

Managing configuration drifts in large computing infrastructures: an experimental approach at CERN

Student: Andrea Giardini

Supervisor: Prof. Anna Ciampolini

Mentor: Ben Dylan Jones

December 19, 2016

University of Bologna

Outline

Introduction

CERN

Configuration management

Package Inventory

Project Structure

Results

Continuous Integration

Project Structure

Results

Conclusions

- European Organization for Nuclear Research
- Situated in the border between Switzerland and France
- 22 Member states
- Big challenges



Data Centres

Two data centers:

- Geneva
- Budapest

Two dedicated links:

- 2 x 100Gbps

The number of resources is growing year by year. As today:

- 18k servers
- 180PB on tape
- 260PB on disk

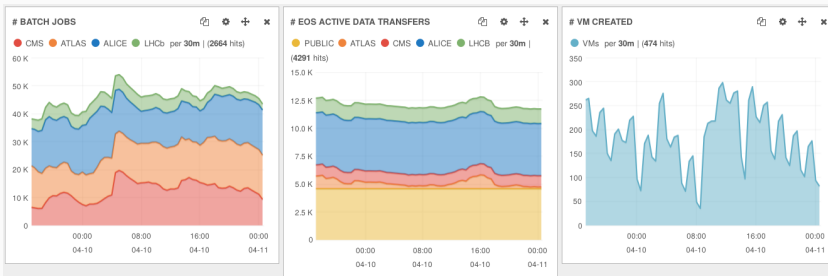


Cloud Computing

Requirements started to grow

- Agile approach was needed

Since a few years we started using **Openstack** to deploy virtual machines for our users and Puppet to configure the services



Configuration Management with Puppet

Puppet is an open-source **configuration management tool**. It is designed to manage the configuration of Unix-like and Microsoft Windows systems declaratively.

- Service configuration
- Users/Groups management
- Automates repetitive tasks



Package Inventory - Introduction

Package drifts started to be a problem:

- Servers with outdated packages
- Difficult to spot
- Users forcing their servers not to update

It is not easy to keep all the packages in sync and guarantee security



Package Inventory - Project Structure

The configuration team needed a tool to query packages over a large number of hosts in a timely manner.

Package Inventory is made by three components:

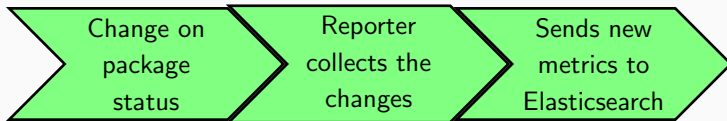
- Reporter
- Elasticsearch Cluster
- Cli (Command-line Interface)

Package Inventory - Reporter

Software installed on every server, integrated with Yum using a plugin.

- Reports to Elasticsearch every package change
- Fault-tolerant
- Real-time updates

Elasticsearch stores and indexes all the metrics to make them searchable.



Package Inventory - Command-line Interface

The **Cli** is used by the users to query the metrics that are stored in the Elasticsearch cluster.

- Possibility to compare clusters of machines
- Package history
- Package version across a set of servers

Optimized to scale for large queries:

Query time to compare over twenty hundreds hosts is two minutes.

```
pkginv -H bi/batch/gridworker/aishare  
-e production  
compare
```

Processing hosts with:

- Hostgroup: bi/batch/gridworker/aishare
- Environment: production

Group 0	Group 1	Group 2
b6b3d71dfa	b6b1576a51	b60ba703db
b636f67ca8	b604a66b3b	b6e493a1b6
b62b21e394	b6c6f66a36	
b65573753a	b67d439ea5	
b6a09855b8		
b6a2bb83c8		
b67af0481a		
b64f0f53c1		
b69b92afd1		
b67f818ae3		
b67cc79825		
b622b1cfdc		

Package Inventory - Command-line Interface

Now that we know which hosts are drifting we can get more details.

Packages can have:

- Different versions
- Different status

Comparing two hosts gives us details about the installed packages.

```
pkginv -m 'b6b3d71dfa b6b1576a51' compare
```

Package	Field	b6b3d71dfa	b6b1576a51
httpd		Present	Not present
httpd-tools		Present	Not present
gridsite		Present	Not present
mod_ssl		Present	Not present
castor-lib	epoch	8.slc6	9.slc6
castor-rfio-client	epoch	8.slc6	9.slc6
castor-ns-client	epoch	8.slc6	9.slc6
castor-devel	epoch	8.slc6	9.slc6

Package Inventory - Results

- Installed in more than **55 hundreds of servers**
- Spotted over **two hundreds of servers** out of sync
- Used extensively to monitor the deployment of security updates
- Debugging performance issues

Continuous Integration - Introduction

Shipping changes to production used to be a manual procedure:

- Multiple actions involved
- Human error

The process can be automated:

- Less time wasted
- More organization

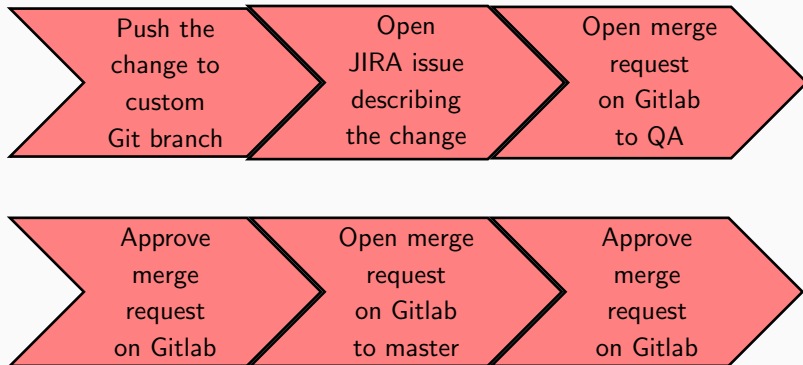
Implementation of a Continuous
Integration platform using **Jenkins**

- Open source software
- Highly customizable
- Active community

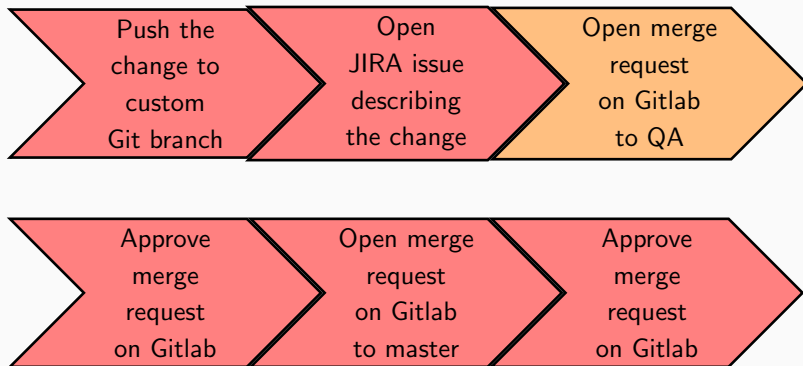


Jenkins

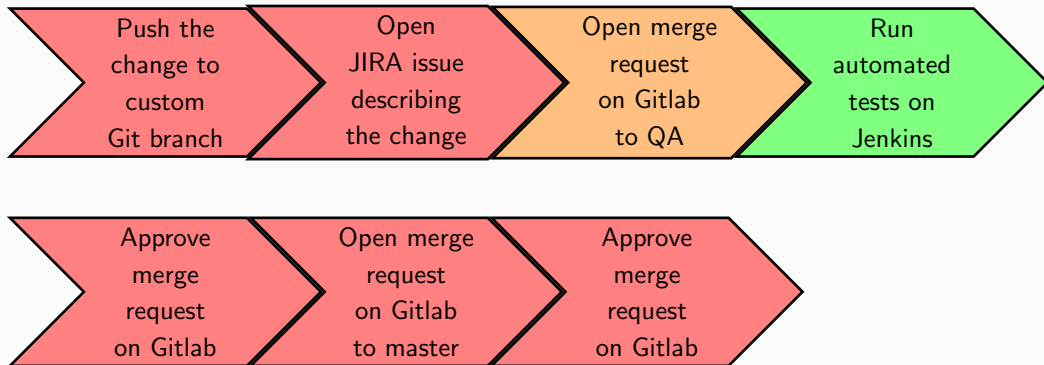
Manual configuration change process



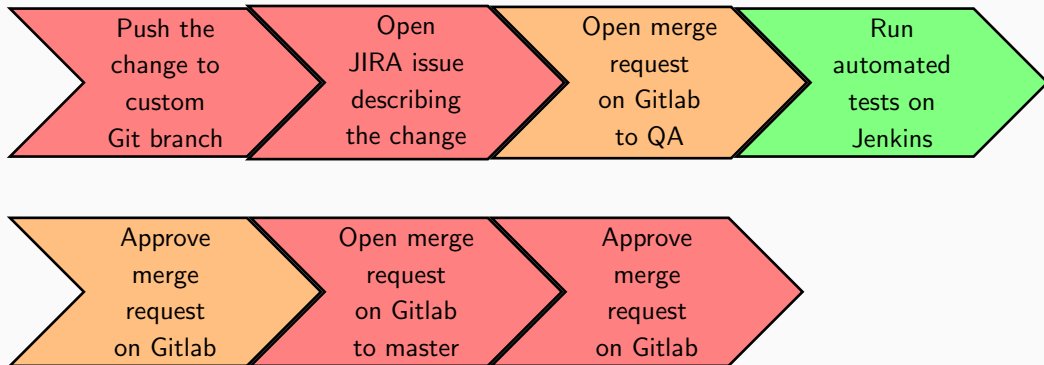
Automated configuration change process



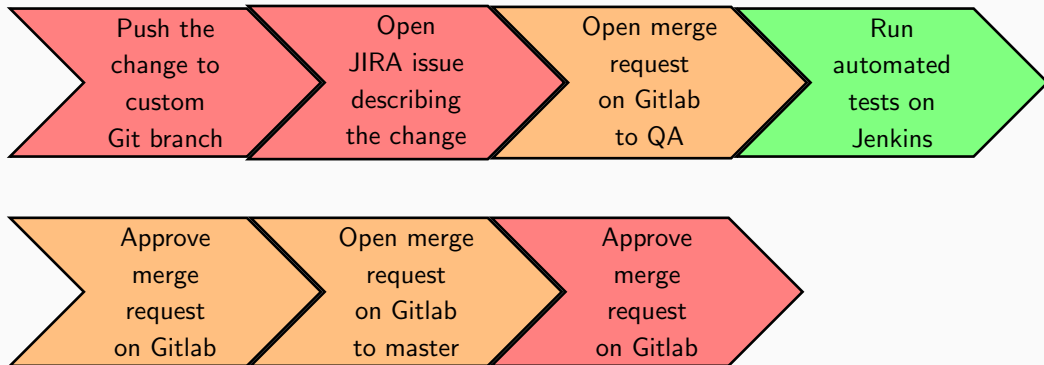
Automated configuration change process



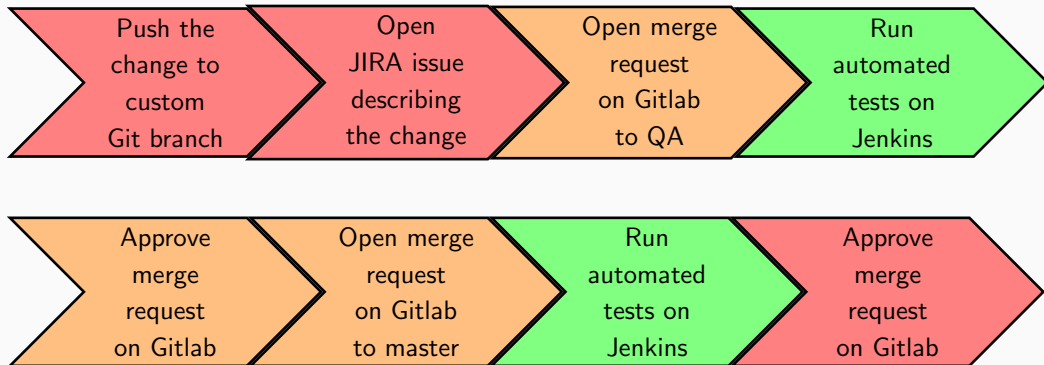
Automated configuration change process



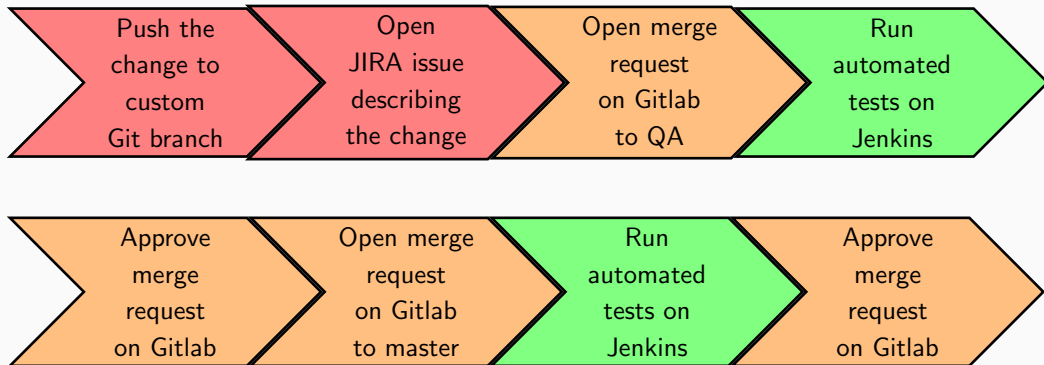
Automated configuration change process



Automated configuration change process



Automated configuration change process



Gitlab merge request

Merged

Merge Request #33 · created by  Jenkins CI · 2 days ago ·  2 days ago

Ignore this ticket

This merge request was triggered by [CRM-1497](#)

Request to merge **mychange** into **qa**



Jenkins CI @jenkinsc · 2 days ago
Build triggered.

Master



Jenkins CI @jenkinsc · 2 days ago
Build Started: <https://jenkins.cern.ch/servant/job/it-puppet-module-pulp/2/>

Master



Jenkins CI @jenkinsc · 2 days ago
Status changed to merged



Jenkins CI @jenkinsc · 2 days ago
Build finished. Tests PASSED. Build results available at: <https://jenkins.cern.ch/servant/job/it-puppet-module-pulp/2/>

Master



- **Extremely customizable infrastructure**
 - Users can define their tests
 - Service managers specify how the test server needs to be built
 - Tests can be re-triggered with a comment on GitLab or pushing new commits
- **Procedure completely automated**
- Every service manager is responsible for its Puppet module
- No need for users to learn how to use Jenkins

Package Inventory and the **Continuous integration** platform have been **successfully integrated in the infrastructure at CERN.**

Package Inventory has been used to spot several inconsistency and it has been used extensively to report several misconfigurations.

The Continuous Integration platform has been used to automate the deployment of changes to production. Unfortunately, writing meaningful tests requires effort: obtaining a fully automated pipeline with complete tests will require time.

Questions ?