Configuration Management: Chef & Puppet

Internet-scale Distributed Systems Seminar Report

Felix Wieser
TU Munich
Munich, Germany
felix.wieser@tum.de

Daniel Federschmidt
TU Munich
Munich, Germany
daniel.federschmidt@tum.de

ABSTRACT

A good report has an abstract!

KEYWORDS

Configuration Management, DevOps

1 INTRODUCTION

1.1 Why configuration management

- Deploy many applications on many machines (a.k.a. nodes)
- Update an application on all nodes simultaneously
- Simplify rollbacks
- Keep environments consistent among multiple entities (i.e. dev/testing/production)
- Keep records of all changes of the infrastructure

2 SYSTEM ARCHITECTURE

- Push config vs. Pull config

High-level overview.

2.1 Configuration Management

This is a citation [?].

2.1.1 Insight A. A subsubsection

2.1.2 Insight B. Another subsubsection.

2.2 Chef

- Components
  - Chef DK
    - ∗ ”Location where users interact with chef”
    - ∗ Creation of cookbooks
    - ∗ Test of cookbooks with Test Kitchen
      - · Describe Test Kitchen here
  - Chef Server
    - ∗ Hub for configuration data (cookbooks)
    - ∗ Pull configuration: Nodes pull cookbooks from server
  - Clients
    - ∗

2.3 Puppet

- Uses a master-slave architecture
- Uses pull config
- Ressource management:
  - Manifests describe the node configuration
  - Groups of ressources can be organized into classes ⇒ i.e. config for entire application can be grouped
  - Modules combine manifests and data to improve code organization
- Server node connection via SSL works as follows:

(1) Node sends normalized data to the Puppet master
(2) Server uses this data to compile a catalog, that specifies how the node should be configured
(3) The node reports back the successful config to the master (Visible on the Puppet Dashboard)

2.4 Evaluation

Actual hard work happens here - many thoughts!

3 CONCLUSIONS

Brilliant results!