GO-Docker: Batch scheduling with containers

Olivier Sallou, Cyril Monjeaud

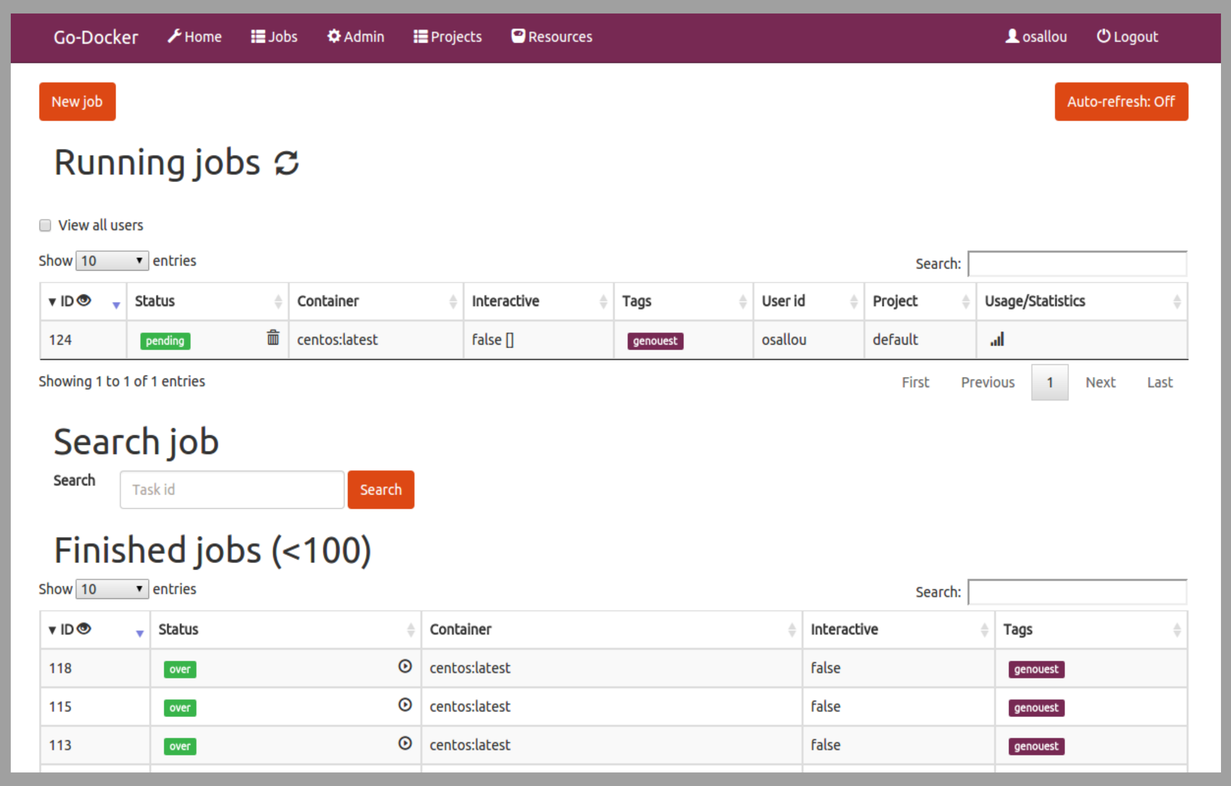
GO-Docker: Batch scheduling with containers

Olivier Sallou, Cyril Monjeaud

https://bitbucket.org/osallou/go-docker/
http://www.genouest.org/godocker/index.html

Jobs in containers

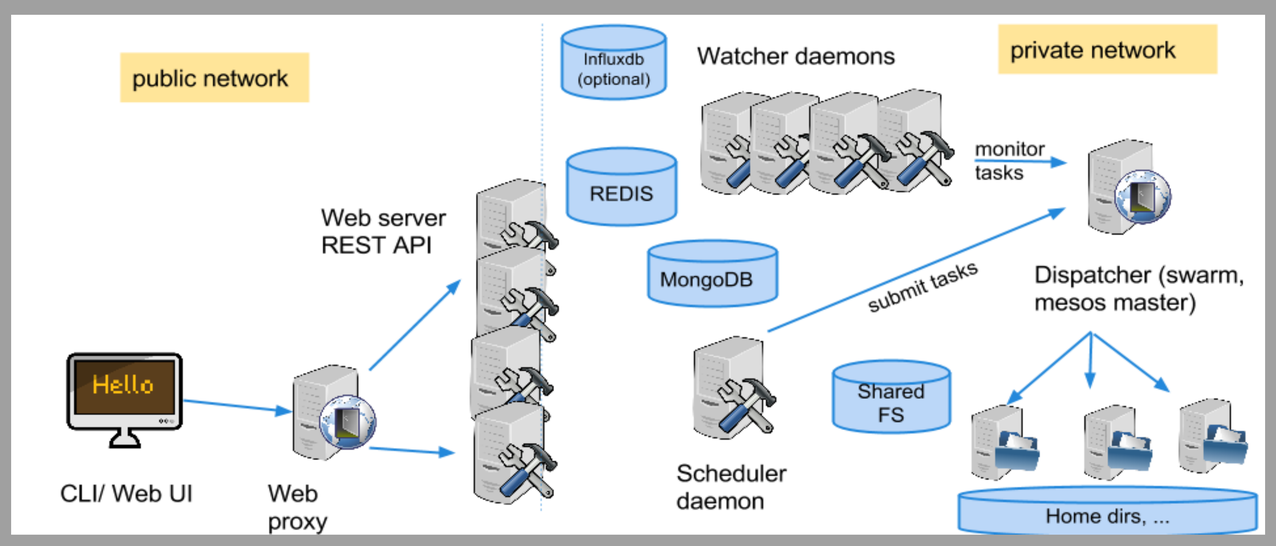
Go-Docker is a batch scheduling job manager.
It executes jobs in Docker containers, providing security/isolation of resources and access to a software catalog and multiple operating systems.
For multi-user systems, it mounts user directories in the container and schedules jobs according to user requirements (cpu, memory, queues).
Quota and priorities are managed at user and group level.
Software is in beta-test at GenOuest BioInformatics core facility, before production, with Apache Mesos.



Jobs can be managed via the web interface, the REST API or the CLI tool

Plugins

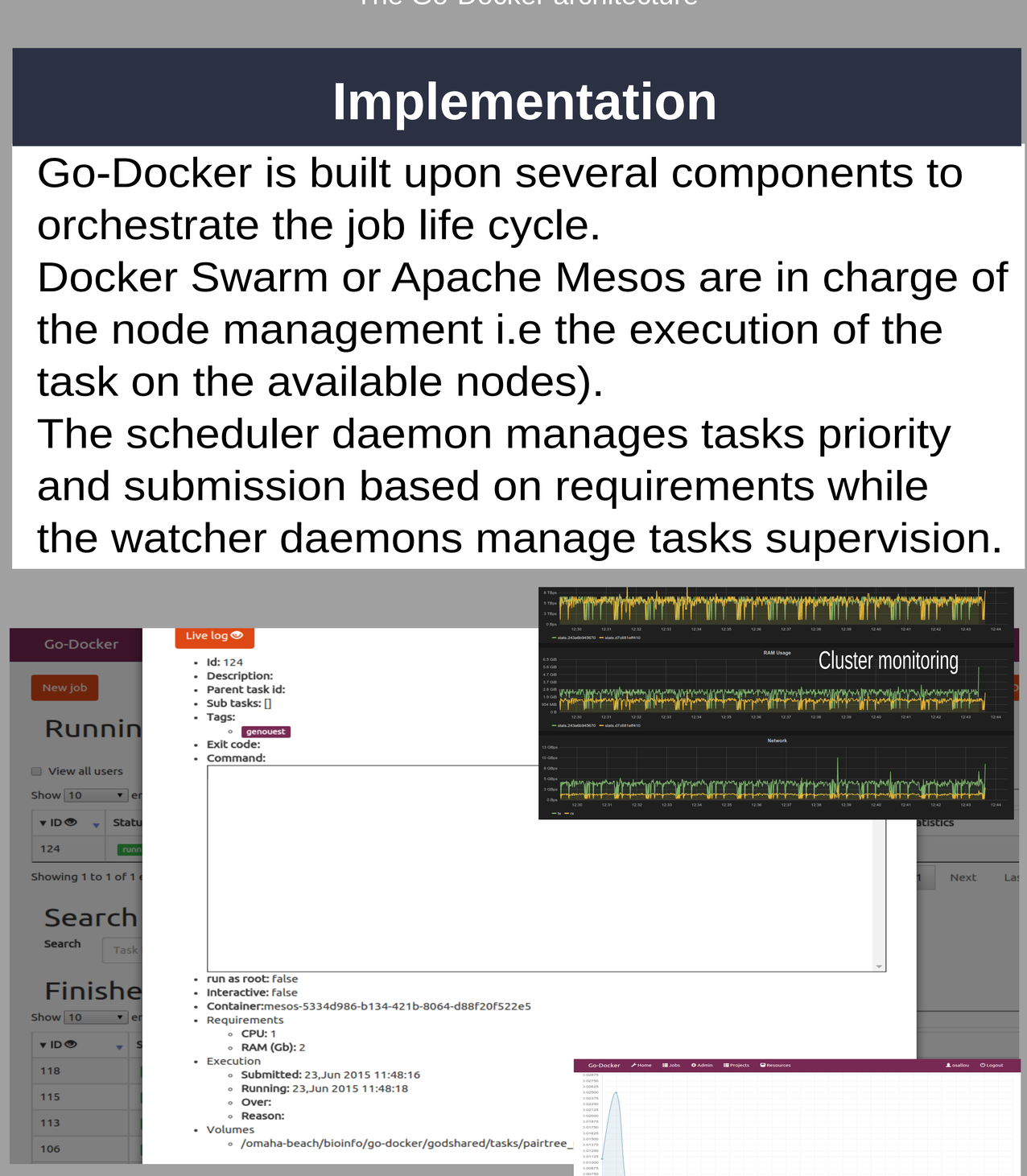
The plugin architecture gives the possibility to add new authentication/authorization scripts, different scheduler algorithms or execution systems. It can also be extended with "watchers" to monitor job life cycle (kill after X days for example).



The Go-Docker architecture

Implementation

Go-Docker is built upon several components to orchestrate the job life cycle.
Docker Swarm or Apache Mesos are in charge of the node management i.e the execution of the task on the available nodes).
The scheduler daemon manages tasks priority and submission based on requirements while the watcher daemons manage tasks supervision.



Jobs have resource requirements and shared volumes
Resources can be monitored with live data.

Scalability

Go-Docker processes and databases are horizontally scalable to scale with jobs load.
The single point of failure is the scheduler, a single process to order all pending jobs according to quota and user/projects priorities.

Scalability and cloud setup will be tested soon with an AWS Research grant in Education.

Conclusion

Go-Docker provides:
- secured job/interative session control with isolation
- jobs live monitoring and past statistics
- optional root access to the container
- immediate access to differents operating systems and a growing catalog of software (Docker registry)
- easy container usage with no super-user rights
- Web UI, CLI, REST interface

INRIA, IRISA, GenOuest Core Facility, Campus de Beaulieu, 35042 Rennes, France
University of Rennes 1, Campus de Beaulieu, 35042 Rennes, France

GenOuest BioInformatics core facility: http://www.genouest.org

Contact: olivier.sallou@irisa.fr