Hybrid automation at Rabobank

Deployment @ Rabobank

André Rozendaal / Jan van der Noll
Rabobank System Engineers

Koen van Bakel
Red Hat Solutions Architect

May 4th 2017
Rabobank

- Top 3 banks in the Netherlands
- +/- 40,000 employees
- Dutch market - Generic bank
- International - Food and Agri
Agenda

- Introduction
- Project Setup
- Initial Design
- Final Solution
- Future developments / ideas
Introduction

- Stable/enforced baseline (2 hours)
- Application preparations up to 2 weeks
- Application specific settings
  - Different methods (rpm, script, manual, ...)
  - Often hard to reproduce
- OS / Middleware migrations very costly
Old style: System manufacturing - component driven
New Style: System manufacturing - turn key solution
Project Initiation

- Solution needed to fit in overall architecture
  - In development at the time
  - Flexible choice (API based) needed
- CloudForms / Satellite 6
  - Satellite 6 -> logical successor Satellite 5
  - CloudForms
    - Native solution for RHEL
    - Can be integrated in other overall solution (using API)
Project Setup

• Project outline defined
• Scrum method
  – Detailing design ‘during development’
• Opportunity to adjust to changing insights
  – Technical
    • Rise of Ansible
  – Organizational
    • Devops teams
Initial design

- Environment
  - Cloudforms / Satellite 6
  - All OS / Middleware and OS related application config defined in Satellite 6
  - Puppet as provider for all defined config
  - No manual changes on the system
  - Application deployment from other tooling (customer choice) optionally kicked off by Cloudforms deployment
Final Solution

- Environment
  - Cloudforms (interface / API provider)
  - Satellite 6 / Puppet
    - OS software and baseline configuration
  - Ansible / Tower
    - Middleware software and configuration
    - OS related application configuration
    - Available for Application Teams for Application development
Functional
Technical Implementation
CloudForms

- Interface (GUI / API)
- Orchestration (internal environments)
  - Manage VM’s within VMWare
  - Deploy / extend / decommission
  - DNS (add/remove)
  - ITIL (Assets, relations)
  - Backups (TSM node/schedule)
  - Monitoring
  - LDAP (ownership / authorizations)
Satellite6 / Puppet

• Build image(s)
  - Basic kickstart / hardening

• Deploy servers based on images (speed)
  - Network configuration via VMWare Custom . . . .
  - Initial activation script (Subscribe to Satellite / initial Puppet run Puppet content depending on environment (Sandbox/DTA/PA/P)

• Baseline / security configuration (Puppet)
• Baseline / Security enforcing (Puppet)
• Puppet - R10K
• Remote execution (i.a. used by CloudForms)
Ansible Tower

- Playbooks from GIT repo’s
- Infra
  - Middleware playbooks
- DevOps teams
  - Application specific OS config playbooks
  - Application playbooks
Security

● Division Sandbox / DTA / PA / Prod
● Access to systems (order and manage) based on department of user and environment (Cloudforms tagging)
● Different Ansible keys for departments and environments (SandBox/DTA/PA/P)
● Access to environments can be differentiated
  Optional: 4-eyes principle for running playbooks
Usage example

- SandBox systems
  - Root access, autoexpire (1 week)
- Within devops flow:
  - Order system (API)
    - Build VM, incl. authorizations and executed playbook
  - Execute automated tests
  - Process / save results
  - Decommission system (API)
Future

- **CloudForms**
  - Integrate OpenShift / Openstack
  - Replace current chargeback
  - Pooling mechanism (faster, reliability, failing api’s)

- **Satellite 6**
  - OpenSCAP
Questions?

andre.rozendaal@rabobank.nl
jan.van.der.noll@rabobank.nl
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews

#redhat #rhsummit
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.