

# 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

Found on [https://docs.chef.io/chef\\_overview.html](https://docs.chef.io/chef_overview.html)

- Components
  - Chef DK (Chef Development Kit)
    - \* Computers running Chef DK are called Workstations
    - \* Creation of cookbooks
    - \* Test of cookbooks with Test Kitchen
      - Describe Test Kitchen here
    - \* Components of workstations
      - Knife
        - Interface between local chef-repo and Chef server
      - The chef-repo
        - Cookbook storage
        - "The chef-repo should be synchronized with a version control system (such as git), and then managed as if it were source code"
  - <https://docs.chef.io/workstation.html#configure-ruby-environment>
    - knife.rb
      - File to specify configuration details for knife
  - Chef Server

- \* Hub for configuration data (cookbooks)
- \* Pull configuration: Nodes pull cookbooks from server
- Chef client
  - \* Must be installed on each node
  - \* Performs
    - Registration and authentication of the node with the chef server
    - Building the node object
    - Synchronization of cookbooks
    - Compilation of the resource collection
    - Configuration of the node
    - Exception and notification handling
- Ohai
  - \* Collects system configuration data for use within cookbooks
  - \* Includes many built-in plugins to detect state
  - \* Attributes contain: OS, Network, Memory, Disk, CPU, Kernel, host names, virtualization, etc.
- Chef Supermarket
  - \* Sharing and management of community cookbooks
- Cookbooks contain
  - attributes
  - cookbook\_file
  - libraries: Ruby code can be included in a cookbook
  - metadata: Stored in *metadata.rb*. Helps the server deploy the cookbooks to the nodes correctly
  - recipes
    - \* Authored in Ruby
    - \* Collection of resources
    - \* Must define everything that is needed to configure the node
  - resources
    - \* Describes the desired state for a configuration item
    - \* Describes the steps to achieve the desired state
    - \* Contains resource type
    - \* Grouped into recipes
  - templates
    - \* Used to dynamically generate static text files
    - \* May contain Ruby
    - \* Intended to manage configuration files
  - tests

### 2.3 Puppet

- Uses a master-slave architecture
- Uses pull config
- Resource management:
  - Manifests describe the node configuration

- Groups of resources can be organized into classes  $\Rightarrow$  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!