System Orchestration with Capistrano and Puppet

Christian Patsch
Berlin, April 9th, 2014

Agenda

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- About me
- Why Puppet ? Why Capistrano ?
- Introduction
- Problem Statement
- Capistrano
- Extension supply_drop
- Alternatives
- Summary

Motivation

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- Cloud Computing is a paradigm change
- Trend towards more agile system administration patterns, driven by business requirements
- Years after Consolidation movement into virtualized environments, the number of hosts is growing heavily → Configuration Drift
- Requirements for the more complex IT infrastructure stay the same: stability, Control/Governance reqs, Security
- Changes and Deployments in higher frequency

Introduction

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- A short history of „System Configuration Tools“
  - scripts / system-specific
  - remote-ssh, cluster-ssh
  - cfengine, FAI,....
  - new ideas around 2005 ff. – declarative vs. imperative etc.
  - Puppet, Chef, Ansible, Saltstack,.....YADT
- Advantages/Disadvantages

Criteria

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- pull/push - mechanisms
- Master as central point of truth at all time ?
- Learning curve
- Developer as Operator, or Operator as Developer ?
- Ad-hoc/do now vs. planned/regularly
- Order of executed tasks
- Level of abstraction needed
- Supported Operating systems

Problem Statement

- Master/Agent and therefore centralized architecture not always desired or best approach depending on environment
- Problems and risks using complete and autonomous deployment from a central instance
- One more service that has to be operated, thinking of all aspects this has in an enterprise data center

Problem Statement -2-

- How to overcome network hurdles ?
- Completely manual usage of puppet recipes needs maintenance and „human power“, explicit access for every puppetized hosts not the most elegant solution
- There are environments where changes are not that frequent, and maybe even standardized
- Example: LDAP-HA with pen LoadBalancer

Capistrano v2

- Coming from Ruby Ecosystem, project focus is remote deployment for applications – especially RoR
- inspired by rake (Ruby make), own DSL
- Explicitly designed for usage of an admin host that executes deployment tasks over SSH connections
- Config-source (central truth) only necessary here
- „Capfile“ as Command Tool

Capfile - Example


```
set user:, 'sysadmin'
role :puppet , „ubumastervm“
desc "Install puppet from puppetlabs"
task :setup_puppet, :roles => :puppet do
  run "#{sudo} wget -O /tmp/puppetlabs-release-precise.deb
http://apt.puppetlabs.com/puppetlabs-release-precise.deb"
  run "#{sudo} dpkg -i /tmp/puppetlabs-release-precise.deb"
  run "#{sudo} rm -f /tmp/puppetlabs-release-precise.deb"
  run "#{sudo} apt-get -y update"
  run "#{sudo} apt-get -y install puppet"
end
```

Capistrano and Puppet

- Combination of both is an nearby idea
- Easy setup
- enables „puppet solo“
- Capistrano opens SSH connection and applies the tasks and also recipes on the configured hosts
- Advanced Configuration possible, e.g. roles etc.
- A cap run can be activated directly, via using the desired task or namespaces

Detailed run

- # cap <taskname>
- Connection to given server(s) over SSH
- Key-authentication not mandatory but recommended ;-)
- Single task descriptions from Capfile are executed , for example:
  - `upload <recipe>.pp`
  - `run <command line task>`
  - `stream <logfile>`
- Example for 'puppet apply'
  - `run puppet apply <recipe>.pp`

Capfile – Example 2


```
desc "Setup check-mk agent"

task :setup_check_mk, :roles => :puppet do

        upload("check_mk.pp", "/tmp/check_mk.pp")

        run "#{sudo} puppet apply /tmp/check_mk.pp"

        run "#{sudo} rm -f /tmp/check_mk.pp"

end

=======
```

Configuration logic stays in the puppet manifest itself, assures that::

- packages for check_mk are installed

- check_mk init script is activated and executed, start using xinetd

=> recipes/manifests have to be created once, can be deployed on selected hosts multiple times.

Advantages

- Easy installation
- Better control of deployments
- Description of configuration on the task-layer allows more granular deployments without much effort
- No additional services have to be activated or installed
- Secure Connections using SSH with well-known and established permissions and access control
- Tracking/reporting of invocations possible with default tools (sudo,...)

supply_drop extension

- Ruby gem for extended integration of capistrano and Puppet
- Specific capistrano tasks for puppet :
  - cap puppet:apply
  - cap puppet:noop
  - …
- No need for manually copying manifests and other files, rsync task included
- Recommended for staging environments/configurations

Other Tools

- Approach of central admin workstation as “control station” is used by other solutions, too
- No final case made for master/agent concept vs. decentral administration
- Key in all cases is the versioning control and the functionality that is created by using a VCS, especially in case of DVCS
- No general recommendation possible – research and PoC needed to define solution that fits best

Fabric

- “capistrano for pythonists”
- Similar approach, but writing small python scripts is needed
- Integration efforts with puppet have already been made (see on github)
- Example:

```
from fabric.api import run

def host_type():
    run('uname -s')
```

iron_chef & chef-solo

- Capistrano gem for chef (https://Github.com/scottvrosenthal/iron_chef)
- Automatically sets up directory structure for project
- List of default tasks for capistrano and chef working together
- `#cap <node> chef:bootstrap`
  `.......`

Ansible

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- Configuration management tool built with the aforementioned ideas in mind
- Written in Python
- YAML playbooks – jinja2-based templating
- Push default – pull possible
- SSH as default transport, nodes do not need any additional software
- http://www.ansible.com, http://galaxy.ansible.com

Summary

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- Testing and learning of tools also worthwile in smaller environments
- Disadvantages of current implementations can be aligned
- Learning curve and initial effort are significantly lower than implementation of complete environments for system configuration mgmt.
- Advantages still exist: Reusability, documentation source, prevention of “silo” thinking....
- Best-Practice approach

Thank you...

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

.....for your attention =) Questions ?

Christian Patsch

Christian.Patsch@GONICUS.de

GONICUS GmbH
Möhnestr. 55
59755 Arnsberg

http://www.gonicus.de

Resources

OSDC.de
OPEN SOURCE DATA CENTER CONFERENCE

- https://github.com/capistrano/capistrano/wiki
  Capistrano v2 Documentation
- https://github.com/pitluga/supply_drop

  Github-Repository for supply_drop Gem
- https://www.braintreepayments.com/braintrust/decentralize-your-devops-with-masterless-puppet-and-supply-drop
  blog entry from author of supply_drop Gem