Automated migration of configuration management systems

A large-scale corporate fusion scenario

UNIVERSITY OF AMSTERDAM

LIA Project Proposal
February 21, 2016

Bram ter Borch
bram.terborch@os3.nl

Siem Hermans
siem.hermans@os3.nl

1 Introduction

Maintaining a consistent configuration across a large series of frequently changing systems can be a daunting task without the use of any configuration management tools. These tools allow companies to automate deployments in a consistent manner and identify changes in hardware and software configuration or general infrastructure in a centralized manner. Most commonly, established tools such as Puppet[1], Chef [2] and Ansible[3] are used for this purpose. However, problems occur when two companies using different mechanisms to manage their systems need to merge. In the end, only one solution has to be selected, which requires a migration trajectory of some sort. An all or nothing approach is hard to perform as it has all the disadvantages which apply to any big bang migration. Therefore, we are interested in what needs to be done to make a gradual migration possible. We are mainly interested in whether it is possible to send commands with one configuration management tool to another, and as such transfer the management role between management systems in a gradual fashion. During the course of this project we will select two of the most popular systems and perform a transition in a small scale lab environment.

2 Related work

Configuration management tools are a frequently researched topic as their presence may lead to improvements in overall workflow. As these tools are mainly aimed at corporate infrastructures, the available literature commonly focuses on specific use cases from an Enterprise standpoint. Frequently these reports present use cases related to security aspects [4] or the repeatability of deployments [5]. Additionally, as time progresses and new tools become available, product comparisons are drawn up. Hardion et al. [6] present a high level comparison of various configuration management tools on the market from the standpoint of a big research facility. Regarding the actual configuration of such tools, Collard et al. [7] describe a method for verifying the configuration of Puppet. However, this paper exclusively focuses on environments with a single configuration management system.

This research project takes a real world migration use case as a starting point. Little official research related to this specific topic is available. Dawson et al. [8] present a motivation for such a migration trajectory and identify Puppet as the most influential configuration management tool. Ansible is identified as a common migration target. In a blog post, Zunker [9] goes into more depth and presents a starting point for performing a gradual migration between Puppet and Ansible. He describes a conceptual method for piping output from Puppet to Ansible and talks about interpreting the output from each tool. This project will form an extension of the work of Zunker.

Moreover, as of late, new concepts for configuration management tools have been surfacing. `mgmt` by Shubin [10] is a self proclaimed 'next generation' configuration management prototype. Up until now, most configuration management tools have been employing a push or pull-based client-server model. `mgmt` uses a distributed architecture which allows for parallelization and a distributed topology. Due to the way `mgmt` is built, the tool may be suitable for a gradual migration trajectory.

3 Research question

As previously mentioned in Section 1, this research project aims to investigate a migration trajectory between two established configuration management tools as a result of a corporate fusion. Because a corporate environment generally requires a stable migration, the focus of this project is to answer the following question:

How to make an operating system automatically migrate between configuration management systems?

As mentioned in Section 2, another interesting aspect may be to look at the architectural differences between established configuration management tools like Puppet, Ansible and next generation configuration management prototypes like `mgmt`. Due to the limited amount of available time, this will be limited to a theoretical analysis. Section 4 will explain what steps need to be taken to properly answer this research question.

4 Methodology

From the list of available management tools in section 2, two of the larger systems are chosen to see if they support the migration from one system to another. These systems are Ansible[3] and Puppet[1]. To get to know them, both systems need to be installed and should manage some servers. Some virtual servers are needed to do this. The way the systems communicate to the clients needs to be investigated and the way of collecting or giving orders should be a big topic during this small project.

The possibility of one management system communicating to another management system is an interesting part.
When migrating from Puppet to Ansible. Puppet needs to send a de-installation of its own management system command, and it should tell Ansible to manage the system from that point on. If the migration is the other way around. From Ansible to puppet, Ansible needs to connect to its client en make it install the Puppet client along the proper configuration. Then it should take the system off the list of management server in Ansible. So the order of all steps is important in these migrations.

When this is all known and there is some time left after researching these topics. The research can look into a new system, `mgmt` which is also referred to in the related work section of this proposal.

Day	Activity
22 February	Familiarization with core concepts in general and Puppet and Ansible specifically.
24 February	Further familiarization with Puppet and Ansible. Evaluation of `mgmt` concepts and architecture.
2 March	Perform proof of concept.
3 March	Perform proof of concept.
7 March	Perform proof of concept.
14 March	Writing the report.
17 March	Finalizing the report.

Tab. 1: General project planning

5 Requirements

In order to perform this project a small lab environment will be created. The hardware for this project is primarily being facilitated by the University of Amsterdam. For this project to work, four virtual servers will be created. Two of the servers will be utilized to run a Puppet and Ansible server respectively. The remaining machines are required to test new management system settings on. Additionally, as we are inexperienced in either configuration management system, we require some quick lessons in Ansible and Puppet. This will mainly be done by via literature study. However, as we are pressed for time we will also attend an inhouse presentation on the topic of configuration management systems.

6 Planning

For the purpose of tracking the progress and to avoid violating the deadline, a rough planning is created. Table 1 reflects the work expected to be done during this project with a date indication. Due to the relatively short length of the project, no further task distribution has been defined. All activities will be performed as a group.

7 Ethical implications

Researching a migration strategy between various configuration management tools has no ethical concerns. Due to the nature of the research, no collection of any personal data will be performed throughout the entirety of the project.

References

[1] What is Puppet. `https://puppetlabs.com/puppet/what-is-puppet`. Accessed: 2016-02-17.

[2] Chef configuration management: Product overview. `https://www.chef.io/chef/`. Accessed: 2016-02-17.

[3] Ansible: Provisioning use case. `https://www.ansible.com/provisioning`. Accessed: 2016-02-17.

[4] Mark Dotson. Security hardening with ansible. *Linux Journal*, 2014(242):4, 2014.

[5] Cristian Ruiz, Salem Harrache, Michael Mercier, and Olivier Richard. Reconstructable software appliances with kameleon. *ACM SIGOPS Operating Systems Review*, 49(1):80–89, 2015.

[6] Vincent Hardion, Darren Paul Spruce, Mirjam Lindberg, Antonio Milan Otero, Julio Lidon-Simon, Jerzy Jan Jamroz, and Andreas Persson. Configuration Management of the Control System. *Proceedings of ICALEPCS2013*, pages 1114–1117, 2013.

[7] Joseph Collard, Nimish Gupta, Rian Shambaugh, Aaron Weiss, and Arjun Guha. On Static Verification of Puppet System Configurations. pages 1–13, 2015.

[8] Chris Dawson, Susan Hall, and Lawrence Hecht. Are docker users migrating to ansible and away from puppet and chef? - the new stack. =http://thenewstack.io/are-docker-users-migrating-to-ansible-and-away-from-puppet-and-chef/, 2014.

[9] Christian Zunker. Migrate from puppet to ansible - codecentric blog. `https://blog.codecentric.de/en/2014/12/migrate-puppet-ansible/`, 2014.

[10] James Shubin. Next generation configuration mgmt. =https://ttboj.wordpress.com/2016/01/18/next-generation-configuration-mgmt/, 2016.