



# Configuration Management @ CERN

## Going Agile with Style

Andrea Giardini

CERN

*andrea.giardini@cern.ch*

24 April 2015 - PuppetCamp Berlin



Configuration Management @ CERN



# Outline

## Introduction

What is CERN

Datacenters Overview

## Puppet @ CERN

Current Infrastructure

Modules, Hostgroups and Environments

## Configuration Management

Managing Changes

Tools

## Conclusions



# What is CERN

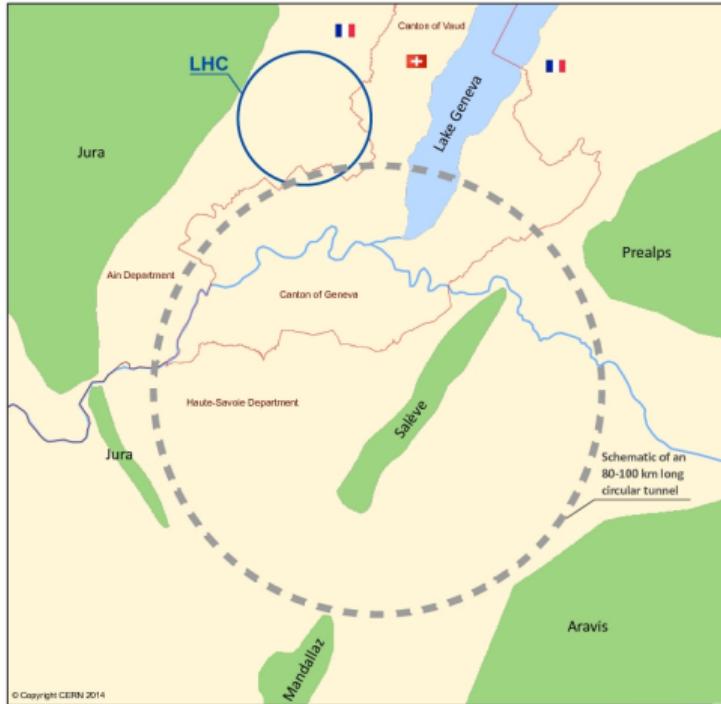
- ▶ European Organization for Nuclear Research
- ▶ Situated in the border between Switzerland and France
- ▶ 21 Member states
- ▶ Big challenges



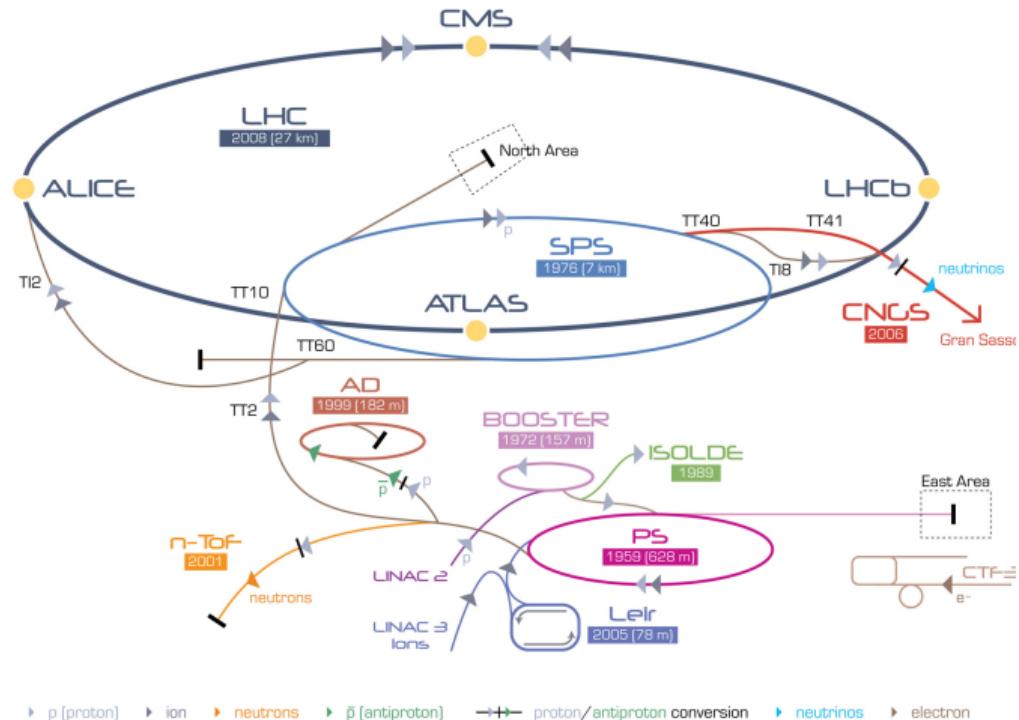
Configuration Management @ CERN



# Big Challenges - The FCC



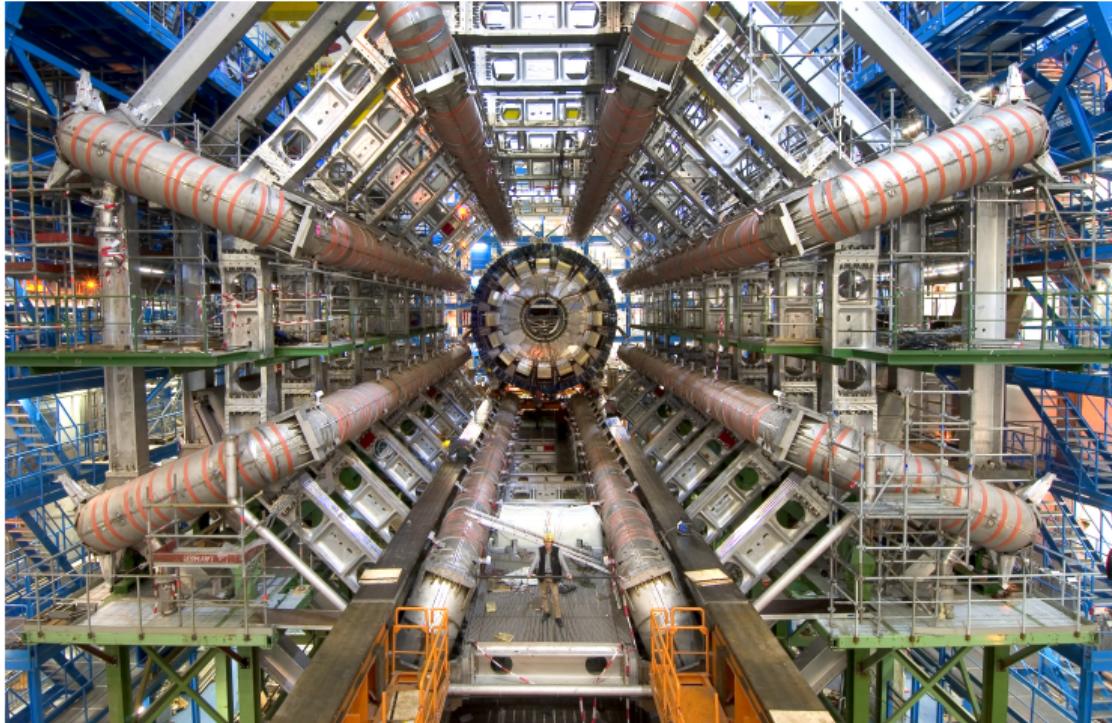
# The LHC



Configuration Management @ CERN

puppet  
camp

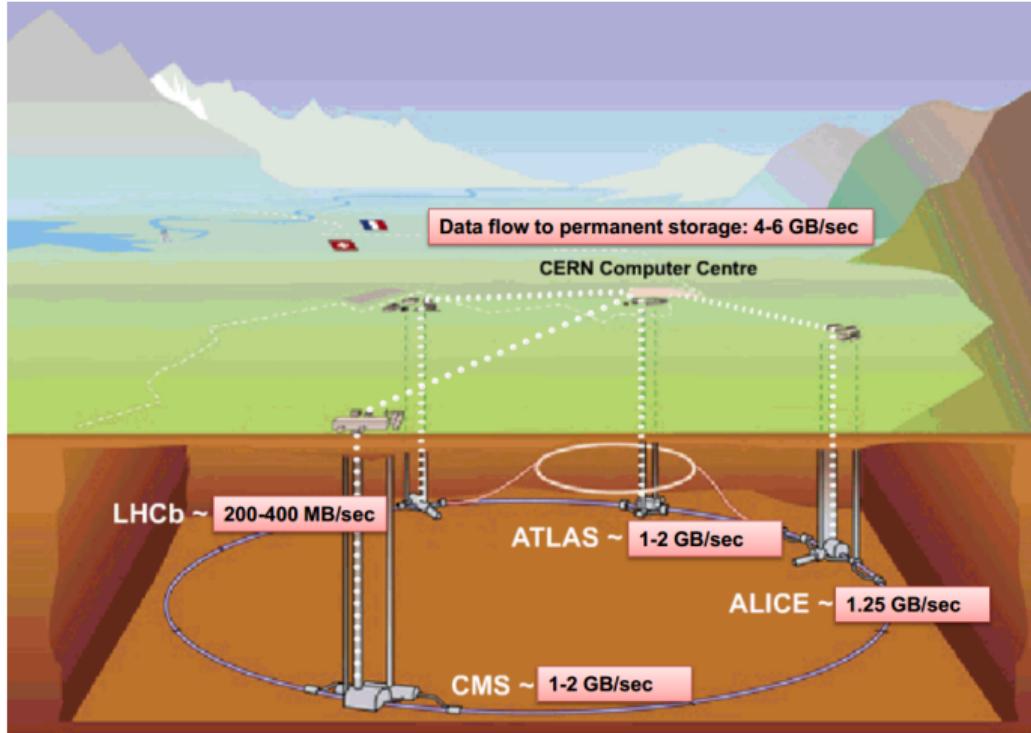
# The Detectors



Configuration Management @ CERN



# Data Flow



# Datacenters in Numbers

Two datacenters:

- ▶ Budapest
- ▶ Geneva

Two dedicated links:

- ▶ 2 x 100Gbps

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

- ▶ 15k servers
- ▶ 100PB on tape
- ▶ 200PB on disk

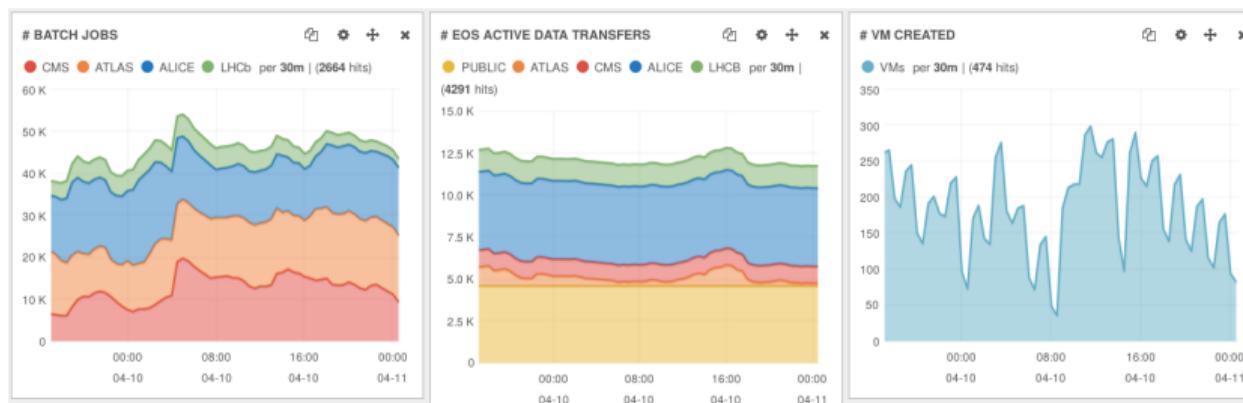


# Going Agile

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 @ CERN



# Our Setup

We started using Puppet a few years ago and, since then, things evolved a lot ...

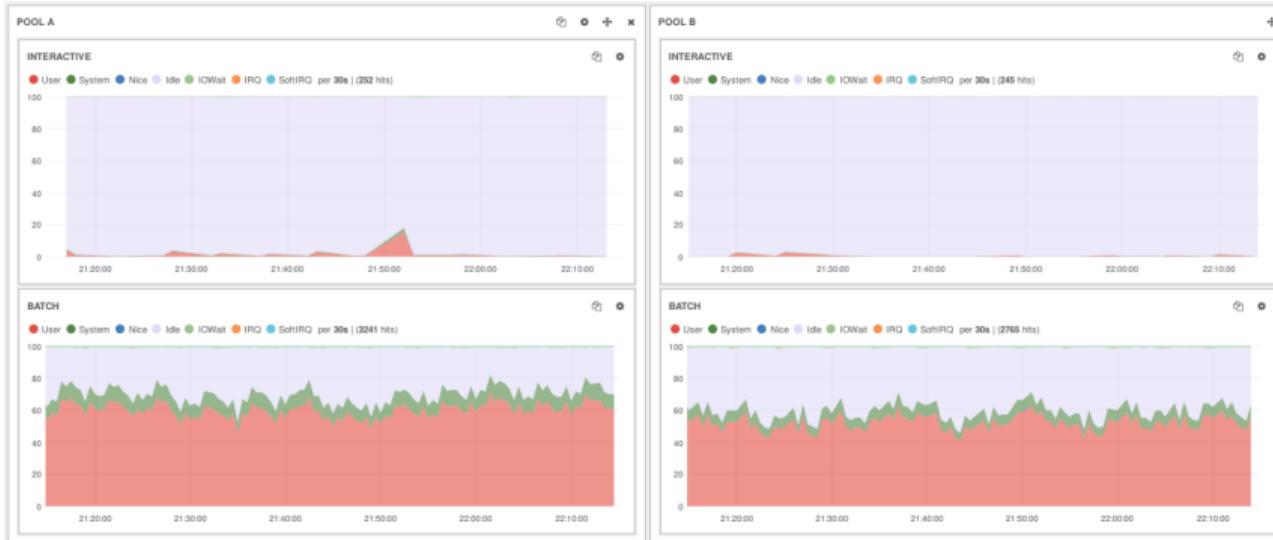
We changed several time the configuration of our puppet masters in order to keep up with the requests and we found out that:

- ▶ Puppet scales horizontally quite well
- ▶ The NFS filer underneath ... does not

NFS is used to share configurations and Puppet code between different masters.  
All the masters used to mount the same shared folder ...



# Clusters and Pools



- ▶ Catalog compilation time  $\sim 90\text{sec}$
- ▶ Catalog compilation per minute  
 $\sim 180$
- ▶ Batch  $\sim 300$  cores
- ▶ Interactive  $\sim 12$  cores



## Few concepts

- ▶ Modules (~ 280)

The various modules available should be viewed as a library that your hostgroup code can reuse.

- ▶ Hostgroups (~ 160)

Groups of nodes that are part of the same service and have some configurations in common.

- ▶ Environments (~ 180)

Collections of modules and hostgroups at different development levels.



# Environments allow us to ...

Environment "production" → All modules/hg from "master" branch

Environment "qa" → All modules/hg from "qa" branch

Custom environments (for testing purpose):

- ▶ Possibility to set a default branch
- ▶ Specify specific branch for one or more modules/hostgroups

```
$ cat ai321.yaml
---
default: qa
notifications: bob@cern.ch
overrides:
  modules:
    sssd: ai456
```

```
$ cat snap1.yaml
# Snapshot created on 2014-03-03 14:25:37.150312 based on production
---
notifications: bob@cern.ch
overrides:
  common:
    hieradata: commit/fb96070c9c77cc442ac60ba273768f547d376c17
    site: commit/fb96070c9c77cc442ac60ba273768f547d376c17
  hostgroups:
    adcmon: commit/8bf3ca9fe39a6f354dfc70377205ed806d6ae540
    [...]
  modules:
    abrt: commit/580cdbcf154dec2fa9ae717f2f55a18abbaebd72
    [...]
```



# Manage changes

Three important concepts:

- ▶ Modules
- ▶ Hostgroups
- ▶ Environments

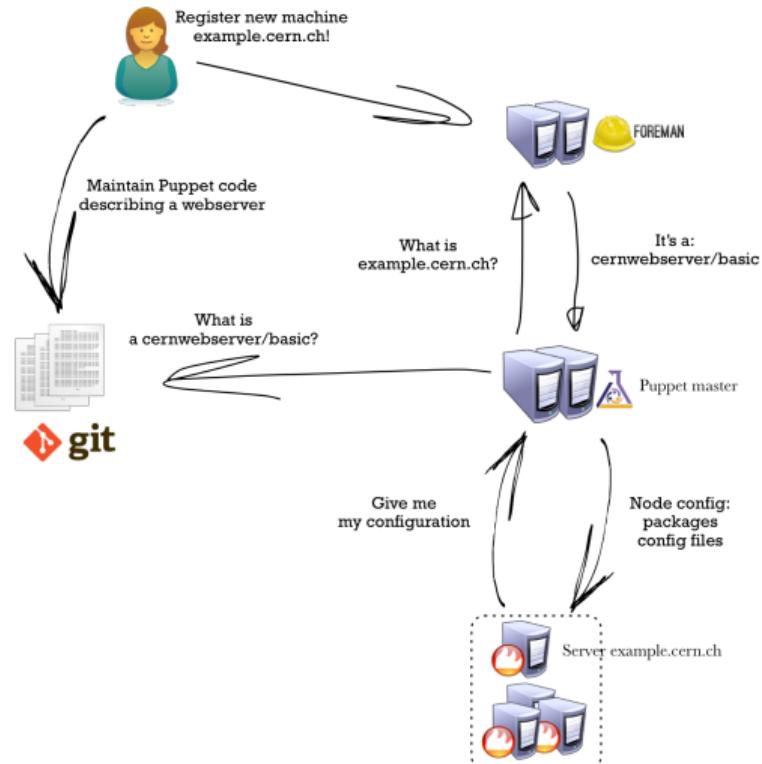
A configuration change has to be approved through a request in Jira.

Every git repo has at least two branches:

- ▶ master
- ▶ qa



# Puppet Run



## Jens

Jens creates Puppet environments for the Puppet Masters

- ▶ Using repository metadata and a list of environments definitions
- ▶ Allows dynamic environments and isolates puppet code for different services

Has recently been opensourced on GitHub:

<https://github.com/cernops/jens>

Useful for those running different services under the same puppet infrastructure



# Configuration Change Process

Configuration change process:

- ▶ Modify a module on feature branch
- ▶ Create a custom env and test the module
- ▶ Open a ticket on Jira and announce the change
- ▶ Merge to qa
- ▶ After one week, merge to production

Service managers use the same module for different services: we need to be sure that all the service managers are happy with the change before merging it to production.



# Jenkins and Continuous Integration process

- ▶ Machines are built and tested before merging a change to production
- ▶ More automation, less manual work
- ▶ Still work in progress, but looks promising



# Dashboard

<p>Changes in QA <b>28</b></p> <p>Last updated at 13:50</p>	<p>Changes currently in QA <b>CRM-1037</b> "Deploy new version of lemon-sensor-tw"</p>	<p>Changes recently deployed to Prod <b>CRM-1051</b> "cvmfs - Set CVMFS_USE_GEOAPI='no' for *.cern.ch repositories."</p>		
<p>CentOS7 Basic Build <b>Passed</b></p> <p>Ability to build a basic CentOS node in QA Last updated at 14:00</p>	<p>SLC6 Basic Build <b>Passed</b></p> <p>Ability to build a basic SLC6 node in QA Last updated at 14:00</p>	<p>SSO Apache Build <b>Passed</b></p> <p>Ability to build and access an SSO enabled web-server in QA Last updated at 14:00</p>	<p>MySQL Build <b>Passed</b></p> <p>Ability to build and access a MySQL server in QA Last updated at 14:00</p>	<p>PostgreSQL Build <b>Passed</b></p> <p>Ability to build and access a PostgreSQL server in QA Last updated at 14:00</p>
<p>LANDB Facts Config <b>Passed</b></p> <p>Ability to configure LANDB facts in QA Last updated at 14:00</p>	<p>Cert Manager Config <b>Passed</b></p> <p>Ability to configure Cert Manager Client in QA Last updated at 14:00</p>	<p>OSRepos Config <b>Passed</b></p> <p>Ability to configure special yum repos in QA Last updated at 14:00</p>	<p>AFS Config <b>Passed</b></p> <p>Ability to configure AFS in QA Last updated at 14:00</p>	<p>Firewall Config <b>Passed</b></p> <p>Ability to configure iptables in QA Last updated at 14:00</p>



## Automating procedures - RunDeck



- ▶ Tedious prone-error tasks replaced by executable code
- ▶ Handing off operational tasks to others
- ▶ Procedures as a list of individual and atomic steps
- ▶ Ability to react to failures



# Renaming hosts

1. Print info	renaming host peterparker.cern.ch to tonystark.cern.ch
2. Verify old hostname in LanDB	Looking up peterparker.cern.ch
3. Verify new hostname absent in LanDB	Looking up tonystark.cern.ch
4. Verify machine is physical	Looking up peterparker.cern.ch
5. Verify machine is in Foreman	Looking up peterparker.cern.ch
6. Rename in Foreman	Renaming peterparker.cern.ch to peterparker.cern.ch in Foreman
6. Rename in Foreman	peterparker.cern.ch found in Foreman
	Renaming host in Foreman...
7. Deactivate in PuppetDB	peterparker.cern.ch was renamed to tonystark.cern.ch in Foreman
7. Deactivate in PuppetDB	Deactivating peterparker.cern.ch in PuppetDB
8. Delete from AIMS	Submitted 'deactivate node' for peterparker.cern.ch with UUID 7a
8. Delete from AIMS	Deleting peterparker.cern.ch from AIMS
	Aims2client failed (No matching hosts found for peterparker.)
9. Rename in LanDB	peterparker-gigeth found in AIMS
9. Rename in LanDB	Renaming peterparker.cern.ch to tonystark.cern.ch in LanDB
	peterparker.cern.ch was renamed to tonystark.cern.ch in LanDB



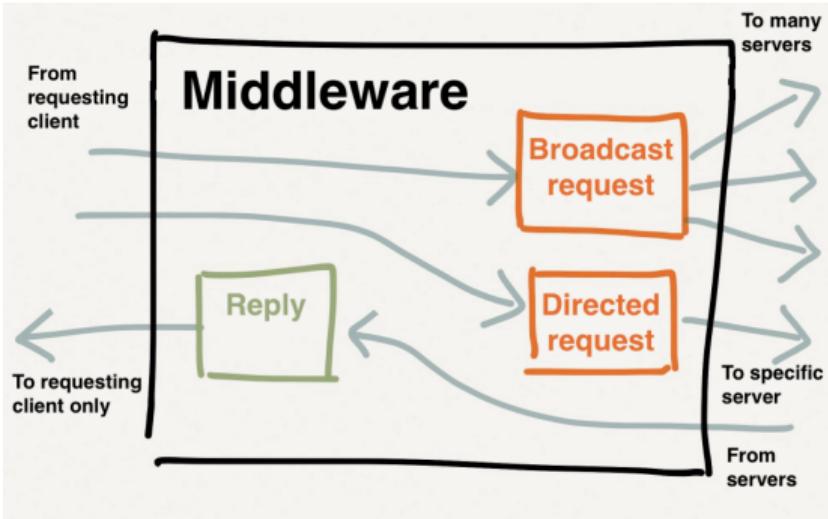
# Mcollective

Framework for server orchestration and parallel job execution

Problems in the past with big clusters  
>> 3000 nodes

Latest improvements:

- ▶ Direct addressing
- ▶ New PuppetDB discovery method
- ▶ Threaded Mode
- ▶ Batched requests



# Configuration Drifts

Configuration drifts started to be a problem:

- ▶ Out of sync machines
- ▶ Possibility for service managers to have snapshots
- ▶ Possibility to freeze their environment

It's not easy to keep all the configuration in sync



# Package Inventory

Centralized service for package inventory:

- ▶ Using Elasticsearch
- ▶ Queryable using Cli
- ▶ Compare a set of hosts
- ▶ Reports differences and misalignments
- ▶ Package History

```
+ cli git:(master) ✘ python cli.py -m 'pkgrep-test1 pkgrep-test2' compare
+-----+-----+-----+-----+
| Package | Field | pkgrep-test1 | pkgrep-test2 |
+-----+-----+-----+-----+
| htop    |        | Present     | Not present   |
| python-boto | epoch | 1.el6      | 4.el6        |
| python-boto | version | 2.38.0    | 2.34.0       |
+-----+-----+-----+-----+
```



# Conclusions

Moving from a traditional infrastructure to an Agile one allowed us to:

- ▶ Optimize our resources
- ▶ Speed up the development cycle
- ▶ Reduce interventions time
- ▶ Have more free time :)



# Conclusions

Puppet gives us the right combination between elasticity and efficiency

- ▶ Big community
- ▶ Active development
- ▶ Highly customizable



