



# Migration from ELFMs to Agile Infrastructure

[vitor.gouveia@cern.ch](mailto:vitor.gouveia@cern.ch)

CERN, IT Department

HEPiX Fall 2013 Workshop

- Configuration Management at CERN
  - ELFms
  - Agile Infrastructure (AI) Project
- Migration from ELFms to AI
- Migration activities
  - meetings
  - training
  - documentation
- AI Challenges
- Conclusions



We aim to provide the tools, process and the infrastructure for the service managers and system administrators to manage the configuration of any machine hosted in the CERN's Computer Centre



## Extremely Large Fabric management system (ELFms)

- includes a set of tools
  - **Quattor** for configuration management
  - **Lemon** for performance and exception monitoring
  - **LEAF** Hardware Management System and State Management System

ELFms



quattor



- Quattor current situation:
  - configuration management software
  - developed in-house
  - first release 2003
  - successfully managing around **8000 servers**
  - we have more that **260 persons managing configurations**
    - Including **system administrators from CERN IT and experiments, computer centre operators and technicians**
- Why changing??
  - not scalable to the number of machines expected in the future
    - Geneva Computer Centre and Wigner Computer Centre
  - the costs to develop and maintain our own configuration solution are not reducing
  - hire professionals trained in a tool developed in-house is not easy
  - small community
  - support problems





## Agile Infrastructure (AI)

- Configuration Management System
  - Puppet for the configuration management
- Cloud infrastructure
  - new VM hosting service – cloud approach
    - OpenStack as cloud software for virtual machines
- Monitoring
  - new alarm system (lemon/GNI)



- based on free and open source tools
  - Puppet and its suite of tools, Foreman, Git, etc.
- scalable to the increasing number of computer nodes
  - aim  $\approx$  300,000 virtual machines
- compatible with the cloud approach
- processes and workflows completely rethought

## Configuration Management



## Benefits of the AI approach:

- Puppet has a very active community: often problems are fixed and missing features implemented before you even report them
- community support is working well until now
- documentation: the Puppet documentation is very good and is continuously being improved
- fast deployment
  - tools already available and “ready to use”
- easier to hire people who have skills in a widely-used tools than our internal tools





## What is the best way to migrate to the new system?

- it was decided to redo everything
- no direct conversion from Quattor managed machines to Puppet managed machines...
  - in other words: no direct conversion of the Quattor configuration files
- Why?
  - easier and faster to redo everything than try to convert Quattor configurations



- **ELFms** and **AI** for the moment they **live in parallel**
- It is expected that over time less and less servers will be configured/managed with ELFms/Quattor
  - we aim to do most of the transition to AI/Puppet during the next two years
- Migration: It was decided to involve system administrators in early stages of the project
  - Why?
    - collect their feedback
    - promote collaboration between the system administrators and the AI architects
    - help us to adapt AI to the system administrators requirements



What have we been doing to introduce the new configuration management system to the system administrators?

- one year after the starting of the project a series of regular meetings started to happen with system administrators and service managers
  - introduce the new Configuration Management System
  - provide information about project and future plans
  - since the first meeting the system administrators have received early access to the Configuration Management System
    - deliberated decision
    - it was important for us their “user experience” with our system



## Example how the collaboration have been working...

- three months after the start of the regular meetings it was introduced the first development process of the Puppet modules
  - puppet modules are reusable set of files and Puppet manifests to configure a piece of software (e.g. apache, mysql, etc.)
  - puppet modules can be shared by several services
    - it is necessary to coordinate their changes
    - we don't want people to step on each other



## Results of meetings:

- feedback on the process
  - we had some shortcomings with the first iteration
    - issues in pushing changes into production at different speeds
    - the test process was not easy
    - the ownership and responsibility of the changes weren't clear
    - etc.
  - it was necessary to change the process  
**several improvements were made...**





Nine months after the introduction of the first version development process...

- a new version was introduced...
  - the new version mitigates several problems encountered on the first version
  - feedback was very positive
  - still on-going discussion
- **a positive collaboration between the AI team and the system administrators improved the development process of the Puppet modules**



- Eight months after the start of regular meetings we wanted to increase the number of participants...
- A new strategy was necessary...
  - during the last summer five training sessions were organised to introduced the new system to the system administrators and service managers
    - hands-on approach
  - helped to get even more system administrators involved in the on-going project
  - after one morning:
    - the users were able to create the first Puppet managed virtual machine
    - once again important feedback was collected during these training sessions



## Keep the documentation of our processes up-to-date has been a great challenge

- the evolution of the project and constant changes make very difficult to have a up-to-date documentation
  - even during the training sessions the things were changing
- information spread across several places
- several tries to mitigate the problem
  - forum (later replaced by standard support lines)
  - the regular meetings with system administrators mitigates the problem but doesn't solve it
    - direct and fast communication of the changes
    - presentation slides become obsolete with the evolution of the project



- Documentation
  - keep up the constantly changing environment
  - define and maintain the documentation of all the processes
- Certain things had to be re-done differently
  - trial and error approach
  - time and effort consuming
- Maturity of Puppet and the other tools
  - e.g. it was necessary to rollback the migration from Puppet 2.7 to 3.0 until a bug fix was provided
- Puppet not completely adapted to an heterogeneous population of system administrators working on very different services
  - secrets isolation
  - code from independent services running on the same infrastructure
- Integration of the various tools we selected proved difficult sometimes



- Approach taken was well appreciated
  - different from the past...
  - release early, release often
    - feedback loop between developers end users/testers has been working quite well
  - the new developments are totally focused on the users needs
  - the feedback has been very positive
- The use of the off-the-shelf open source tools has some benefits
  - documentation
    - you even have books on it
  - large and active community is supporting Puppet





The benefits have been great:

- for the first time during the development of a project in the IT department system administrators are participate closely in the development of a new project
  - The development process is entirely open...
- **Puppet and friends model is moving forward**
  - **4000 hosts in production**
  - **150 persons managing configurations**



# Thanks!!



## Questions?