[40] Many big companies, such as Google, IBM, Pixar, Intel, ebay and Facebook, use CFEngine.[41]

Comparison study

One interesting aspect of CFEngine that really distinguishes it from the other CMS available on the market is that it is based on C. Most competitors have been developed using higher level programming languages, such as Python or Ruby.

Since C is a low-level language it should, at least in theory, be possible to make the software both more efficient and smaller. And according to CFEngine themselves that is also the case. They claim that CFEngine 3 is only a tenth of the size of Puppet. They also claim that it is up to 40 times more efficient, but the source that they refer to is not available to read for free.[42]

There is however a study on this topic available. The blog Blogcompiler[43] did a comparison of the efficiency of Puppet and CFEngine on September 30 2012, which

---

[35] CFEngine. CFEngone Cmoparison. 2014. https://cfengine.com/cfengine-comparison (Retrieved 2014-04-23)
[36] CFEngine. CFEngine Docs Getting Started Installing CFEngine. 2014. https://cfengine.com/docs/3.5/getting-started-installation.html (Retrieved 2014-04-23)
[37] CFEngine. Welcome to the CFEngine Community. 2014. https://cfengine.com/community (Retrieved 2014-04-23)
[38] CFEngine. CFEngone Cmoparison. 2014. https://cfengine.com/cfengine-comparison (Retrieved 2014-04-23)
[39] Private conversation with Ed Howard at CFEngine.
[40] CFEngine. help-cfengine Forum. 2014. https://groups.google.com/forum/#!forum/help-cfengine (Retrieved 2014-04-23)
[41] CFEngine. CFEngine. 2014. http://cfengine.com (Retrieved 2014-04-23)
[42] CFEngine. CFEngine Tech FAQ. 2014. https://cfengine.com/techFaq#puppet (Retrieved 2014-04-29)
[43] Blogcompiler. {Collecting Solutions}. Scalability of CFEngine 3.3.5 and Puppet 2.7.19. 2012-09-30. http://www.blogcompiler.com/2012/09/30/scalability-of-cfengine-and-puppet-2/ (Retrieved 2014-04-29)

makes it a bit dated. But since neither Puppet nor CFEngine has changed their system dramatically since 2012, the comparison should still have some relevance, even if it is probably not completely accurate anymore. It should also be noted that the author is affiliated with CFEngine, which means that the experiment is made by a non-objective part. And since different solutions use different environments, this experiment does not map exactly to other environments. Therefore the interesting result is the overall trends of the systems and not the exact numbers. It should also be taken into consideration that the experiment uses very simple manifests and policies, leading to a quite non-realistic experiment. The exact setup of the experiment is available and presented in the blog, which means that it is possible to reproduce the experiment to verify it.

In the experiment 50 clients were added every 15 minutes until 300 clients were used. The Puppet Manifest and CFEngine policy were changes twice during the test by adding 100 echo commands each time. This was done in order to make it possible to examine how the tools handled a simple increase in workload. The results show a huge difference in efficiency. At 50 clients Puppet uses 10 times as much CPU and runs 20 times slower than CFEngine. At 300 clients with 200 echo commands Puppet uses 18 times as much CPU and runs 166 times slower than CFEngine. This is because of two reasons, according to the author. Firstly, Puppet uses a much more centralized architecture than CFEngine, which means that the Puppet master server does a lot of work for each client, while the CFEngine server functions as a file server. Secondly Puppet uses a Ruby interpreter, which is less efficient than CFEngine's C-based approach.

There are some drawbacks with CFEngine however. Mike Baukes did a comparison of Puppet and CFEngine in a blog post on the webpage Scriptrock on May 16, 2013, where he claims that one of the main complaint that people have about CFEngine is that its learning curve is very steep.[44] Puppet has, he claims, a more model-driven approach that takes control over dependencies management in the system, which makes it easier to use. However, according to Baukes, some argue that Puppets system is somewhat limited and might result in unexpected behavior.

This picture of CFEngine being a powerful but complicated tool is further sustained by Jacques Chester, who wrote a blog post on June 27 2012 about CFEngine, Puppet and Chef.[45] He writes that he has tried out the three CMS and that he in the end prefers Puppet. He had two problems with CFEngine. Firstly, he did not like its DSL. According to Chester, there was too much unnecessary syntax and sometimes he found the naming to be confusing. The second and biggest problem for Chester was that he did not like CFEngine's underlying model. CFEngine uses policies to gradually converge to the desired state and Chester found it to be both time consuming and difficult to learn. On November 10, 2012 Brian P O'Rourke commented on Chester's blog post and stated that he, just like Jacques Chester, found the user experience to be much better with Puppet; but in the end he used

[44] Baukes, Mike. ScriptRock. Puppet vs CFEngine. 2013-05-16. https://www.scriptrock.com/blog/puppet-vs-cfengine/ (Retrieved 2014-05-05)
[45] Chester, Jacques. Journal de Jacques. A not-so-brief aside on reigning in chaos. 2012-06-27. http://chester.id.au/2012/06/27/a-not-sobrief-aside-on-reigning-in-chaos/ (Retrieved 2014-05-05)

CFEngine since the Puppet system used too much resources. This supports the earlier ideas presented in this discussion that CFEngine is a fast and efficient system, but that it has a steep learning curve.

Chef

General study

The company Chef (formerly Opscode) was founded in 2008 and is the company behind the CMS Chef.[46] Jesse Robbins, Adam Jacob, Barry Steinglass and Nathan Haneysmith founded the company.[47] Since then Chef has been developed and receives updates on a regular basis. The latest release on the open source version on Github was on April 23 2014.[48] In 2013 sales grew by 188 percent and their customers include for example Facebook and General Electric. In fact, 70% of their sales come from fortune 1000 companies.[49]

The open source version on Chef is available under an Apache license.[50] The Chef infrastructure consists of three main elements: a chef server, at least one workstation and one or more clients. The Chef server is the central point in the system and is available to every node. It stores cookbooks and recipes, which are abstract definitions written in Ruby that describe how different parts of the system should be built and managed. The nodes are the virtual or physical machines that are to be managed by the Chef server. Each node has to have a client installed that makes sure that the node is in the state that the cookbooks and recipes on the chef server have defined. A workstation is a computer that interacts with a single chef server. It contains a Chef-repo that contains the important files for the project and should be used with version control software, such as for example Git. The workstation uses the command line tool knife that provides an interface between the local chef-repo and the server. This means that developers can work at their workstations and then export changes to the Chef server when new functionality is ready. The connections required for the communications use RSA public key-pairs.[51] Chef can also be run in a standalone mode, without a master server.[52]

Chef supports most common platforms, such as the common Linux distributions, Mac OSX, Unix and Windows.[53] Being an open source software Chef can be used for free, but Enterprise Chef is also offered. It is a product that includes some

[46] LinkedIn. Chef. 2014. http://www.linkedin.com/company/opscode (Retrieved 2014-04-24)
[47] Chef. Chef Executive Team. 2014. http://www.getchef.com/executive-team/ (Retrieved 2014-04-24)
[48] GitHub. opscode chef. 2014. https://github.com/opscode/chef/releases (Retrieved 2014-04-24)
[49] Cook, John. IT automation startup Chef plans to double staff as sales continue to rise. *GeekWire*. 2014-02-12. http://www.geekwire.com/2014/automation-startup-chef-plans-double-staff-sales-continue-rise/ (Retrieved 2014-04-24)
[50] Chef. Why We Choose The Apache License. 2009. http://www.getchef.com/blog/2009/08/11/why-we-chose-the-apache-license/ (Retrieved 2014-04-24)
[51] Chef. Chef Documents An Overview of Chef. http://docs.opscode.com/chef_overview.html (Retrieved 2014-04-24)
[52] Chef. Chef Documents Standalone. http://docs.opscode.com/server_deploy_standalone.html (Retrieved 2014-04-24)
[53] Chef. Install Chef. 2014. http://www.getchef.com/chef/install/ (Retrieved 2014-04-24)

additional functionality, which is free for up to 5 nodes. For more than 5 nodes the price depends on the number of nodes that are to be managed. For example 100 nodes costs $7,200 annually. With Enterprise Chef you get standard support, but it is possible to buy extra support on top of the standard fee.[54] An additional feature that comes with the enterprise version is role based access control, which means that different people can log in to the system with different authorization. A non-administrator user can log into the Chef system by a web interface.[55] The enterprise version of Chef supports push jobs, which makes it possible to execute a command or an action on a node directly, without having to wait for a scheduled run.[56] Enterprise Chef also comes with an install-reporting feature, which gives reports of what happens during a Chef-execution.[57]

Chef offers a hosted solution called Hosted Enterprise Chef, where Chef hosts the Chef server in a cloud solution. It has the same automation capabilities as other Chef servers, but does not need to be set up by the customer.[58]

Chef offers extensive documentation on their homepage.[59] They also offer a mailing list, a few IRC channels, a bi-weekly podcast and a collection of popular cookbooks that are widely used by the community and available to anyone using Chef.[60]

Comparison study

In articles and blogs on the Internet it is very common that Chef is compared to Puppet, which is expected since they both are based on Ruby. The fact that Chef was released a few years after Puppet as an alternative to it also explains why these comparisons are so common. The two main differences between Chef and Puppet are their languages and the way they handle dependencies.

While Puppet uses a DSL built on Ruby, Chef is configured with Ruby scripts. Which approach is preferable have been discussed in many articles. Mark Phillips, who wrote an article comparing Chef, Puppet and Ansible, stated that the fact that Chef uses Ruby makes it hard to understand for people who are not experienced programmers.[61] In a comparison article between Puppet, Chef, Salt and Ansible at InfoWorld written by Paul Venezia, he states that Chef has a steep learning curve and that it requires Ruby knowledge. He also points out that Puppet is easier to

[54] Chef. Enterprise Chef. 2014. http://www.getchef.com/enterprise-chef/ (retrieved 2014-04-24)
[55] Chef. Chef Documents Authorization. http://docs.opscode.com/auth_authorization.html (Retrieved 2014-05-28)
[56] Chef. Chef Documents Push Jobs. http://docs.opscode.com/push_jobs.html (Retrieved 2014-05-28)
[57] Chef. Chef Documents Install Reporting. http://docs.opscode.com/install_reporting.html (Retrieved 2014-05-28)
[58] Chef. Chef Documents An Overview of Chef. http://docs.opscode.com/chef_overview.html (Retrieved 2014-04-24)
[59] Chef. Chef Documents All about Chef … http://docs.opscode.com (Retrieved 2014-04-24)
[60] Chef. Chef Community. 2014. http://www.getchef.com/community/ (Retrieved 2014-04-25)
[61] Phillips, Mark. Probably. Puppet vs Chef vs Ansible. 2013-09-01. http://probably.co.uk/puppet-vs-chef-vs-ansible.html (Retrieved 2014-05-09)

understand.[62] Jacques Chester wrote about another language-based concern with Chef in a blog post.[63] He claims that Chef's metaphoric wordplays can be confusing, especially since they sometimes use descriptive names and other times not. However, for people used to Ruby Chef might be better. Nathen Harvey wrote in a blog post about why his company CustomInc chose to replace Puppet with Chef. One of the reasons was that the Chef language came more natural to them, since they were used to working with Ruby.[64]

One of Chef's core principles is that the users themselves are the ones who know best how their environment should be designed and maintained. Chef is designed so that the clients do not make any assumptions about these things. In contrast to Puppet, Chef offers no dependency management, but in return they can guarantee that the resources are always applied in the same order for every run. In Puppet dependencies can be stated separately. Puppet then makes sure that things are executed in an order that works.[65] Chester wrote in the same blog post that were referred to earlier, that Chef's requirement for scripts being written in the right order gives the developers a lot of responsibility. He prefers Puppet's approach where the system can help with dependencies. However, whether this is good or bad design is of course up to the user to decide. Harvey, who wrote about his company CustomInc, found Chef's principles, where the user writes everything in the right order instead of modelling dependencies, to be both easier and better.

Steve Hall compared Salt to Chef in an article at ScriptRock on February 19 2014.[66] He writes that Chef can be hard to learn and deploy for beginners and that the documentation needs a lot of work, since it is hard to understand. However, he claims that Chef is a more mature solution with a much better GUI and a large collection of modules and recipes. Venezia also mentioned the poor documentation as one of the downsides with Chef in his comparison study on InfoWorld. He claims that Ansible and Salt are better suited for the needs of system administrators, while Chef and Puppet are better options for developers. Out of the software in the comparison he finds Chef to be the most difficult option to learn for non-programmers. However, he claims that it is well designed and that its pure Ruby approach makes it powerful. He points out that it might be the most natural fit for development-minded administrators or developers.

---

[62] Venezia, Paul. Review: Puppet vs. Chef vs. Ansible vs. Salt. *InfoWorld*. 2013-11-21. http://www.infoworld.com/d/data-center/review-puppet-vs-chef-vs-ansible-vs-salt-231308 (Retrieved 2014-05-09)

[63] Chester, Jacques. Journal de Jacques. A not-so-brief aside on reigning in chaos. 2012-06-27. http://chester.id.au/2012/06/27/a-not-sobrief-aside-on-reigning-in-chaos/ (Retrieved 2014-05-15)

[64] Harvey, Nathen. Nathen Harvey's blog. Why We Chose Chef Over Puppet at CustomInk. 2011-11-21. http://www.nathenharvey.com/blog/2011/11/21/why-we-chose-chef-over-puppet-at-customink (Retrieved 2014-05-09)

[65] Chef. Chef Documents Why Chef?. http://docs.opscode.com/chef_why.html (Retrieved 2014-05-09)

[66] Hall, Steve. ScriptRock. Salt vs. Chef. 2014-02-19. http://www.scriptrock.com/articles/salt-vs-chef/ (Retrieved 2014-05-09)

Puppet

General study

Luke Kanies founded puppet and Puppet Labs, the company behind Puppet, in 2005. The goal that Kanies had with Puppet Labs was to make better operating tools and improve management of systems.[67] Since its first release, Puppet has been updates and developed and the latest release was on April 16 2014.[68] Puppet is still in development and has many big companies as customers, such as at&t, Intel, Nasa, twitter and many more.[69]

Puppet is Ruby based and developed under the Apache-license. Puppet can be run in standalone mode, where no Puppet master is used, but usually a master-agent model is used. With this set-up, the Puppet master contains the configuration info for the clients. Puppet master daemons run on the clients and connect to the master via SSL to get information about their configuration. The clients are configured if they are not in the appropriate state, otherwise Puppet will do nothing. The idea is that the user declares which state the system should be in and then Puppet automatically sorts out the rest. By default a Puppet agent connects to the master every 30 minutes, but that can be changed if needed.[70]

Puppet supports the common platforms, such as the common Linux distributions, Unix, Mac OSX and Windows. Windows is only supported for clients though.[71] Puppet offers an open source version of the software. However, there is a commercial version of Puppet called Puppet Enterprise available. It provides services that are not available within the open source version. One of the features available is a GUI. It also provides the possibility to create different user accounts with different user settings.[72] Another feature is support for working with VMware machines and tools to set up VMware machines quickly.[73]

With Puppet Enterprise there is also a feature called Puppet Orchestration that provides a more detailed overview of the Puppet system. Normally the Puppet agents run in the background and order a run every 30 minutes, but with Puppet Orchestration it is possible to take control over this behaviour. It is possible to start, stop, enable and disable agents, view the status of any number of clients or get statistics from the latest runs.[74] With the orchestration tool it is also possible to

---

[67] Puppet Labs. Management Team. 2014. http://puppetlabs.com/about/management (Retrieved 2014-04-24)
[68] Puppet Labs. Docs: Puppet 3.5 Release Notes. 2014. http://docs.puppetlabs.com/puppet/3.5/reference/release_notes.html (Retrieved 2014-04-24)
[69] Puppet Labs. Customers. 2014. https://puppetlabs.com/about/customers (Retrieved 2014-04-24)
[70] Krum, Spencer; van Havelingen, William; Karo, Ben; Turnbull, James; McCune, Jeffrey. Pro Puppet. Second Edition. New York: Springer Science + Business Media. 2013. E-Book. p.1-3.
[71] Krum, Spencer; van Havelingen, William; Karo, Ben; Turnbull, James; McCune, Jeffrey. Pro Puppet. Second Edition. p.1-3.
[72] Puppet Labs. Enterprice vs. Open Source. 2014. http://puppetlabs.com/puppet/enterprise-vs-open-source (Retrieved 2014-04-28)
[73] http://docs.puppetlabs.com/pe/latest/cloudprovisioner_vmware.html (2014-04-28)
[74] Puppet Labs. Docs: PE 3.2 Orchestration Controlling Puppet. 2014. http://docs.puppetlabs.com/pe/latest/orchestration_puppet.html (Retrieved 2014-04-28)

inspect and compare the different resources that exist on different nodes in the system.[75]

Puppet Enterprise offers the possibility to get support and maintenance for the managed system. How high the cost will be depends on the level of support required.[76] Puppet Enterprise can be used for free for up to 10 nodes with no support. The cost for 100 Puppet Enterprise nodes with standard support is $10,500 annually.[77]

Puppet offers documentation on their website.[78] There is also an active community available where it is possible to ask questions or discuss different Puppet related subjects.[79] They also offer IRC channels and a Q&A website where it is possible to find different questions with answers.[80]

Comparison study

Puppet is the most established system in the comparison study and therefore the CMS that its competitors oftentimes are compared to. And according to many it is a very capable system. According to Paul Venezia, who compares Puppet, Chef, Ansible and Salt in an article from Info World made available on November 21 2013, Puppet provides the most complete solution in terms of user interfaces, actions and modules.[81] A relatively simple installation process and a straightforward command line interface also impress him. However, he points out that it is highly recommended to have a solid background with Ruby. He also claims that the configuration process can be very time consuming and that both Ansible and Salt are easier to use.

Steve Hall compared Puppet with Ansible in an article published on ScriptRock.[82] He agrees with Venezia about Puppet having the most mature interface and a more developed GUI. He adds that Puppet has better platform support than Ansible. However, he claims that Puppet has become too large for its own best, which leads to it not being agile. He writes that implementations of feature requests and bug fixes take too long.

---

[75] Puppet Labs. Docs: PE 3.2 Orchestration Browsing Resources. 2014. http://docs.puppetlabs.com/pe/latest/orchestration_resources.html (Retrieved 2014-04-28)
[76] Puppet Labs. Support Plans. 2014. http://puppetlabs.com/services/support-plans (Retrieved 2014-04-28)
[77] Puppet Labs. How to Buy. 2014. http://puppetlabs.com/puppet/how-to-buy (Retrieved 2014-04-28)
[78] Puppet Labs. Docs: Puppet Labs Documentation. 2014. http://docs.puppetlabs.com (Retrieved 2014-04-24)
[79] Puppet Labs. Community. 2014. http://puppetlabs.com/community/overview (Retrieved 2014-04-24)
[80] Puppet Labs. Get Help. 2014. http://puppetlabs.com/community/get-help (Retrieved 2014-04-24)
[81] http://news.idg.no/cw/art.cfm?id=D21968B0-EE1C-1249-85D672B34C0DA5BB (Retrieved 2014-05-09)
[82] Hall, Steve. ScriptRock. Ansible vs Puppet. 2014-02-27. https://www.scriptrock.com/articles/ansible-puppet/ (Retrieved 2014-05-09)

In another article from ScriptRock, Mike Baukes compares Puppet to Salt.[83] He is also of the opinion that Puppet is a more proven solution with a good GUI, but claims that SaltStack is quicker to listen to and help the community. This was a negative aspect of Puppet that Hall also pointed out when comparing Puppet to Ansible in his article. Baukes mentions that some think that Puppet forces users to pay for Puppet Enterprise by providing important features only to enterprise users, which is very different from Salt's approach where all features are available in the open source version. He also claims that Salt is easier to use since it is based on Python and that it is more scalable because of its several masters feature.

In a comparison between Puppet, Chef and Ansible posted on Mark Phillips' blog, he writes that Ansible is his preferred CMS.[84] He finds it easier to use than its competitors, while still providing all functionality that he needs. He claims that Puppet works very well, except for it being slow when used with many clients. He can see no reason to use Chef, since he found it to be a more complex version of Puppet without Puppet's DSL, which he prefers to Chef's Ruby approach. He claims that even for Ruby programmers Puppet is a better alternative.

Luke Kanies has posted his reasons for choosing a DSL for Puppet on Puppet's webpage.[85] He points out that it makes the language easier to use and that it forces users to use the software in the intended way. The downside is that it limits what is possible to do with the language. It also forces users to learn a new unique language.

On their webpage the team behind Puppet explain that they support dependencies, which means that it is possible to state how different resources depend on each other. Then Puppet can figure out in which order to execute the scripts.[86] Chef has a different approach, as can be read on their webpage. Dependencies are not supported, but instead it is guaranteed that the clients apply the resources in the same order every time.[87]

Whether Puppet or Chef is preferable of course depends on the user's preferences. Jacques Chester writes on his blog on June 27 2012 that he prefers Puppet's approach to both CFEngine's and Chef's solutions.[88] He found that the possibility to write dependencies with Puppet made his work much easier, while Chef instead pushed a lot of the difficult work onto him. He also claims that Puppet in a better way that its competition models how real systems work. However, Nathen Harvey

---

[83] Baukes, Mike. ScriptRock. Puppet vs. Salt. 2013-08-13. https://www.scriptrock.com/articles/puppet-vs-salt/ (Retrieved 2014-05-09)
[84] Phillips, Mark. Probably. Puppet vs Chef vs Ansible. 2013-09-01. http://probably.co.uk/puppet-vs-chef-vs-ansible.html (Retrieved 2014-05-09)
[85] Kanies, Luke. Puppet Labs. Why Puppet has its own configuration language. 2012-08-02. http://puppetlabs.com/blog/why-puppet-has-its-own-configuration-language (Retrieved 2014-05-09)
[86] Puppet Labs. Docs: Learning Puppet – Resource Ordering. 2014. http://docs.puppetlabs.com/learning/ordering.html (Retrieved 2014-05-09)
[87] Chef. Chef Documents Why Chef?. http://docs.opscode.com/chef_why.html (Retrieved 2014-05-09)
[88] Chester, Jacques. Journal de Jacques. A not-so-brief aside on reigning in chaos. 2012-06-27. http://chester.id.au/2012/06/27/a-not-sobrief-aside-on-reigning-in-chaos/ (Retrieved 2014-05-09)

writes on his blog on November 21 2011 that CustomInk liked Chef better than Puppet.[89] Since they were Ruby developers they preferred Chef's language to Puppet's. He also mentions that they valued Chef's order matters philosophy above Puppet's dependency management, which they found to be tedious.

In an article from ScriptRock posted on May 16 2013, Mike Baukes compares Puppet to CFEngine.[90] He states that CFEngine is faster, user less memory and has fewer dependencies than Puppet. He does however, point out that CFEngine is known to be very hard to learn and master. He also mentions that Puppet can take responsibility for dependency management and that it is easier to learn than CFEngine for users with little experience of programming.

Salt

General study

In 2011[91] Marc Chenn and Thomas Hatch founded SaltStack, the company behind the CMS Salt.[92] Salt has since then been available under the Apache 2.0 license.[93] It is still developed and the latest release was made available on April 15 2014.[94] Salt supports the commonly used platforms, such as the common Linux distributions, Mac OSX, Windows and Unix. The master cannot be a Windows server though, but the clients, called minions, have full support for Windows.[95]

Salt is divided into two main functionalities. The first part is a configuration management tool capable of maintaining defined states in remote nodes. This means for example to make sure that a certain package is installed on a node. The other main part of Salt is a remote execution system, which makes it possible to execute commands and query data from remote nodes.[96]

Salt uses a master-agent model, or a master-minion topology as they call it themselves. A master server is the central control unit for the minions.[97] In order to use the system, the software has to be installed on both the master and the

[89] Harvey, Nathen. Nathen Harvey's blog. Why We Chose Chef Over Puppet at CustomInk. 2011-11-21. http://www.nathenharvey.com/blog/2011/11/21/why-we-chose-chef-over-puppet-at-customink (Retrieved 2014-05-09)
[90] Baukes, Mike. ScriptRock. Puppet vs CFEngine. 2013-05-16. https://www.scriptrock.com/blog/puppet-vs-cfengine/ (Retrieved 2014-05-09)
[91] LinkedIn. SaltStack. 2014. http://www.linkedin.com/company/salt-stack-inc?trk=prof-following-company-logo (Retrieved 2014-04-23)
[92] SaltStack. About SaltStack. 2014. http://www.saltstack.com/about/ (Retrieved 2014-04-23)
[93] SaltStack. Salt Documentation version 2014.1.3. p.2. http://docs.saltstack.com/en/latest/ (Retrieved 2014-04-23)
[94] SaltStack. Salt 2014.1.3 Release Notes. 2014. http://docs.saltstack.com/en/latest/topics/releases/2014.1.3.html# (Retrieved 2014-04-25)
[95] SaltStack. Salt Documentation version 2014.1.3. p.5-21. http://docs.saltstack.com/en/latest/ (Retrieved 2014-04-23)
[96] SaltStack. Salt Documentation version 2014.1.3. p.1. http://docs.saltstack.com/en/latest/ (Retrieved 2014-04-23)
[97] SaltStack. SaltStack Walk-through. 2014. http://docs.saltstack.com/en/latest/topics/tutorials/walkthrough.html (Retrieved 2014-04-23)

minions.[98] It is also possible to use Salt with a standalone model, and then it is enough to install the software on just the minions.[99] The minions can be configured both by the master pushing changes to the minions or by the minions pulling information from the masters.[100] Salt also allows a multiple master configuration where redundant masters have control of the minions within the system, which increases the scalability of the system.[101] Salt is written in Python and uses SLS (SaLt State Files) for configuration management. SLS describes the desired state of the system. By default the data is represented in the YAML format.[102] The main languages used within Salt are Python and a domain specific version of Python called pyDSL.[103] The communication layer within Salt is built on zeroMQ.[104]

Salt is completely open source and all features are available for open source users. SaltStack offers a program for companies called SaltStack Enterprise that offers support, training sessions and consultants. The price is set individually for each costumer.[105] The list price for 100 SaltStack Enterprise nodes is $15,000 annually.[106] There is a GUI called Halite being developed for Salt. It is available on Github and is available to anyone, but is currently only available in a pre-alpha version.

SaltStack has had some momentum during the last years. They were included in GitHub's Octoverse list in 2013 as one of the largest open source projects in the world. They have also been awarded for their software Salt by a few technical magazines.[107] Their customers include for example LinkedIn, Orange, Apple, Harvard University and hulu.[108]

Comparison study

Salt has been favoured for being easier to learn and use than older CMS like Chef and Puppet. Corey Quinn, who is a long term SaltStack developer, has expressed this opinion. He wrote about Salt in a blog post, where he praised the simplicity of the system. However, Luke Kanies, the founder of Puppet, pointed out in the first comment for the blog post that the simplicity in Salt results in it not being able to

---

[98] SaltStack. Salt Documentation version 2014.1.3. p.5. http://docs.saltstack.com/en/latest/ (Retrieved 2014-04-23)
[99] SaltStack. Salt Documentation version 2014.1.3. p.23. http://docs.saltstack.com/en/latest/ (Retrieved 2014-04-23)
[100] SaltStack. SaltStack Enterprise. 2014. http://www.saltstack.com/enterprise/ (Retrieved 2014-05-05)
[101] SaltStack. Multi Master Tutorial. 2014. http://docs.saltstack.com/en/latest/topics/tutorials/multimaster.html (Retrieved 2014-05-05)
[102] SaltStack. Salt Documentation version 2014.1.3. p.35-36. http://docs.saltstack.com/en/latest/ (Retrieved 2014-04-23)
[103] SaltStack. Salt Renderers pyDSL. 2014. http://docs.saltstack.com/en/latest/ref/renderers/all/salt.renderers.pydsl.html (2014-05-05)
[104] S Hatch, Thomas. ZeroMQ. Salt. 2011-04-17. http://zeromq.org/story:4 (Retrieved 2014-05-05)
[105] SaltStack. SaltStack Enterprise. 2014. http://www.saltstack.com/enterprise/ (Retrieved 2014-04-23)
[106] Private conversation with Tyler Kirkham at SaltStack.
[107] SaltStack. SaltStack awards & accoledes. 2014. http://www.saltstack.com/awards/ (Retrieved 2014-04-23)
[108] SaltStack. SaltStack. 2014. http://www.saltstack.com (Retrieved 2014-04-23)

provide the same functionality as Puppet.[109] Paul Venezia stated that he shares this opinion in an article at Info World where he compared Chef, Puppet, Salt and Ansible. [110] He points out that Salt and Ansible are more administration oriented, while Chef and Puppet are better in development scenarios. He claims that Puppet is the most mature solution available, but writes that an advantage with Salt is its high scalability because of the possibility to have multiple masters.

Ben Hosmer wrote an article about Salt for the Linux Journal. [111] He mentions that Salt is easier to use that other CMS and that it is a fast system because of the communication by zeroMQ. Furthermore he claims that an advantage with Salt is that it is entirely open source and that no additional Salt features are locked behind a payment wall.

Stephen Wood, who is a system engineer at a company called MOZ,[112] wrote a post about Salt on his blog in October 2013. He praises Salt's high scalability and adds that it comes from Salt's use of zeroMQ for sending messages, which requires less from the master than pure SSH. This means that it is possible for masters to manage very many minions.[113]

Salt is often compared to Ansible since they both are recently released CMS with a lightweight approach. The main difference between Salt and Ansible is that Salt uses agents on the client servers. Some have argued that Ansible's agent-less approach is more secure. Steve Hall is of this opinion, as he wrote in a comparison between Salt and Ansible on Scriptrock on January 10 2014.[114] Hall also mentions that a negative aspect of Salt is that it forces users to learn Python or PyDSL.

Jens Rantil, a Swedish software engineer at Think AB, wrote about the differences between Salt and Ansible on his blog on March 17 2014. He draws the same conclusion as Hall about the security of Ansible and Salt. He says that Ansible is safer, but adds that it should not be very hard to maintain a secure environment with Salt because of its simple architecture. Jens also praises the scalability of Salt and points out that the Salt minions give Salt an advantage in a cloud environment, since each new cloud instance can automatically become a minion.[115]

---

[109] Quinn, Corey. Smartbear. A Taste of Salt: Like Puppet, But Less Frustrating. 2013-04-25. http://blog.smartbear.com/devops/a-taste-of-salt-like-puppet-except-it-doesnt-suck/ (Retrieved 2014-05-05)
[110] Venezia, Paul. Review: Puppet vs. Chef vs. Ansible vs. Salt. *InfoWorld*. 2013-11-21. http://www.infoworld.com/d/data-center/review-puppet-vs-chef-vs-ansible-vs-salt-231308 (Retrieved 2014-05-05)
[111] Hosmer, Ben. Getting Started with Salt Stack-the Other CMS Built with Python. *Linux Journal*. 2013-01-09. http://www.linuxjournal.com/content/getting-started-salt-stack-other-configuration-management-system-built-python (Retrieved 2013-05-05)
[112] Moz. Stephen Wood. 2014. http://moz.com/about/team/stephen (Retrieved 2014-05-05)
[113] Wood, Stephen. Hey Stephen Wood. Why SaltStack is Better. 2013-10-09. http://www.heystephenwood.com/2013/10/why-saltstack-is-better.html
[114] Hall, Steve. ScriptRock. Ansible vs. Salt. 2014-01-10. https://www.scriptrock.com/articles/ansible-vs-salt/ (Retrieved 2014-05-05)
[115] Rantil, Jens. Jens Rantil's Blog. Salt vs. Ansible. 2014-03-17. http://jensrantil.github.io/salt-vs-ansible.html (Retrieved 2014-05-15)

Discussion

This section contains information about why we found these CMS to be among the top five ones in the general study. It also contains a summary of the results from the comparison study and opinions of how the top five CMS compares to each other. A summary of the results from the general study can be seen in Table 1 and a summary of the results from the comparison study can be seen in Table 2. The section Discussion for the CMS that were not included in the comparison study can be found in the appendix.

This section also contains our reflections about the trends in the CMS industry, problems that were encountered during the study, thought about the method used for this project and recommendations for further studies.

Ansible

Ansible's approach to configuration management is a bit different compared to older systems such as CFEngine or Puppet. It uses a push model instead of a pull model and does not use daemons. Ansible tries to make the configuration tasks simpler and has in its relatively short time on the market already got a lot of attention. It is quite different from the other products in our in-depth study and has relatively good platform support, even if it does not support Windows. It offers commercial support, has an active community, a few big customers and is under development with new releases coming regularly. That is why we decided to include Ansible in the comparison study.

The comparison study shows that many seem to be very pleased with Ansible's simpler architecture and flat learning curve. However, this appears to be one of its weaknesses as well, since the simple model might not translate as well for more development heavy or very large systems. Ansible's GUI is also a few steps behind the solutions from older CMS, such as Puppet. Since the software is newer than its competitors it is not as proven, which might be a problem for some users.

Ansible provides a simple system with minimal dependencies, a flat learning curve and no daemons. However, it does not have much experience in the business compared to for example Puppet. It also lacks a well-developed GUI and focus on large and complex systems. These aspects make Ansible a good choice for some users.

CFEngine

CFEngine is one of the oldest alternatives on the software configuration management market and is also one of the most interesting. It has a big community, lots of documentation, many big companies using it and offers commercial support. All this, together with the fact that it does the configuration management tasks in an efficient way, makes us want to do a more in-depth study on CFEngine.

The comparison study of CFEngine shows that it is a powerful and efficient tool. It is one of the oldest, perhaps even the oldest, CMS available on the market, but it has managed to evolve together with the industry. Being C-based and decentralized, it is very fast and efficient. Some claim that it is hard to learn and master though, and compared to newer alternatives such as Ansible and Salt, it might not be worth the extra effort, especially for smaller solutions. During the comparison study it was easier to find comparisons of the other CMS. The ones we found about CFEngine were often somewhat dated. This indicates that the new users are focusing more on other alternatives.

This makes us draw the conclusion that while CFEngine is a robust and fast CMS with a lot of functionality, it is not as easy to learn and use as some of the new CMS, such as Ansible and Salt.

Chef

Chef is a well-known actor in the configuration management world. It is a mature system that is used by many large companies. Chef has an active community and an enterprise version that provides commercial support and some additional features. Therefore we chose to make a comparison study on Chef.

Chef systems are configured in Ruby, which makes it easy to learn for people with experience in Ruby programming. It also makes it powerful since the whole Ruby language is made available for the users. However, this can make it difficult to learn and use for people without a lot of programming experience. Our comparison study shows that many think that Chef has a steep learning curve. The advantage with the language used in Chef compared to Puppet's DSL is that it gives experienced programmers used to Ruby more flexibility. Compared to Salt and Ansible, Chef is a more mature system with more features and better platform support.

In conclusion Chef is a powerful CMS because of its use of Ruby. It might therefore appeal to developers or users with a background of Ruby programming. However, this approach makes it hard to understand and learn for users without programming experience.

Puppet

Puppet is one of the most established systems in the CMS market and offers very good platform support along with a mature solution. It has lots of active users, a company behind it providing support, an enterprise program and lots of well-known companies and institutions as customers. This, together with the fact that Puppet is one of the most popular and well-known CMS available, made us include it in the comparison study.

The comparison study shows that Puppet offers a very capable solution. Several claim its GUI provided in the enterprise version to be well developed. It has been praised for its dependency handling and easy to use DSL. However, it has been criticised for being hard to configure and for becoming slow when used in the client-server mode with many clients. Some argue that Ansible and Salt are easier to learn

and that their Python-based architecture is preferable to Chef and Puppet's Ruby-based software. A quality of Puppet is that it can provide a more mature and proven solution compared to newer software such as Ansible and Salt. However, some claim that the upcoming competitors are faster to implement requested features and bug fixes. Compared to CFEngine Puppet is easier to learn and use, but it uses more memory and requires more dependencies because of its Ruby base and runs slower.

In conclusion Puppet provides a modern and capable solution with lots of functionality. It is easier to learn and understand than CFEngine, but more complex than Salt and Ansible. The biggest difference from Chef is that Puppet offers a DSL and dependency management.

Salt

Salt is a rather new CMS that has had some success during the last few years and is now used by well known companies and institutions. Salt has a more lightweight approach than for example Chef and Puppet, but still uses the master-agent architecture that Ansible does not. This makes Salt an interesting alternative somewhere in between the older CMS and Ansible. Salt provides support for the most common platforms, offers commercial support and has an active community. Therefore we decided to include it in the comparison study.

The main opinion from the comparison study is that Salt is more lightweight than Chef, Puppet and CFEngine and also easier to learn. The multiple master support and fast zeroMQ communication makes it very scalable. On the other hand it has been expressed that the simplicity restrict the system from providing the same functionality as the older alternatives. Another concern with Salt is the GUI, which is not as developed as the solutions offered by Puppet and Chef. The master-agent architecture makes Salt more scalable than Ansible. However, it has been said that Ansible's architecture is more secure than Salt's.

In conclusion Salt is a modern CMS that is well suited for a cloud environment. Salt is easy to learn and use and its main advantage compared to the other systems in the comparison study is the scalability of the system.

Trends in the industry

CMS have been available since 1993 and of course a lot has happened in the industry since then. This section gives a picture of different trends in the industry over the years and how today's and perhaps even tomorrow's development climate differ from the early years.

A trend that we have discovered during this study is that the CMS that have been released in the latest few years try to beat the older systems not in raw features, but in simplicity. The comparison study in this report shows that CFEngine, Chef and Puppet all are very capable tools with a lot of functionality, but that the newer alternatives Salt and Ansible seem to be easier to understand. The simplicity is a very important parameter to consider; after all one of the main purposes of using a

CMS is to make the configuration tasks easier. It becomes problematic when the configuration of the CMS takes a lot of time and effort to get right. Or as Michael DeHaan wrote in a blog post called “The Origins of Ansible” published on December 8 2013: “As a developer myself, I wanted to write development code, I didn't want to spend 50% of my time fighting with the automation tooling and have the automation itself be a source of frustration”.[116] Cdist is another example of this trend towards simplicity, since it just as Ansible tries to make the configuration management as simple as possible. They both exemplify that the newer systems focuses on simplicity instead of new features.

Something that we discovered while doing the study was that the development of the CMS has been transferred from universities to companies. LCFG, Radmind and CFEngine are all old CMS that were first developed at Universities. CFEngine later transferred its development to a company in order to be able to expand the product and provide additional features, such as support and professional help. It is also apparent that no CMS released during the latest few years has come from universities, meaning that the new solutions are coming from outside of the academic world.

This shift can probably be explained with the fact that the audience for these products have changed quite dramatically over the years. When the first version of CFEngine was developed in 1993, the only ones that would have use for CMS were universities and big companies, since they were practically the only ones that used big and complex computer systems. But today, with virtualisation, virtual machines and lots of people using very big systems with a great number of instances, the audience for these products is much bigger than it was when the first CMS were released.

This means that there now is room for several solutions on the market and that even people that are not involved with big projects on universities or large companies need to deal with configuration management. The latest CMS, such as Ansible and Salt, seem to come from one or a few persons that start open source projects, often available on GitHub, and then, if they get enough attention, establish companies offering additional features and support. This shows a big difference between the newer and the older systems.

Problems during the study, discussion of method and recommendations for further studies

Before the study we did not have any experience or knowledge of configuration management. Therefore one of the most difficult tasks of the study was to define the parameters that the CMS should be evaluated from, since we did not know which parameters that were important before we started the project.

Since we wanted to make a recommendation of the most interesting CMS for a company, we needed to include opinions from people that have used the CMS.

[116] DeHaan, Michael. Ansible Blog. The Origins of Ansible. 2013-12-08. http://www.ansible.com/blog/2013/12/08/the-origins-of-ansible (Retrieved 2014-05-13)

Since we did not have any experience of the CMS ourselves, we had to collect information from articles, on forums and in blog posts. We tried to use as neutral sources as possible and to include opinions that have been phrased by more than one person. However, there is no guarantee that these people's opinions reflect reality. Another problem with using webpages as sources is that there is no guarantee that the information is stored for a long time. Links can quickly become invalid if the developers change the webpages for their products. It might have been better to interview people with experience of working with the systems, but there was no time for us to make a selection. The audience for CMS is rather small and it would have been hard to find people with experience of several CMS.

Another issue with our study is that the CMS are constantly updated and changed. A lack of functionality or a problem described in an article might have changed since the article was written. This holds true even for very new articles, since the updates are so frequent. In order to avoid transferring these errors to our report we checked if the information of the articles were up to date. However, even if the information was correct by the time of writing, updates will be made available to the CMS after this report has been published. Then it can no longer be guaranteed that the information in the report is up to date.

Since we have not been able to use the CMS ourselves, it would be interesting to extend the study by using the systems for realistic scenarios. By doing this in a future study it would be possible to evaluate the conclusions drawn in this study for a realistic experiment. We did not have time to do an in-depth study of the security aspects of the different CMS. That is something that would be interesting to include in a future study.

Conclusion

Only the five most interesting CMS from the original twenty in the general study were included in the comparison study. These are all good alternatives that provide the functionality we are looking for, but their approaches differ a bit from each other. Salt and Ansible are more lightweight systems that are written in Python. They are both easy to learn and use while Chef, Puppet and CFEngine are more complex systems. Chef and Puppet are written in Ruby while CFEngine is C-based.

If a complex system with a lot of functionalities should be implemented CFEngine, Puppet and Chef are the most interesting candidates. CFEngine is the most efficient system, but it requires some time and effort to learn and understand. If the user is familiar with Ruby programming, Chef or Puppet might be a better choice. Chef can provide a bit more flexibility than Puppet, but on the other hand Chef's pure Ruby can be hard to get into for people without Ruby experience. Puppet provides a DSL that is logical and easy to learn and might therefore be more interesting for some users. Both Ansible and Salt are easier to use and should be chosen if the most important thing is to be able to administer a system in a simple way. In contrast to Salt Ansible does not use any agents, which makes it quicker and easier to install.

Another difference is that Ansible's GUI is more developed. However, Salt is more scalable and supports Windows.

We have come to the conclusion that it is hard to make a recommendation for a cloud provider to support only one CMS for all types of users, since the CMS all have different approaches. Therefore we think that the best solution, if it is possible, is to provide support for more than only one system, so that solutions for several types of customers can be provided.

If it is not possible to support all five systems we would recommend including at least one of the more lightweight systems and one of the more complex systems. Among the more complex ones we do believe that Puppet generally is the best choice. This is because of it providing a very capable solution with a mature GUI and a language that it is easier to learn and understand than Chef's and CFEngine's. Among the lightweight ones we recommend Salt, since it is more scalable than Ansible. It also has Windows support, which makes it useful in more situations. If only one system can be chosen we would recommend Puppet since it is the most comprehensive system and can be used in the widest range of situations, without being too hard to learn and master.

Appendix

The information about the 15 CMS that were not included in the comparison study can be found in this appendix. The information gathered during the study can be found under section Results and the information of why they were not included in the comparison study can be found under section Discussion. A summary of the information from the general study can be found in Table 1.

Results

This section contains the information from the general study for the 15 different CMS that were not among the 5 most interesting ones. The CMS presented in this section can be divided into two categories. CMS not relevant to our scenario belong to the first category. This includes Isconf, Juju, OCS Inventory, PIKT, Radmind, Rundeck and STAF. The second category includes CMS that are relevant but not as good as other alternatives in the comparison study. This includes BCFG2, cdist, LCFG, OPSI, Quattor, Rex, Spacewalk/Satellite and SmartFrog. A summary of the results from the general study can be found in Table 1.

BCFG2

BCFG2, pronounced bee-config, is a CMS that was developed at the department of mathematics and computer science at Argonne National Laboratory when their experts became tired of configuring their many computer systems manually. BCFG2 has support for openSUSE, Fedora, Gentoo and Debian. Their many derivatives are also supported and through the use of the developer's distribution independent Encap packages FreeBSD, AIX, Solaris and Mac OSX. This means that the Linux and Unix support is good. Windows is not supported though.[117] BCFG2 was first released on August 12 2004. [118] It is still being developed and updated with its latest release made available on February 25 2014.[119]

BCFG2 is written in python and is released under the BSD-license. The software is based on an operational model, where the specifications can be used to change and validate the configuration of several clients. BCFG2 uses a client-server model, where the descriptions of the clients are stored and managed server-side. In order to set up the system the server has to be installed, the BCFG2 client package has to be installed on the clients and a configuration file has to be stored on each client. BCFG2 uses SSL (a protocol for secure connections over the Internet) for the connections. The system can be set up so that the clients request updates from the server at regular intervals, or you can have the server actively contact the clients.[120] BCFG2 has a feature that in an objective way valuates how well the configuration of the clients has been set up. The specifications use the XML format.

---

[117] Jung, Marco. Magnus, Nils. The Dean p. 30. *Linux Magazine*. Issue 101. (2009) http://www.linux-magazine.com/Issues/2009/101/Bcfg2 (Retrieved 2014-05-25)
[118] BCFG2. Wiki.2014. https://trac.mcs.anl.gov/projects/bcfg2/wiki/Trivia (Retrieved 2014-04-22)
[119] BCFG". Download. 2014. http://bcfg2.org/download/ (Retrieved 2014-04-22)
[120] Jung, Marco. Magnus, Nils. The Dean p. 30-31. *Linux Magazine*. Issue 101. (2009) http://www.linux-magazine.com/Issues/2009/101/Bcfg2 (Retrieved 2014-05-25)

There is no information about companies using BCFG on their official webpage.[121] No professional support is offered on BCFG2's official website, but there is documentation, FAQ:s and other kinds of help pages available. There is also an IRC chat channel where it is possible to discuss different topics with the community. A mailing list is available, but there is no information about whether it is still active or not.[122]

Cdist

Released in 2011[123], cdist is a relatively new CMS created by Nico Schottelius. The software is compatible with most Linux- and Unix-distributions, including for example Mac OSX.[124] Cdist is written in Python and does not use a domain specific language (DSL); actually the language that is used with cdist is shell script, which should be well known to UNIX system engineers. The software is still being developed and the latest release was made available on Github on April 12 2014.[125]

The software does not require agents or a high level programming language in order to be able to function on the hosts. It is required that the hosts have a posix compatible shell and a SSH (a protocol for secure connections over the Internet) server running, but that is all that has to be set up. A push-based model is used, which makes it easy to set up the system. Cdist is written in order to keep the configuration management as simple as possible, to avoid unnecessary complexity that makes it hard to understand the system.[126]

It has been mentioned on Twitter as an upcoming CMS that might be able to compete with more established actors such as Chef and Puppet.[127] Cdist is an active project that is developed on Github.[128] It is possible to get commercial support for cdist from Nico Shottelius' own company.[129] There is no information on the official website about any businesses using cdist in their environment and there is no community page available on the official webpage. However, there is an active mailing list and an IRC channel available. [130]

---

[121] BCFG2. http://bcfg2.org (Retrieved 2014-04-22)
[122] BCFG2. Help.2014 http://docs.bcfg2.org/help/index.html (Retrieved 2014-04-23)
[123] Nico Schottelius. cdist, next generation configuration management p5. *Linux Erfa*. 2013. http://www.nico.schottelius.org/software/cdist/speeches/2013-11-22_eth_linux_erfa.pdf (Retrieved 2014-04-22)
[124] Nico Schottelius. cdist. Supported operating systems. 2012. http://www.nico.schottelius.org/software/cdist/os/ (Retrieved 2014-04-22)
[125] Github. cdist. https://github.com/telmich/cdist/releases (Retrieved 2014-04-25)
[126] Nico Schottelius. Why should I use cdist?. 2013 http://www.nico.schottelius.org/software/cdist/why/ (Retrieved 2014-04-23)
[127] Topsy. cdist. 2014. http://topsy.com/trackback?url=http%3A//www.nico.schottelius.org/software/cdist/ (Retrieved 2014-04-23)
[128] Github. cdist. 2014. https://github.com/telmich/cdist (Retrieved 2014-04-23)
[129] Nico Schottelius. cdist. Support. http://www.nico.schottelius.org/software/cdist/support/ (Retrieved 2014-04-23)
[130] Nico Schottelius. cdist- usable configuration management. 2013 http://www.nico.schottelius.org/software/cdist/ (Retrieved 2014-02-23)

Isconf

Isconf was developed as a standard toolset for managing and assembling large number of machines to simplify good IT infrastructures.[131] The main idea behind Isconf is to have an infrastructure where all the servers are managed in a uniform way by images provided by an install server.[132] There is no commercial support or forum available where it is possible to discuss problems or ask questions, nor is there any information about any companies currently using the software on their webpage.[133] On sourceforge there is a mailing list for Isconf users, but nothing has been updated there since 2010.[134]

In Isconf you define the order of installation for the servers and it will be deployed in that specific order that is controlled by daemons installed on each machine. If something fails the whole operation will be restored to the state before the error. Isconf update is preformed while booting the machines and the whole system has to be rebooted in order too make software updates.[135] Isconf is written in Python and is distributed under a GPL license.[136] Isconf is constructed in order to run a large number of UNIX machines, but there is no information of whether it runs on other operating systems.[137]

The latest Isconf release was in 2006.[138] This system can do part of what modern CMS are able to do, but lack functions required to run advanced applications due to complicated update routines. This is according to Luke Kaines who was involved in the Isconf project and later founded the popular CMS Puppet.[139]

Juju

Juju is an orchestration management tool that automatically deploys and connects different servers for applications.[140] It was launched on April 3 2013. Juju manages Ubuntu servers and is developed by Canonical Ltd, the company behind Ubuntu.[141] Juju is configured by charms, which are bundles of files that define the relationships between the servers. The charms can be written in any programing language that

---

[131] Traugott, Steve. Huddleston, Joel. Traugott, Joyce Cao. 2007. http://www.infrastructures.org/ (Retrieved 2014-04-16)
[132] Traugott, Steve. Huddleston, Joel. Traugott, Joyce Cao. 2007. http://www.infrastructures.org/bootstrap/hostinstall.shtml (Retrieved 2014-04-16)
[133] Traugott, Steve. 2014. http://www.isconf.org/ (Retrieved 2014-04-16)
[134] Sourceforge. Isconf maling list. 2010. http://sourceforge.net/p/isconf/mailman/isconf-devel/ (Retrieved 2012-04-16)
[135] Kanies,Luke. ISconf: Theory, Practice, and Beyond. *LISA '03 Paper.* San Diego. 2003 https://www.usenix.org/legacy/publications/library/proceedings/lisa03/tech/full_papers/kanies/kanies_html/index.html (Retrieved 2014-04-16)
[136] Ohloh. 2014. http://www.ohloh.net/p/11758 (Retrieved 2014-04-16)
[137] Traugott, Steve. Huddleston, Joel. Traugott, Joyce Cao. 2007. http://www.infrastructures.org/bootstrap/introduction.shtml (Retrieved 2014-04-25)
[138] Isconf. Browser. http://trac.t7a.org/isconf/browser/branches (Retrieved 2014-04-15)
[139] Kanies,Luke. ISconf: Theory, Practice, and Beyond. *LISA '03 Paper.* San Diego. 2003. https://www.usenix.org/legacy/publications/library/proceedings/lisa03/tech/full_papers/kanies/kanies_html/index.html (Retrieved 2014-04-16)
[140] Canonical limited. Juju Features. 2014. https://juju.ubuntu.com/features/ (Retrieved 2014-04-17)
[141] Canonical limited. Juju Install. 2014. https://juju.ubuntu.com/install/ (Retrieved 2014-04-14)

runs on Ubuntu. CMS, such as Chef and Puppet, can be used inside the charms for features like deployment and software update. The charms define the relation between servers. Juju is currently bound to Ubuntu servers.[142] The system is controlled from a machine where the platform support is wider. It can be used on Windows, Mac OSX and Ubuntu.[143]

The latest juju update was released on June 11 2014.[144] It is possible to get help with juju both from the community and commercial support by Canonical.[145] There is also an active forum available.[146] However, there is no information about any companies currently using juju.[147]

LCFG

The development of LCFG, which stands for Local ConFiGuration system, started during 1993 at the Department of Computer Science at Edinburgh University. It was developed to automatically configure and maintain a large number of UNIX systems. A new version of LCFG called LCFG(ng) has been released, where the core has been redesigned and many component have been rewritten. Officially the platform support is limited to Solaris, Fedora and Scientific Linux. This means that some Linux- and Unix-distributions are supported, but there is no support for Mac OSX and Windows.[148] Documentation can be found on the homepage, but no support or training sessions are provided.[149] There is no information about any companies currently using LCFG on their webpage and there is no forum available.[150] There is however an active mailing list.[151] The software is still under development and new versions are released each week.[152]

LCFG uses a master server. This is where the configurations for all the nodes are stored as source files, which should be written in the Perl language.[153] The compiler then compiles the source files into a single XML profile for each node. The nodes are notified if something has changed in their profiles on the server. Agents on the nodes need to be installed. They use different software called component to remove or install software packages specified by the server configuration. They can also ask the server for configuration synchronisation regularly, if they should happen to miss

---

[142] Canonical limited. The top 12 questions about Juju. 2013. http://insights.ubuntu.com/resources/factsheets/top-12-questions-about-juju/ (Retrieved 2014-04-14)
[143] Canonical limited. Juju Install. 2014. https://juju.ubuntu.com/install/ (Retrieved 2014-04-14)
[144] Launchpad. Juju-core. 2014 https://launchpad.net/juju-core/+milestone/1.18.1 (Retrieved 2014-05-12)
[145] Canonical limited. Support. 2014. http://www.ubuntu.com/support (Retrieved 2014-05-15)
[146] Ubuntu. Forum Juju and coud. 2014. http://ubuntuforums.org/forumdisplay.php?f=392 (Retrieved 2014-04-25)
[147]Canonical limited. Juju. 2014. https://juju.ubuntu.com (Retrieved 2014-04-25)
[148] Lcfg. faq. 2007. http://www.lcfg.org/faq/ (Retrieved 2014-04-16)
[149] Lcfg. doc. 2011. http://www.lcfg.org/doc/ (Retrieved 2014-04-17)
[150] Lcfg. http://www.lcfg.org (Retrieved 2014-04-25)
[151] Lcfg. Lcfg discussion list. http://lists.inf.ed.ac.uk/mailman/listinfo/lcfg-discuss (Retrieved 2014-04-25)
[152] Lcfg. Download. 2007 http://www.lcfg.org/download/ (Retrieved 2014-04-25)
[153] Anderson, Paul. The complete guide to LCFG p.23. 2006. http://www.lcfg.org/doc/guide.pdf (Retrieved 2014-04-17)

the initial message.[154] It is also possible to install new machines automatically by creating source files on the server for the new systems and then booting from a network system.[155]

LCFG is available under the GNU GPL version 2 license.[156] A subset of LCFG was used for the European Data Grid project, but they later changed to Quattor for the configuration management.[157] The goal of the project was to build a better computing environment for intensive computations and analysis of shared databases. One of the contractors was Cern.[158]

OCS Inventory

OCS Inventory NG, which stands for Open Computers and Software Inventory Next Generation, is French software that was first released in 2001 by FactorFX. It is still being updated and supported. The latest release was made available on February 6 2014.[159] The software is released under the GNU General Public License version 2.[160] OSC Inventory's primary function is to collect information about the hardware and the software in a network. Agents are installed on the clients that are to be monitored and then the collected information can be visualised using a web interface. It uses a master-agent model where the server receives inventories sent by the agents in the XML format, which are stored in a MySQL database. The connections are made using http or https.[161]

OCS Inventory also supports package deployment. It is possible to upload packages to the clients from the central management server and use the agent to launch the packages.[162] The software open source and most common platforms are supported, including Windows, Unix, Mac OSX and the common Linux distributions.[163] Different kinds of support and training sessions can be bought on FactorFX's homepage.[164] There is also an active forum, IRC channels and mailing lists available on the OCS Inventory webpage.[165]

OPSI

[154] Lcfg. Documentation http://www.lcfg.org/doc/guide.pdf (2014-04-17)
[155] Anderson, Paul. The complete guide to LCFG p.29. 2006. http://www.lcfg.org/doc/guide.pdf (Retrieved 2014-04-17)
[156] Lcfg. 2009. https://svn.lcfg.org (Retrieved 2014-04-17)
[157] Lcfg. Faq. 2007. http://www.lcfg.org/faq/ (Retrieved 2014-04-17)
[158] Eu-datagrid. http://eu-datagrid.web.cern.ch/eu-datagrid/ (Retrieved 2014-04-17)
[159] Ocs inventory. 2013. http://www.ocsinventory-ng.org/en/ (Retrieved 2014-04-15)
[160] Ocs inventory. Licence. 2013. http://www.ocsinventory-ng.org/en/about/licence.html (Retrieved2014-04-15)
[161] Ocs inventory. Documentation. 2013 http://wiki.ocsinventory-ng.org/index.php/Documentation:Newbie (Retrieved 2014-04-15)
[162] Ocs inventory. Deployment. 2013 http://www.ocsinventory-ng.org/en/about/features/deployment.html (Retrieved 2014-04-15)
[163] Ocs inventory. Supported os. 2013. http://www.ocsinventory-ng.org/en/about/features/supported-os.html (Retrieved 2014-04-15)
[164] Ocs inventory. 2013. http://ocsinventory-ng.factorfx.com/en/ (Retrieved 2014-04-15)
[165] Ocs inventory. Community. 2013 http://www.ocsinventory-ng.org/en/community/ (Retrieved 2014-04-25)

OPSI is a client management system for Windows clients developed by the German company uib GmbH. Linux servers management of the Windows clients, which include automated solutions for OS installation, software distribution and patch management. It can also be used for hardware and software inventories.[166] OPSI, which stands for Open PC Server Integration, is open source. Support and training can be bought from uib GmbH. Cofounding projects are also available. They are projects that are at first exclusive to those who help finance the projects, until they are added to the open source code.[167] OPSI was previously available under the GPLv3 license, but has moved to AGPLv3 since version 4.0.3.[168] The software is still developed and updated and the latest release was in February 2014. [169]

OPSI uses agent installed on the clients, which means that in order for the system to work, the server has to be configured and the agent have to be installed. The agents can then check for updates and patches automatically and install them. OPSI supports automatic OS-installation, in which it is possible to make the entire customization server-side and distribute it automatically. Hardware and software inventory is also supported.[170] The management interface is available by command line, a web service or a graphical interface that are accessed via a web-browser.[171] There is an active OPSI community and there is commercial support available.[172] A map of where OPSI is used is available on their web page, where a few web pages and companies can be found. The software is used mainly in Europe.[173]

PIKT

PIKT is software for system monitoring and configuration management primarily used for problem fixing and reporting.[174] PIKT uses a central master that controls machines with PIKT daemons (similar to agents) installed. In order to detect and report problems, PIKT scripts are installed by the daemons and when a problem occurs it is reported to the system administrator by e-mail. The configuration part of PIKT is that you can install files on your PIKT clients by commands from the PIKT master. The installation is not automatized and each new installation requires manual work. The last update to PIKT was released in September 2007 and a beta-version was released in 2008. PIKT is written in C and the scripts are written in a DSL. The software is open source and is distributed under a GPL-licence. PIKT supports only Linux-based platforms.[175]

---

[166] Opsi. About OPSI http://uib.de/en/opsi/about-opsi/ (Retrieved 2014-04-15)
[167] Opsi. Manual Opsi version 4.0.4 p. 2. http://download.uib.de/opsi_stable/doc/opsi-manual-stable-en.pdf (Retrieved 2014-04-16)
[168] Opsi. License http://www.opsi.org/en/license (Retrieved 2014-04-16)
[169] Opsi. News. http://www.opsi.org/en/news (Retrieved 2014-04-16)
[170] Opsi. Features. http://www.opsi.org/en/features (Retrieved 2014-04-16)
[171] Opsi. Feature-list. http://www.opsi.org/en/features/feature-list (Retrieved 2014-04-16)
[172] Opsi. Support. http://www.opsi.org/en/support (Retrieved 2014-04-23)
[173] Opsi. Opsi-Map. http://www.opsi.org/en/opsi-map (Retrieved 2014-04-24)
[174] Osterlund, Robert. Report problem. 2014. http://pikt.org/pikt/intro/intro_report_problem.html (Retrieved 2014-04-14)
[175] Osterlund, Robert. Intro configure system. 2014. http://pikt.org/pikt/intro/intro_configure_system.html (Retrieved 2014-04-14)

There is no active PIKT community or commercial support available on the official web page.[176] It is claimed on the PIKT web site that there are several big companies using PIKT, but there is no information about which companies or whether they still use PIKT.[177] PIKT has not been updated since 2007.[178]

Quattor

Quattor is a CMS that was originally developed in the European Data Grid project, which was started in 2001 and ended in 2003. CERN was active in the development and they have used Quattor to handle parts of their data grid solutions.[179] Today a community develops Quattor, which is open source under an Apache license.[180] Quattor had its latest update in March 2014.[181]

Quattor was designed to manage large numbers of UNIX machines. It also supports Linux.[182] Quattor has functionality for automatic installation and update. For update Quattor uses a pull model, where configuration modules are installed on the Quattor machines, which pull updates from a configuration database. Each module is responsible for certain functionality on the machine.[183] A high level language called pan, which is developed for system administration and configuration, manages Quattor.[184] Each machine run by Quattor has to have the pan compiler installed.[185]

CERN was involved with the development of Quattor and initially used it for their data solutions, but they later transferred from Quattor to Puppet. According to Ben Jones, who worked with system administration on CERN in 2012, they switched because it was common that small human mistakes resulted in lots of machines failing.[186]

It is not possible to buy support or training sessions on their official web page, there is no information about any companies currently using the software and there is no

[176] Osterlund, Robert. Index. 2014. http://pikt.org/pikt/index.html (Retrieved 2014-04-22)
[177] Osterlund, Robert. Index. 2014. http://pikt.org/pikt/faq.html#who (Retrieved 2014-04-22)
[178] Osterlund, Robert. Index.2014. http://pikt.org/pikt/index.html (Retrieved 2014-04-14)
[179] R. García Leiva, M. Barroso López, G. Cancio Meliá, B. Chardi Marco, L. Cons, P. Poznański, A. Washbrook, E. Ferro and A. Holt. Quattor: Tools and Techniques for the Configuration, Installation and Management of Large-Scale Grid Computing Fabrics. J*ournal of Grid Computing* (2004) 2: 313–322. http://link.springer.com/article/10.1007%2Fs10723-004-7648-2#page-1 (Retrieved 2014-04-22)
[180] Quattor. 2013. http://quattor.org/index.html (Retrieved 2014-04-17)
[181] Quattor. Announcing Quattor. 2014. http://quattor.org/news/2014/03/04/announcing-quattor-14.2.1.html (Retrieved 2014-04-17)
[182] Sourceforge. Quattor. 2014. http://sourceforge.net/projects/quattor/ (Retrieved 2014-04-17)
[183] Edgewall Software. Quattor. 2014. https://trac.lal.in2p3.fr/Quattor/wiki/Web?redirectedfrom=Web/Overview#no1 (Retrieved 2014-04-17)
[184] Github. Quattor. 2014 https://github.com/quattor/pan (Retrieved 2014-04-17)
[185] Jouvin, Michael. Quattor: Managing (Complex) Grid Sites. *Journal of Physics: Conference Series* 119 (2008) 052021. http://iopscience.iop.org/1742-6596/119/5/052021/pdf/1742-6596_119_5_052021.pdf (Retrieved 2014-04-17)
[186] Ben Jones. Configuration Management @CERN - Ben Jones - PuppetConf ’12. 2012.(video). https://www.youtube.com/watch?v=ehVMxbJdld8 (Watched 2014-04-17)

active discussion forum on their website.[187] There are active mailing lists available however.[188]

Radmind

Radmind is a tool created to configure and administer multiple file system on Unix machines. It was developed at Michigan University and the latest release was in 2010. The Radmind software can monitor machines running Unix systems with a master running Mac OSX, Windows, Solaris or Linux.[189] It was first released in 2002.[190]

The main purpose behind Radmind, which stands for remote administration daemon, is to distribute a setup from a single Mac OSX system to several other machines, making sure that they have the same configuration. Radmind can check if the clients are different from the configuration specified by the server and can optionally change the client so that it matches the server. Radmind can be used to update several Macs in a simple way. First the updates are installed on one Mac and then the configuration can be forced onto the other Macs.[191]

Radmind is open source and available under a BSD license, which means that the restrictions for redistribution are at a minimum.[192] There is no information about commercial support, any discussion forums for users or any companies currently using the software.[193] It is possible to subscribe to a few mailing lists that are still active though.[194]

Rex

Rex, or R(?)ex as it is called on the official webpage, is a deployment, orchestration and CMS.[195] Jan Gehring, who released the first version 0.0.1 on Github in 2011, started the project.[196] Rex is still under development and the latest release was

[187] Quattor. Documentation. http://quattor.org/documentation/index.html (Retrieved 2014-04-17)
[188] Sourceforge. Mailing list. http://sourceforge.net/p/quattor/mailman/ (Retrieved 2014-04-25)
[189] University of Michigan. Radmind. 2014. http://rsug.itd.umich.edu/software/radmind/ (Retrieved 2014-04-14)
[190] Sourceforge, Radmind 0-6-1. 2014. http://sourceforge.net/p/radmind/radmind/ci/radmind-0-6-1/tree/ (Retrieved 2014-05-20)
[191] Webapps. Radmind. 2012. http://webapps.itcs.umich.edu/radmind/index.php/Radmind_Manual_for_Mac_OS_X (Retrieved 2014-04-14)
[192] University of Michigan. Software copyright. 2003. http://rsug.itd.umich.edu/software/copyright.html (Retrieved 2014-04-14)
[193] University of Michigan. Radmind. 2014. http://rsug.itd.umich.edu/software/radmind/ (Retrieved 2014-04-14)
[194] Sourceforge. Radmind. 2014. http://sourceforge.net/p/radmind/mailman/ (Retrieved 2014-04-25)
[195] Rex. How to use Rex. http://www.rexify.org/howtos/start.html (Retrieved 2014-04-22)
[196] Github. Rex release 0.0.1. 2014. https://github.com/krimdomu/Rex/releases/tag/0.0.1 (Retrieved 2014-04-22)

made available on April 14 2014.[197] Rex is open source and released under the Apache 2.0 license.[198]

Rex supports the common Unix- and Linux-distributions. Rex itself need only be installed on the administrator's machine, and the platform support on that machine is wider. Rex can even be used on Windows or Mac OSX, as long as a Perl environment is present on the machine. The administrator's machine and the hosts connect to each other using SSH and no agent has to be installed on the clients for the system to function.[199] Rex uses a simple DSL for the descriptions of the servers, but can also be used in shell scripts. The central point of a project in Rex is the Rexfile, which describes different tasks. A task can be for example to install a package.[200] Using this setup configuration management tasks can be done, such as getting health reports, install or delete software and change the configuration of the clients.

There is documentation available from the official webpage, but it is not very extensive according to Tim Schürmann, who writes for Admin Magazine.[201] The people behind Rex do not offer any support, but they redirect to a company called inovex, who offers professional support for Rex. Their homepage is in German, but they offer support in English. They also have a user group where community support is available.[202]

Rundeck

Rundeck was developed in 2010 by people working in connection to the Ville-games such as FarmVille and CityVille as a solution to problems that occurred when upgrading the games.[203] Today developed on Github, Rundeck is a tool that is designed to automate data centers and cloud solutions. It is not a pure CMS, but rather a tool that creates an overview for the configuration scripts and nodes in order to make them work together over platforms. Rundeck also let you define workflows in your system that can be run on a schedule or on demand.[204] The latest Rundeck update was released on April 23 2014.[205]

It is possible to operate Rundeck in a web-based interface or by command line. Rundeck is an on going open source project under an Apache license that is

---

[197] Github. Rex release 0.45.3. https://github.com/krimdomu/Rex/releases/tag/0.45.3 (Retrieved 2014-04-22)
[198] Rex. http://www.rexify.org (Retrieved 2014-04-22)
[199] Schürmann,Tim. King of Computers: Managing Computers with Rex. *Admin Magazine* 19.02.2013 http://www.admin-magazine.com/Articles/Rex/(language)/eng-US (Retrieved 2014-04-22)
[200] Rex. How to start Rex. https://www.rexify.org/howtos/start.html (Retrieved 2014-04-22)
[201] Schürmann,Tim. King of Computers: Managing Computers with Rex. *Admin Magazine* 19.02.2013 http://www.admin-magazine.com/Articles/Rex/(language)/eng-US (Retrieved 2014-04-22)
[202] Rex. Support. http://www.rexify.org/support/index.html (Retrieved 2014-04-22)
[203] Honor, Alex. Alex Honor on Rundeck's history, purpose, and upcoming 2.0. 2013.(video). https://www.youtube.com/watch?v=ZN_LyxmkgWc (Watched 2014-04-15)
[204] SimplifyOps. Rundeck introduction. 2014 http://rundeck.org/docs/manual/introduction.html (Retrieved 2014-04-15)
[205] SimplifyOps. Rundeck Release 2.1.0. http://rundeck.org/docs/history/version-2.1.0.html (Retrieved 2014-04-25)

developed continuously. It is written in Java and supports for example Puppet, Chef and Salt plugins. Currently Rundeck runs on Linux and Mac OSX and testing has begun on Windows.[206] A company called SimplifyOps provides commercial Rundeck support[207] and there are active discussions about Rundeck on Google Groups.[208] There is information about one company that uses the product on their web page, but it not a well known company.[209]

Smartfrog

Smartfrog stands for ”Smart framework for open groups”. It is an open source CMS written in Java and it has its own configuration language. HP developed Smartfrog and some of the components were first developed in 1996.[210] Smartfrog was released as open source during 2003 under an LPGL license.[211] The latest Smartfrog update was released in 2007.[212] It can be used on any platform that supports a valid Java environment, which means that all common platforms are supported.[213] It is not possible to buy support from their homepage, there is not any active community for Smartfrog on their webpage and there is no information about any companies using it.[214] However, a mailing list was used in 2013.[215]

Smartfrog consists of three main parts. The first part is the configuration language that allows the users to define complex systems within templates. The second part is a component model that defines how the different machines within the framework shall be integrated. Examples of components are distributed workflows, failure detection and script execution.[216] The third part of Smartfrog is the configured machines within the system. A daemon is installed on each machine in the system. The daemons read the descriptions from the component models and ensure that the machines provide the underlying support that is required to run the system.

According to Smartfrog developers at HP-labs there has been problems making the different components work together when using Smartfrog. The lifecycle models for different components are very diverse and do not work together in a good way. The Smartfrog language is not as powerful as the developers wanted it to be and during 2009 an update was in progress. No update has been released and therefore the

[206] Github. Rundeck wiki. 2014. https://github.com/rundeck/rundeck/wiki/FAQ (Retrieved 2014-04-15)
[207] SimplifyOps. 2013. http://simplifyops.com/ (Retrieved 2014-04-22)
[208] Google groups forum. Rundeck discussion. 2014
https://groups.google.com/forum/#!forum/rundeck-discuss (Retrieved 2014-04-22)
[209] SimplifyOps. Stories. 2014. http://rundeck.org/stories/ (Retrieved 2014-04-22)
[210] The SmartFrog Configuration Management Framework in ACM SIGOPS Operating SystemReview archive Volume 43 Issue 1, January 2009 Pages 16-25 by Patrick Goldsack, Julio Guijarro, Steve Loughran, Alistair Coles, Andrew Farrell, Antonio Lain, Paul Murray, Peter Toft.
http://dl.acm.org/citation.cfm?id=1496915 (2014-04-22)
[211] Sourceforge, SmartFrog, 2014. http://sourceforge.net/projects/smartfrog/ (2014-05-20)
[212] Smartfrog. News. http://www.smartfrog.org/display/sf/News (Retrieved 2014-04-22)
[213] Smartfrog. FAQ. http://www.smartfrog.org/display/sf/FAQ# (Retrieved 2014-04-22)
[214] Smartfrog. http://www.smartfrog.org/display/sf/SmartFrog+Home (Retrieved 2014-04-22)
[215] Scourceforge. Smartfrog mailing list. 2014.
http://sourceforge.net/p/smartfrog/mailman/smartfrog-users/ (Retrieved 2014-04-25)
[216] Smartfrog. FAQ. http://www.smartfrog.org/display/sf/FAQ# (Retrieved 2014-04-22)

problems remain. Java has also been developed and has a lot of features today that the Smartfrog language does not use.[217]

Satellite/Spacewalk

In 2002 Red Hat released a CMS called ”Network Satellite” for managing Linux servers. Spacewalk is a community-developed version of Satellite that was released in 2008.[218] It is licensed under the GNU GPL v2.[219] The two systems are very similar but Satellite does not contain all of the features developed in the Spacewalk project. Satellite has commercial support provider by Red Hat. Since the company Red Hat stands behind Satellite they focus more on stability. Only the parts of Spacewalk that are stable and meet Red Hat’s vision for Satellite are included. This means that Spacewalk has more features and applies them faster, but no company guarantees stability for the system.[220] Satellite has no commercial support but there is IRC channels and mailing lists available for the community.[221] Both Spacewalk and Satellite are open source.[222]

Spacewalk/Satellite are mostly written in Java but some parts are written in Perl and Python. When using Spacewalk/Satellite agents are installed on the clients and configured from a master server. Spacewalk/Satellite then uses a pull method in order to install and update software for the machines.[223] Spacewalk/Satellite also have features for hardware and software inventory and the software can also be managed from a web interface.

Satellite only supports management for Red Hat Enterprise Linux systems, but Spacewalk also supports some Linux distributions.[224] On the Satellite webpage there are information about some companies using the software.[225] Spacewalk’s

---

[217] Goldsack, Patrick. Guijarro, Julio. Loughran, Steve. Coles, Alistair. Farrell, Andrew. Lain, Antonio. Murray, Paul. Toft, Peter. The SmartFrog Configuration Management Framework. ACM SIGOPS Operating SystemReview. Archive Volume 43. Issue 1. 2009. p.16-25. http://dl.acm.org/citation.cfm?id=1496915 (Retrieved 2014-04-22)

[218] Spacewalk. FAQ. 2012. https://fedorahosted.org/spacewalk/wiki/SpacewalkFaq (Retrieved 2014-05-015)

[219] Redhat inc. Download Spacewalk. 2014 http://spacewalk.redhat.com/download.html (Retrieved 2014-04-25)

[220] Redhat inc. Spacewalk Support. 2014. http://spacewalk.redhat.com/faq.html#support (Retrieved 2014-05-15)

[221] Redhat inc.Spacewalk mailing list. 2014 http://spacewalk.redhat.com/communicate.html (Retrieved 2014-05-15)

[222] Redhat inc. FAQ spacewalk. 2014. http://spacewalk.redhat.com/faq.html#support (Retrieved 2015-05-14)

[223] Ha, John. Brindley, Lana. Macpherson, Daniel. Chan, Athene. O’Brien,David. Lewis, Megan. Red hat Satellite 5.6. https://access.redhat.com/site/documentation/en-US/Red_Hat_Satellite/5.6/html-single/Getting_Started_Guide/index.html (Retrieved 2014-05-15)

[224] Redhat. Spacewalk Support. 2014. http://spacewalk.redhat.com/faq.html#support (Retrieved 2014-05-15)

[225] Redhat inc. Satellite case studies. 2014. http://www.redhat.com/products/enterprise-linux/satellite/case-studies.html (Retrieved 2014-05-14)

latest release 2.1 was made available on March 4 2014[226] and the latest version of Satellite, version 5.6, was released on October 1 2013.[227]

STAF

STAF, which stands for Software testing automation framework, was developed for building test cases and test environments in a simple way, so that developers can focus on building automation solutions instead of test environments. However, it was not designed to do any configuration management tasks, such as deploying servers or install software on servers.[228] It is a project by IBM[229] and was first released in 1998[230] and is still being updated. The latest update was on March 30 2014.[231]

STAF supports a wide range of platforms, including Windows, the common Linux distributions, Unix and Mac OSX. [232] It is open source and is licensed under the EPL (Eclipse Public License) since version 3.2.5. It is possible to interact with STAF using Java, C, C++, Python, Perl, Tcl or Rexx. [233] STAF is actively discussed on Sourceforge[234], but there is no commercial support available on their web page and there is no information about any companies currently using the software.[235]

Discussion

In this section each subsection contain information about why we chose not to include these CMS in our in-depth study. Table 1 also contains a short summary of why the CMS were not included.

BCFG2

BCFG is CMS that fulfil most of our criteria, but it does not seem to be used by larger companies, there is not any commercial support available and the system lacks support for Windows. Other software similar to BCFG has these features. Since there are better alternatives on the market we will not do a comparison study on BCFG.

---

[226] Redhet inc. Spacewalk announcement. 2014. https://www.redhat.com/archives/spacewalk-announce-list/2014-March/msg00000.html (Retrieved 2014-05-14)
[227] Redhat inc. Satellite webinars. 2014. http://www.redhat.com/products/enterprise-linux/satellite/webinars.html (Retrieved 20134-05-14)
[228] Sourceforge. STAF-FAQ. http://staf.sourceforge.net/current/STAFFAQ.htm#d0e109 (Retrieved 2014-04-14)
[229] Negrello, Fabio. Test automation and continuous integration with STAF/STAX2013. *DeveloperWorks*. 2013 http://www.ibm.com/developerworks/opensource/library/os-test-stafstax/index.html?ca=dat (Retrieved 2014-04-14)
[230] Sourceforge. STAF history. 2014. http://staf.sourceforge.net/history.php (Retrieved 2014-04-14)
[231] Sourceforge. STAF index. 2014. http://staf.sourceforge.net/index.php (Retrieved 2014-04-14)
[232] Scourseforge. STAF FAQ. http://staf.sourceforge.net/current/STAFFAQ.htm#d0e109 (Retrieved 2014-04-14)
[233] Scourseforge. STAF installation. 2014. http://staf.sourceforge.net/current/STAFUG.htm#HDRINSTALL (Retrieved 2014-04-14)
[234] Sourseforge. STAF Discussion. 2014 http://sourceforge.net/p/staf/discussion/ (Retrieved 2014-04-14)
[235] Sourseforge. STAF. 2014 http://staf.sourceforge.net (Retrieved 2014-04-14)

Cdist

Since there is no information about it being used in a wide range, it is hard to know whether it really is a good alternative to Chef or Puppet as some claim. We will not go deeper into our study on cdist within this report since Ansible is a more interesting alternative with a similar approach, but with a much bigger following and many more customers.

Isconf

Isconf has not been updated since 2006 and a lot has happened with the CMS since then. Isconf lacks functionalities that modern configuration management tools provides, therefore there will be no further studies of it within this report.

Juju

Juju is an orchestration management tool, which works as a layer above the CMS, rather than being an actual CMS. If you want to install software or update software to a machine, juju can use different CMS within the charms. It can be interesting to support juju in a cloud environment, but it is not software that we will study more closely, since it lacks some of the configuration management capabilities that we are looking for and focuses on orchestration instead.

LCFG

LCFG is a CMS that seems to do the important tasks that we are looking for. It is still being updated and has been used for the European DataGRID project. But there are some aspects with the software that are problematic. It supports very few platforms, no professional support is provided and the European DataGRID project later changed from LCFG to Quattor. Cern, which was one of the contractors, then changed from Quattor to Puppet (see information about Quattor). This leads us to think that there are more modern tools available than LCFG, and therefore we will not include LCFG in the comparison study.

OCS Inventory

The main focus of OCS Inventory is to monitor hardware in a network. While there is support for package deployment as well, the main focus is not configuration management in a cloud environment. Several of the most critical parameters that we are looking for, such as automation of installation and software updates are not supported. Therefore we will not choose OCS Inventory to be one of our main candidates for further review.

OPSI

OPSI uses Linux Servers to configure Windows servers. While OPSI can do some of the features that we are interested in, such as automatically update and install software, the master server has to be a Linux server and it can only manage Windows servers, which reduces its usefulness. Its main focus is hardware

inventory, which is not relevant for this study. Therefore we will not do the comparison study on OPSI.

PIKT

PIKT is not a good candidate for our scenario since the configuration management part of the software is limited. Automated installation and update of software is not possible, which is one of the main criteria we have for our CMS. Furthermore PIKT has not been updated since 2007, which indicates that it is not a modern tool.

Quattor

Quattor is a configuration management that offers automatic installation and update. A limitation with Quattor is that is only runs on Unix machines. It should also be taken into consideration that CERN has replaced Quattor with Puppet, which indicates that Quattor is somewhat dated. In addition it is hard to find information about how to install and use Quattor. Therefore we will not do any further studies on it.

Radmind

Radmind is not designed to manage servers and automate updates. Rather the main focus of the software is to make copies of a single Mac system and distribute them to other machines. While the software seems to do this well, it is not the kind of functionality that we are interested in. Together with the fact that it has not been updates since 2010, which is a long time in this context, makes us draw the conclusion that Radmind is not a good candidate for our scenario.

Rex

Rex is a CMS, which is still active and can be used to do the configuration management tasks that we are interested in. For some it could be positive that no daemons are required, which means that installation and update is a bit faster and easier. However, the platform support is somewhat limited and the support documentation is not very extensive. Ansible also has the daemon-less approach, but with a bigger community and big companies actually using it. Therefore we will do the comparison study on Ansible instead of Rex.

Rundeck

Rundeck is not a CMS but a tool to keep track of other CMS and easily creating workflows. Rundeck might be a complement while using CMS, but since it is not a CMS itself there will be no further studies of it in this report.

Smartfrog

Smarfrog has not been updated since 2007, even though there are known issues that have not been resolved. There is no commercial support for Smartfrog and no active community. These aspects make it hard to use Smartfrog in an efficient way. Therefore there will be no further studies on Smartfrog within this report.

Satellite/Spacewalk

Satellite and Spacewalk are software that support a lot of the configuration management features we are looking for, such as automatic update and installation. The fact that they are bound to Linux distributions is limiting and there are other CMS on the market that have bigger companies using it. Therefore we will be no further studies of the software in this report.

STAF

Since STAF was not built to do the configuration management tasks that we need for our scenario, we will not take a closer look at it. It is clear that it has its focus on building test environments and not configuring and managing servers in a cloud environment, which is what we are looking for in this study.