



Enhancing Grid Infrastructures with  
Virtualization and Cloud Technologies

## **StratusLab Development, Certification and Release Procedures in Place**

Milestone MS7 (V1.0)  
10 September 2010

### **Abstract**

The StratusLab Development, Certification and Release Procedures are in place and have been successfully used to support StratusLab release v0.1.



StratusLab is co-funded by the  
European Community's Seventh  
Framework Programme (Capacities)  
Grant Agreement INFSO-RI-261552.



The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE STRATUSLAB COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2010, Members of the StratusLab collaboration: Centre National de la Recherche Scientifique, Universidad Complutense de Madrid, Greek Research and Technology Network S.A., SixSq Sàrl, Telefónica Investigación y Desarrollo SA, and The Provost Fellows and Scholars of the College of the Holy and Undivided Trinity of Queen Elizabeth Near Dublin.

This work is licensed under a Creative Commons Attribution 3.0 Unported License  
<http://creativecommons.org/licenses/by/3.0/>



## Contributors

Name	Partner	Sections
Marc-Eliañ Bégin	SixSq	All

## Document History

Version	Date	Comment
0.1	6 Dec. 2010	First draft for internal review.
1.0	10 Dec. 2010	Final draft.

# 1 Introduction

This report briefly describes the required procedures for the development, certification and release of the StratusLab distribution.

Most of the information for these procedures are documented in the project wiki. These procedures have been used successfully in order to release StratusLab v0.1. During the process, these procedures have evolved. Further, we are likely to improve and adapt these procedures as the project progresses.

An important aspect on which we are investing effort is in the automation of some of these procedures, such that they can be triggered automatically.

## 2 Development Procedures

StratusLab follows Scrum as its agile development process. We have chosen a cycle duration (called sprints in Scrum) of 3 weeks. We also have chosen to release publicly every two sprints, which means that our development process must produce production ready increments every two sprints.

A number of tools have been deployed as services and configured to support our development process, notably:

- Management, Bug and Progress Tracking: JIRA + GreenHopper <sup>1</sup>
- Continuous Integration: Hudson <sup>2</sup>
- Package repository: Yum Repositories <sup>3</sup>

To simplify credentials management for these tools, they are all connected to an LDAP server managed by LAL.

A ‘Bill of Rights and Obligations’<sup>4</sup> was also put in place to specify what is required of each project member before a technical task can be considered ‘done’. This includes package layout, testing activities, documentation, etc.

To build the packages that form the StratusLab distribution, we use the Maven tool<sup>5</sup> which is triggered by Hudson jobs. This means that all of the StratusLab project packages are built from neutral, well-configured, and well-controlled machines.

Git<sup>6</sup> is used for version control. As a standard policy, we try to avoid using branches and encourage development to be performed in small increments on the main master branch of the Git repositories.

We have two distinct Yum repositories containing the StratusLab packages composing the distribution: Snapshot and Release. The build and certification activities use the Snapshot repository while the release process uses the Release repository.

---

<sup>1</sup><http://www.atlassian.com/software/jira>

<sup>2</sup><http://hudson-ci.org/>

<sup>3</sup><http://yum.baseurl.org/>

<sup>4</sup><http://stratuslab.eu/doku.php?id=billofrightsandobligations>

<sup>5</sup><http://maven.apache.org/>

<sup>6</sup><http://git-scm.com/>

## 3 Certification Procedures

Before software can be released, it must be tested in representative conditions. For this purpose, a set of Hudson jobs are triggered which build from the ground-up the entire StratusLab distribution, including major components such as OpenNebula. The packages are then installed and configured using the manual and Quattor-based installation tools StratusLab ships. We deploy the StratusLab system in different configuration and following different deployment models, representative of the way it is intended to be deployed in production.

On top of these regular (at least daily) automated tasks, we also have a set of machines on which we can perform a clean manual installation following the latest online documentation and tutorials.

Once these steps have been successfully performed and monitored during the daily stand-up meetings, the project, including the technical group, decide if the software is declared ready for release.

## 4 Release Procedures

Once the StratusLab software is considered ready for release, Maven tasks, also integrated in Hudson, are used to trigger the generation of release packages. The software corresponding to these packages is then tagged in Git, renamed and re-generated such that they have consistent and simple names using a simple release version increment scheme. The resulting packages are then published to the public Yum repository such that end-users will be able to access them using standard package management software.

Release notes are also included so that users better understand the new features, any known problems and workarounds for the specific version released.

The updated documentation is also made public (e.g. tutorial and training).

From this point, the dissemination activities announce the availability of the release.

The reference infrastructures are also upgraded to the new release, such that users can also take advantage of new features and bug fixes without having to deploy their own infrastructure.