

# Performance Testing and Monitoring

## A Guided Tour Through CLIF Tools and Features



Bruno Dillenseger  
Orange Labs

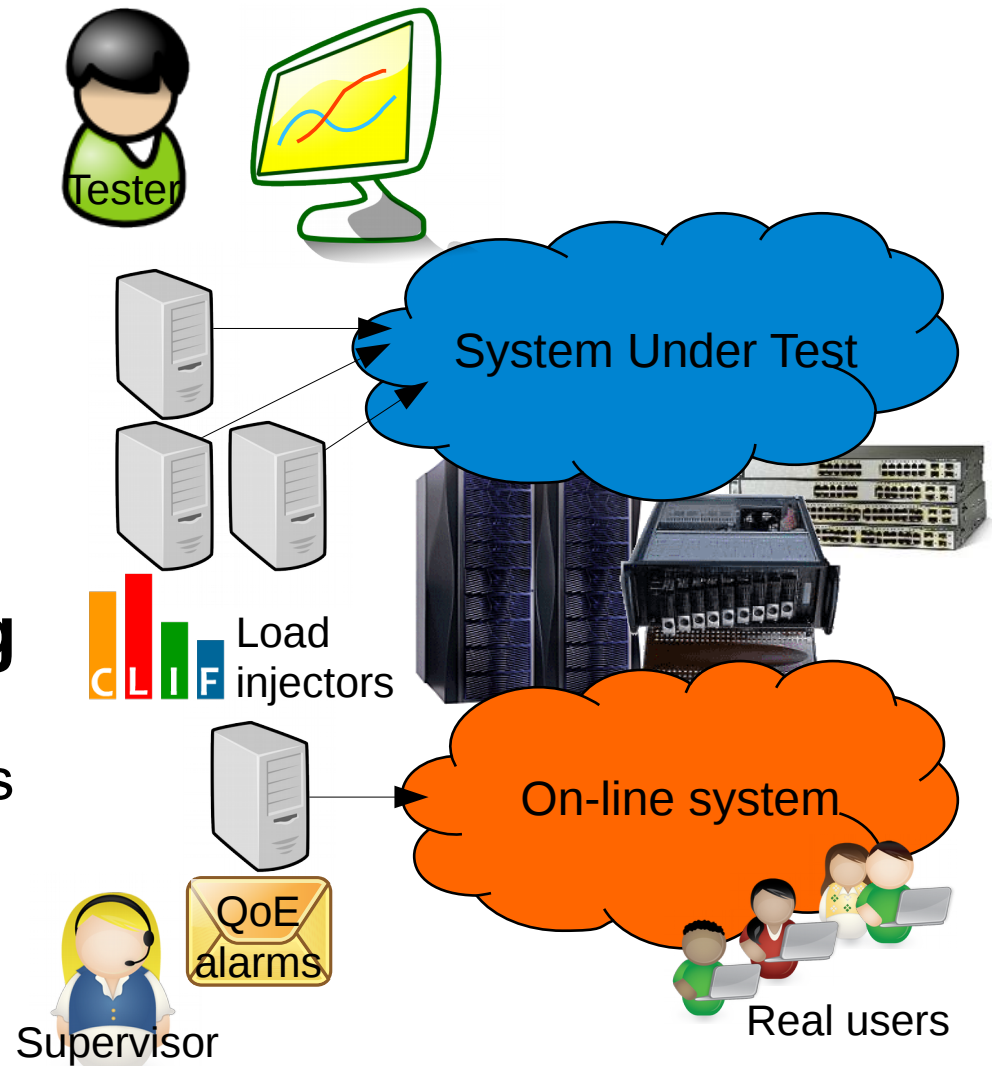
# Performance testing and monitoring

## → Performance testing

- emulate real user workload with load injectors and
- check the behavior of a system under test through selected metrics

## → Performance monitoring

- generate a light traffic on an on-line system with real users
- check metrics about user Quality of Experience.



# CLIF in a nutshell

- ➔ A generic and adaptable Java framework for distributed performance testing and monitoring
  - traffic generators measuring response times and throughput
    - supported protocols: TCP, UDP, DNS, FTP, GIT, HTTP, IMAP, JDBC, JMS, LDAP, MQTT, RTP, SIP, SVN, TCP
    - write/wrap your own protocols
    - write/wrap your own data set providers
  - probes measuring resources usage
    - CPU, disk, RAM, network, JVM, RTP
    - write/wrap your own probes
- ➔ High power
  - up to 1000 load injectors in parallel X millions of virtual users...
  - ... but also OK with 1 virtual user on a single load injector

# Happy 15th birthday to CLIF!

→ CLIF has been developed, used and adapted following trends of computing science R&D

- Component-Based Software Engineering
  - component-based distributed architecture based on OW2 Fractal model
- Java technologies
  - CLIF is pure Java, CLIF's main GUI is based on Eclipse IDE
- Autonomic computing
  - control loops for self-driven test campaigns (CLIF/Selfbench module)
- Cloud Computing for on-demand testing infrastructures
  - OpenStack, OW2 ProActive Scheduling & Workflows
- Automation and containers
  - CLIF plug-in for Jenkins
  - CLIF Docker image



# Main CLIF modules/distributions

- CLIF server
  - basic CLIF runtime with full-fledged command line interface
  - required wherever you intend to run a load injector or a probe
- CLIF swingGui
  - CLIF server enhanced with a simplified GUI for running tests and building custom performance reports
- CLIF Eclipse-based console (main GUI)
  - full-fledged GUI for writing and running complex test scenarios
  - IDE for extending CLIF itself
    - e.g. Eclipse wizard for writing custom plug-ins for load injection
- CLIF plug-in for Jenkins
  - automates test runs, performance reporting, QoE alerting

# Installing and using CLIF

- Download a distribution from [clif.ow2.org](http://clif.ow2.org)
  - unzip
  - dependency: java 8+ runtime
- Get **CLIF Performance Testing** plug-in for Jenkins
  - Manage Jenkins > Manage Plugins > Available
- Get container image from Docker hub
  - `docker pull dillense/clif`
- Get support from [clif.ow2.org](http://clif.ow2.org)
  - user guide, javadoc, tutorials, videos, examples, presentations...
  - e-mail [clif@ow2.org](mailto:clif@ow2.org)

# Command line interface

## clifcmd ...

- **analyze**
- **change** testplan\_name id param\_name param\_value
- **collect** testplan\_name [id1:id2:...idN]
- **config** [registry\_host[:registry\_port] [codeserver\_host[:codeserver\_port]]]
- **deploy** testplan\_name testplan\_file
- **gui**\*
- **help**
- **init** testplan\_name testrun\_id
- **join** testplan\_name [id1:id2:...idN]
- **launch** testplan\_name testplan\_file testrun\_id

\* this command is available only with distribution clif-swingGui

- **listservers** [test plan file names...]
- **params** testplan\_name id
- **probehelp** probe\_type
- **quickstats** [report\_directory]
- **registry**
- **resume** testplan\_name [id1:id2:...idN]
- **run** testplan\_name testrun\_id [id1:id2:...idN]
- **server** [name]
- **start** testplan\_name [id1:id2:...idN]
- **stop** testplan\_name [id1:id2:...idN]
- **suspend** testplan\_name [id1:id2:...idN]
- **version**
- **waitservers** [testplan\_file]

**Available for Windows, MacOSX, Linux, etc.**

# Defining and running tests

The screenshot displays the Clif Perspective within the Eclipse IDE, titled "Clif Perspective - Dummy/dummy.ctp - Eclipse". The interface includes a menu bar (File, Edit, Navigate, Search, Project, Run, CLIF, Window, Help) and a toolbar with icons for file operations and development tools. On the left, the Navigator shows a project structure for "Dummy" with files like .project, clif.opts, dummy.ctp, dummy.props, dummy.xis, and various test-related files. The CLIF Tree at the bottom left shows a "local host" with an "injector 0".

The main workspace is divided into several panes. The "Test Commands" pane shows "Injectors and probes" with a table of injectors:

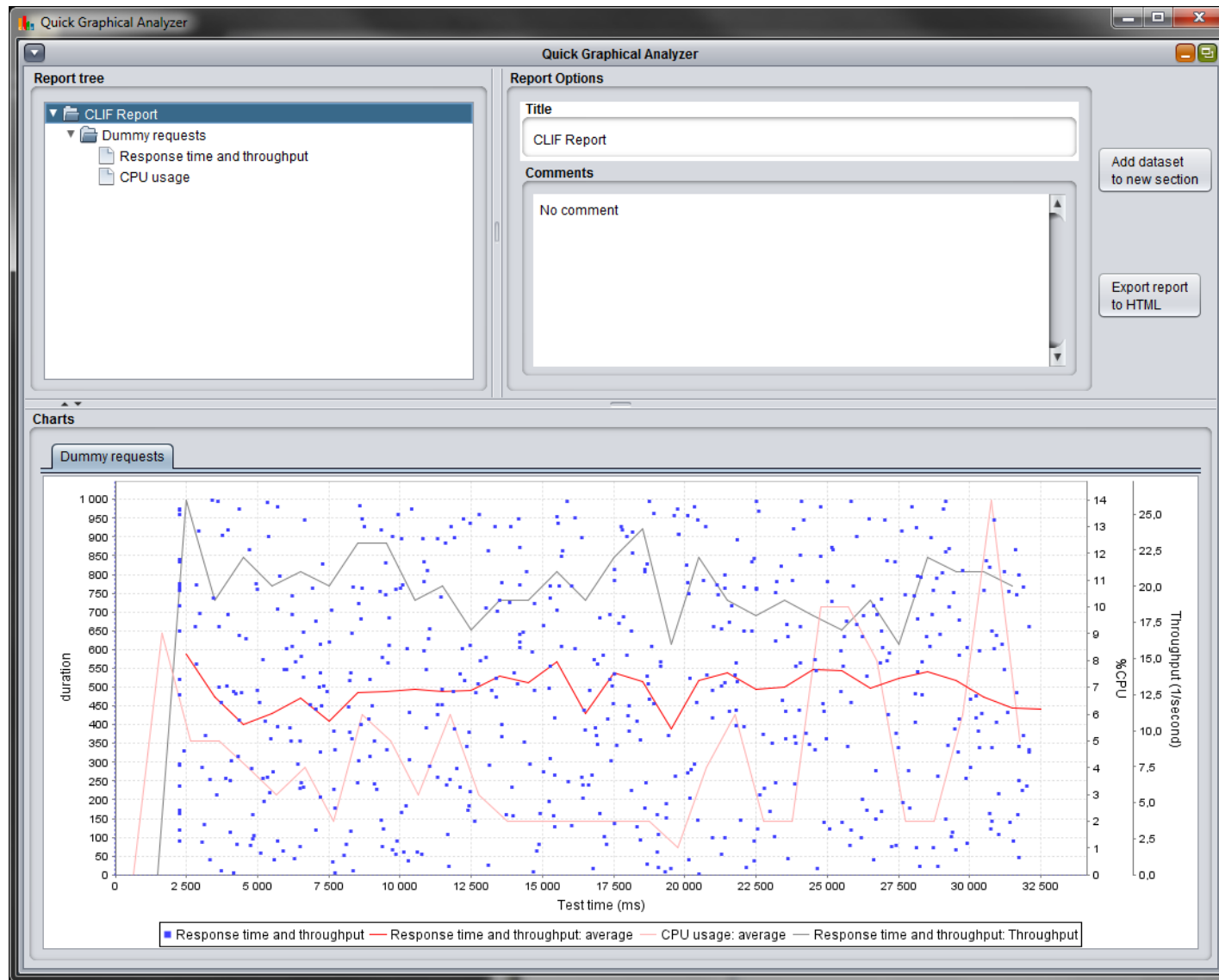
Id	Server	Role	Class	Arguments	Com...	State
0	local host	injector	IsacRunner	dummy.xis		completed

Buttons for "Select All", "Deselect All", and "Global state: complete" are visible. Below the table are buttons for "Initialize", "Start", "Suspend", "Stop", "Collect", and "Parameters".

The "Monitor" pane shows a graph of "action throughput (actions/s)" over time. The graph has a y-axis from 10 to 23 and an x-axis with time markers from 32" to 04'10". A blue line represents the throughput, which rises sharply and then levels off. The "Drawing time frame (s)" is set to 300, and the "Polling period (s)" is set to 1. Buttons for "Refresh" and "Reset" are at the bottom.



# Custom performance report



# CLIF plug-in for Jenkins: detailed performance report

## Build Performance Report






calcupd20x2 - Tue Jan 24 15:33:21 CET 2017

Server - clif2

Probe - jvm ( Argument=1000 60 Comment= )

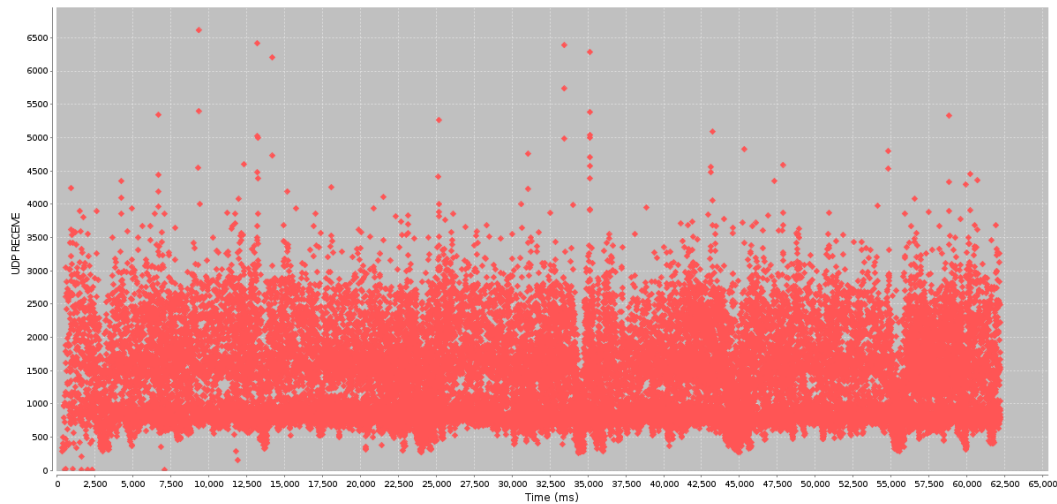
Label	Samples	Average	Median	Min	Max	Standard deviation
<a href="#">free memory (MB)</a>	61	131	131	119	146	7.75
<a href="#">used memory %</a>	61	15	16	4	23	5.14
<a href="#">free usable memory %</a>	61	89	90	84	97	3.38

Injector - inj2 - IsacRunner ( Argument=calcupd20.xis Comment= )

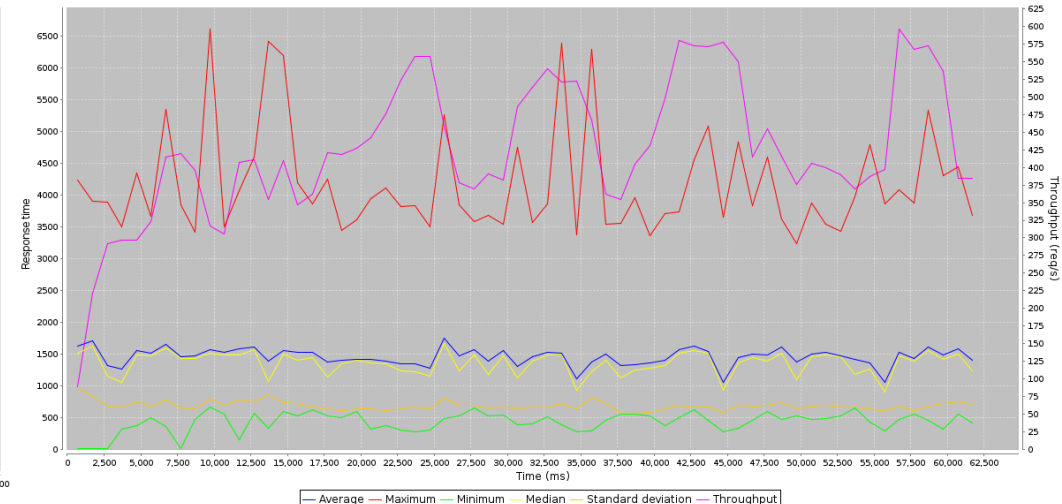
Type	Requests	Success	Errors	Average	Median	Min	Max	Standard deviation	Throughput	Error rate
UDP CONNECT 	20	20	0	36	36	23	66	11.38	0.32	0%
UDP RECEIVE 	8427	8,265	162	1429	1060	8	136096	3238.21	133.64	1.92%
computation error 	883	0	883	0	0	0	0		0	100%
UDP SEND 	8427	8,427	0	36	35	7	1206	35.26	136.26	0%

# CLIF plug-in for Jenkins: performance report graphs

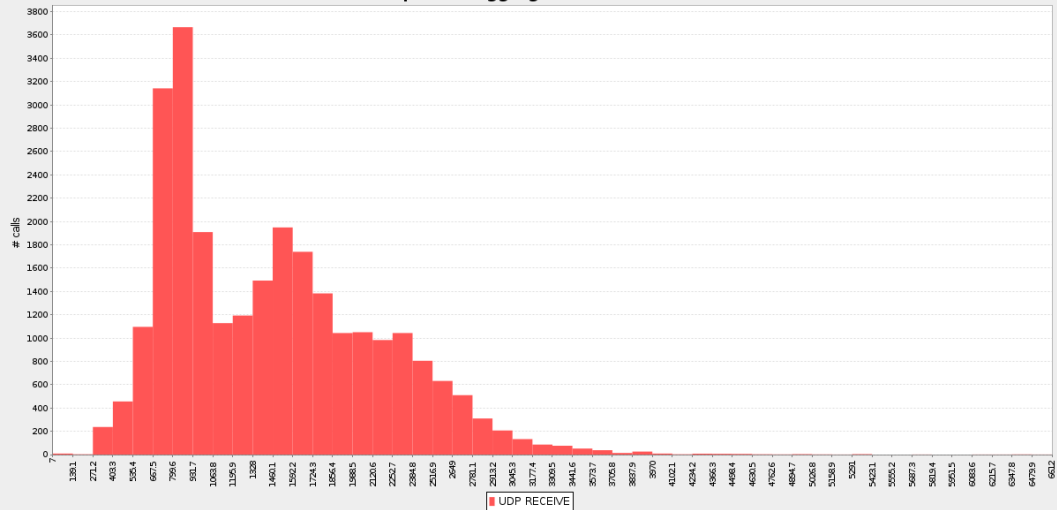
calcdp20x2 - aggregated - UDP RECEIVE



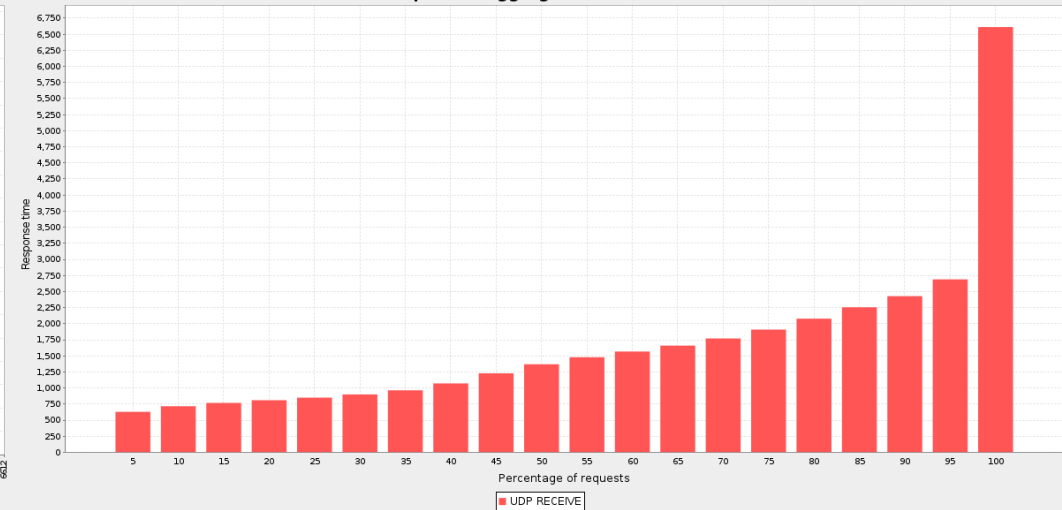
calcdp20x2 - aggregated - UDP RECEIVE Moving Statistical Period (1s)



calcdp20x2 - aggregated - UDP RECEIVE



calcdp20x2 - aggregated - UDP RECEIVE

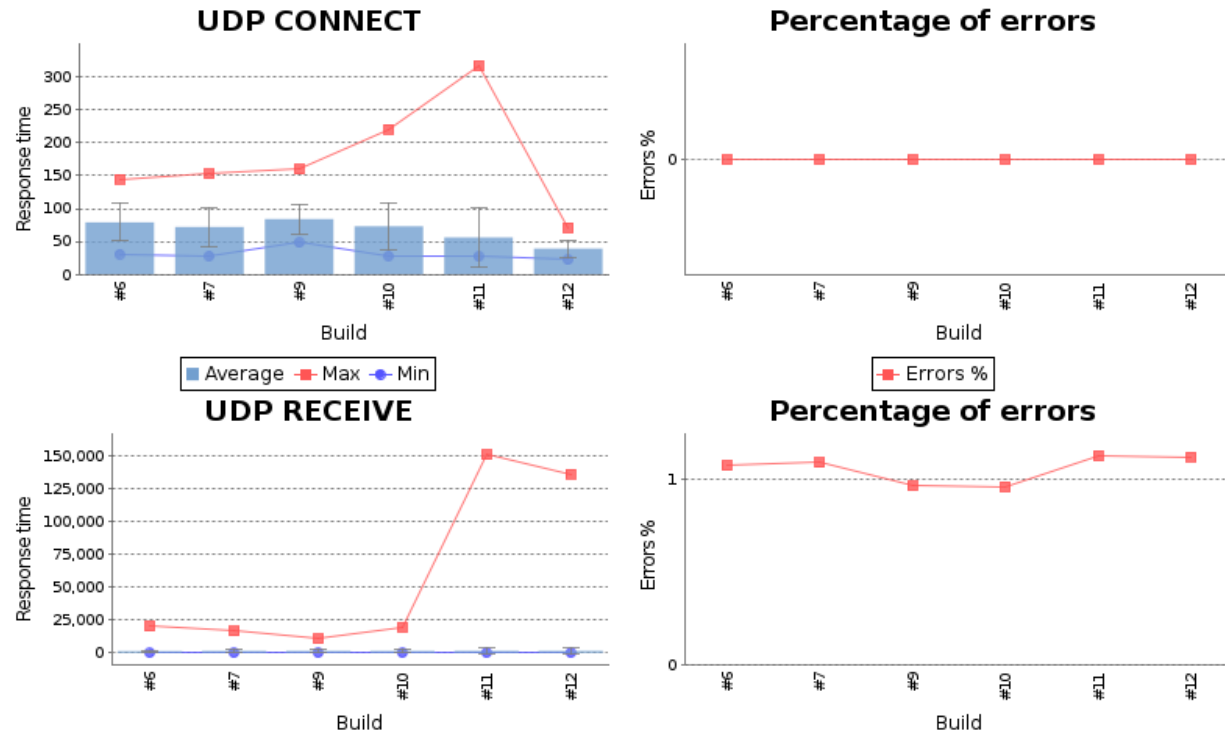


# CLIF plug-in for Jenkins: performance trend

## Clif Performance Trend Report





[Last Report](#)

calcup20x2



# What's coming next?

## → More automation for users

- OW2 ProActive-based deployment over Metal as a Service 
  - collaboration with ActiveEon 
- Ansible roles and playbooks 
- OpenStack environment 

## → Evolutions for developers

- move to OW2's Gitlab instance...
- ... coming with big code base reorganization
- Continuous Integration: Bamboo to be replaced by Gitlab CI
- move from "legacy" CLIF to "ProActive CLIF"

## → Your Questions?