Secure Provisioning of Cloud Services
based on SLA Management

SPECS Project - Deliverable 7.1.2

Communication tools, project information package and control procedures

Version 1.0
30 April 2016

The activities reported in this deliverable are partially supported
by the European Community's Seventh Framework Programme under grant agreement no. 610795.

Deliverable information

Deliverable no.:	D 7.1.2
Deliverable title:	Communication tools, project information package and control procedures: Public project site, flyers, internal and external wiki. Document templates. Organization of the knowledge base. Timetable and checklist. Parameters to be monitored. Includes quality and risk management plan
Deliverable nature:	Report
Dissemination level:	Public
Contractual delivery:	30 April 2016
Actual delivery date:	30 April 2016
Author(s):	Valentina Casola (CerICT)
Contributors:	Dana Petcu (IeAT), Silviu Panica (IeAT)
Reviewers:	Massimiliano Rak (CeRICT)
Task contributing to the deliverable:	T7.1, T7.2, T7.3
Total number of pages:	50

Executive summary

The Deliverable 7.1.2 is the second deliverable focused on communication tools, project information package and control procedures. In particular, D7.1.1, was setup at start of the project, and it described the set up of communication tools, the definition of control procedures and tools to monitor the activities, together with Quality and Risk Management plans. Deliverable D7.1.2 describes the changes and updates applied to the above tools during the project and their possible use after the end of the project.
It is worth noticing that some of the tools were updated in order to meet dissemination KPIs and provide a more effective internal and external communication.
This deliverable focuses on the description of renewed dissemination tools and, in particular, to those supporting the sw development activities, during and after the project.
Section 2, describes the Project Information Package, with a special focus on the renewed project web site. Moreover, it reports the social media used and the set of flyers and posters produced.
Section 2 focuses on the SPECS Open Source Software repository. More in details, it reports about the SPECS-team account on bitbucket repository, how it is structured, how it is possible to install and use the SPECS software. The SPECS repository will be available after the end of the project and will be maintained by the owners of the software modules.
Section 3 focuses on the maintenance of the software repository after the end of the project.
Section 4, instead, focuses on maintenance of the archives of communication tools and of the project information package after the end of the project.
The deliverable ends with Section 5, which summarizes the conclusions on the tools and software solution adopted during the project.

Table of contents

Index of figures

Index of tables

1. Introduction

Deliverables D7.1.1 and D7.1.2 are related to Tasks T7.1, T7.2 and T7.3 and dedicated to the description of all tools that support the management, scientific and dissemination activities. The first version of this deliverable D7.1.1 outlined the SPECS project management methodology and included the planning of scientific and management activities, the procedures to monitor and control the effective advance of all project activities and also the quality control procedures.

To support communication among all SPECS partners and to promote the ongoing activities and project results, the following communication tools have been set up:

- A Project information package, including a Content Management System (CMS) for implementing the official web site and different social media;
- A software repository, to share the open source code is available;
- A Wiki system, to enable partners to share, manage, and download documents and information (reported in D7.1.1);
- Monitoring and Control procedures tools, to monitor the advances of the project and the development activities;

The goal of these tools has been devoted to internal and external communication and dissemination.
This deliverable is dedicated to the description of all tools that support the sw development and dissemination activities. The Wiki System and the internal tools for control and monitoring were largely described in D7.1.1_ver1.1, they will not be reported in this deliverable.

In particular, in this deliverable, we will report about the renewed project web site (Section 2) describing its functionalities from the user perspectives and we will describe the SPECS software repository from the developer perspectives. In particular in Section 3 we will report about the SPECS-team bitbucket repository, how it is structured, how it is possible to install and use the SPECS software.
Furthermore, in Sections 4 and 5, we will report about SW repository maintenance and accessibility and data archives to be managed after the end of the project, with links and references.

2. Project Information Package

In order to pursue interactive communication, the SPECS project uses many web facilities. Since the start of the project a web site was published; in the following sections we illustrate the technical details and web presentation. Furthermore, the SPECS project exploits social media networks as they constitute a successful technique for involving users in different areas and can help provide a fast dissemination of the latest news about the project. Different KPIs have been defined, related to these dissemination tools, they are reported in WP6 related deliverables.

1.1 Project Public Web Site

The official SPECS project web site URL is: http://www.specs-project.eu/.

The project web site was created in June 2013 and completely renewed on November 2014. It is continuously updated in order to report all information related to the project advances and to the numerous initiatives of the involved communities.
The web site was implemented with WordPress, a free and open source blogging tool and content management system (CMS) (see http://wordpress.com/ for further details). Wordpress represents a flexible and powerful means to build websites, and allows for easily adding/modifying pages, incorporating multimedia content, and managing additional features such as newsletters and blogs. One of the most important features of Wordpress is the high availability of plug-ins, which offer custom functions and features so that users can tailor their site to their specific needs. Moreover, Wordpress provides a variety of support resources, represented by a wide documentation and an active community.

This section describes the website realized for the SPECS project. The website has been released since the beginning of the project in order to enable the communication on the project activities and provide an up-to-date list of events related to it. Moreover it provides the necessary background to understand project objectives and supports the dissemination of the project results.
The aim of the section is to provide the description of the website from the user perspective in order to give an overview of the possible interactions with it. Moreover some references to the technological platform at the basis of the website are provided in the next section, in order to better support the remaining discussion.

1.1.1 Technical background

The SPECS website is realized by means of a Content Management System (CMS) which is a well-know technique to used to manage the editing, the modification, and removal of contents from a website.
As described in the Deliverable 7.1.1 the adopted CMS is Wordpress (see http://wordpress.com/ for further details) which is an open source CMS born for personal blogging and growth as one of the most adopted platform for the realization of professional websites.

The wide adoption of Wordpress for professional websites is mainly due to the possibility of easily provide plugins for add new features highly customizable.
Next sections provides a deep description of the website sections, providing also some screenshot of the most interesting web pages.

1.1.2 Website Homepage

The homepage of the SPECS website is depicted in Figure 1.

Figure 1 - SPECS website Home page

According to the figure, three different sections can be identified in the homepage

1) The header, at the top of the page, containing the project logo and the menus for the access to the SEPCS sections. The detail of the header is provided in Figure 2.

The menus provided by the header are the same in each section of the website; besides the home, the other sections of the website are:

- Project, in which are provided the aims of the SPECS project;
- Events, which reports current and past events related to the dissemination of the project;
- Security SLA, containing the description of the service level agreements related to the project activities;
- Publications which provides an up-to-date list of published papers in peer-reviewed conferences and journals and deliverables;
- Media which contains all the materials related to the public communication of the project
- Resources containing descriptions of the SPECS platform and the links to the repositories;
- Portfolio which provides the description of some key technological solutions, involved in the SPECS project

In the next paragraphs each of the above-mentioned sections are discussed in detail.

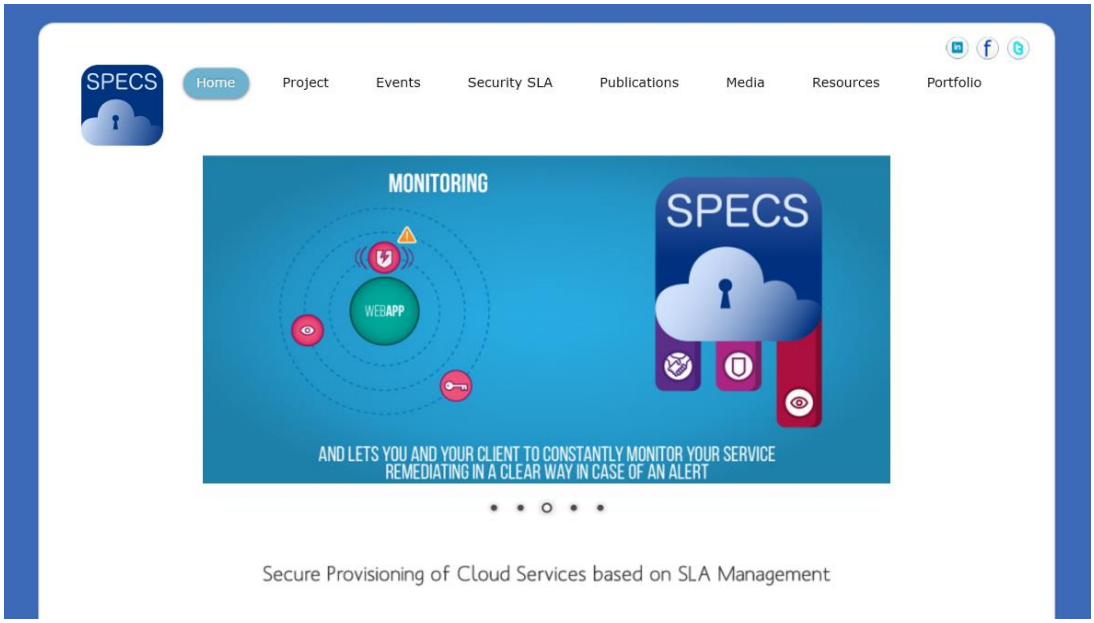

Figure 2 - SPECS website header

2) The central part of the main page contains the advertisements. In fact, the left side of the Figure 3 provides the latest news related to both public events, as the conferences and workshops and project activities, as surveys among partners or demos.
The right-side is instead related to the rss feeds and social network communication.
According to this, an rss reader has been inserted by means of Wordpress plugin. Such rss reader collects the updated from the portal of the FP7 program of the European Community, from the website of the EMC2 partner and the blog of the Cloud Computing Alliance (CSA).
Moreover, the right-side part of the homepage, provides a scrollable windows, based on a Wordpress plugin and showing messages coming from the Twitter account of the Specs project (https://twitter.com/FP7SPECS).

Figure 3 - SPECS website home page Central section

3) The last section, the footer, is depicted in Figure 4. As the header, it is present in each section of the website. The footer contains the logos of the SPECS partners (see Project section for further details) and in particular of the FP7 program of the European Commission with the related grant agreement.

Figure 4- SPECS home page footer

Moreover the footer provides the links to the social networks pages of Linkedin, Facebook and Twitter and the link to access to the web form for contact the SPECS organization.
A detail of the contact form is provided in Figure 5. The contact form provides the necessary fields to fill in order to send a specific request to the SPECS organization and provides to it the user details of the response. In addition, as in the home page, a reduced version of the latest news is provided in the right-side of the page.

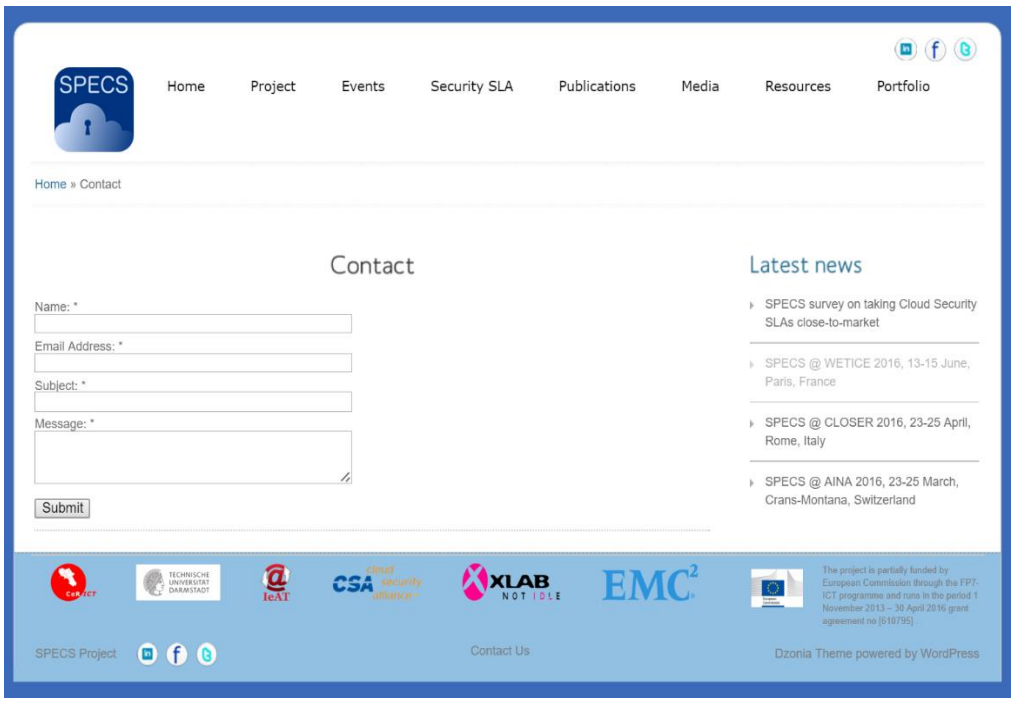

Figure 5 - SPECS website contact form

1.1.3 Project Section

The project Section summarizes some relevant information about the SPECS project activities. In particular it is composed by the following sub-sections:

- Description which provides a general description SPECS projects aims and describes the high-level architecture of the framework that has been developed during the project activities.
  The description webpage is depicted in Figure 6.

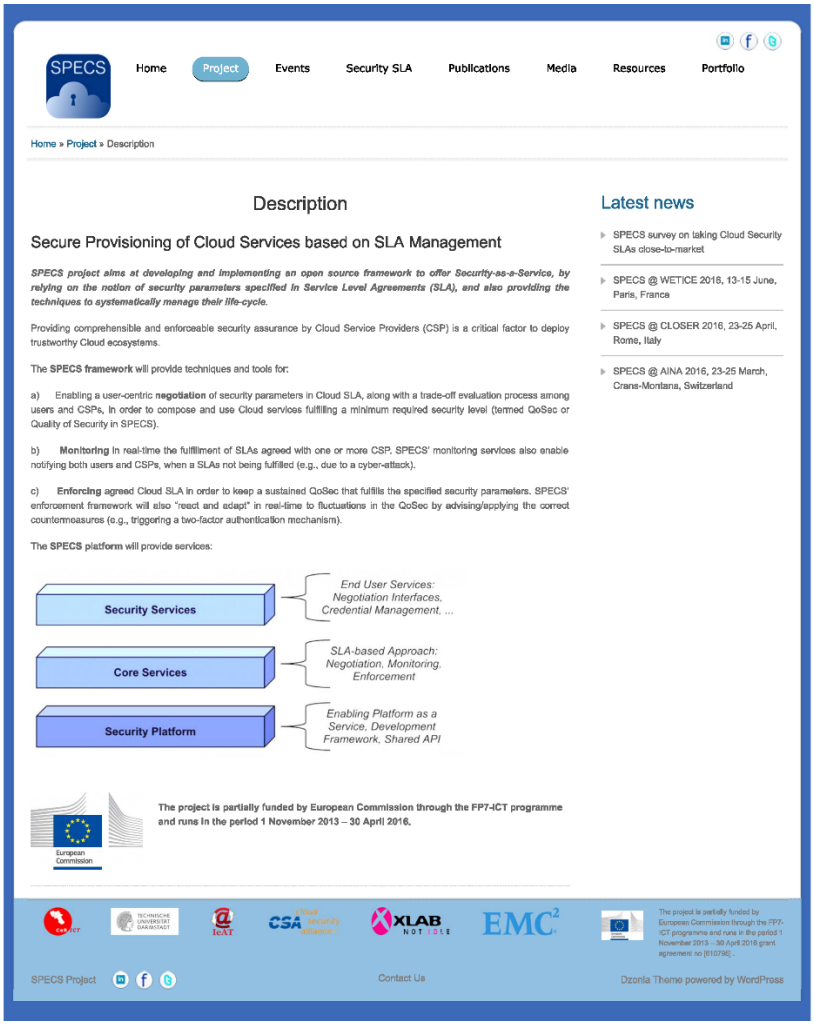

Figure 6- SPECS website Project Section web page

- Partners: this section provides a brief description of the partners involved in the SPECS project activities and the link referring to their website pages. A deep description of partners and the details about their tasks has been provided within the specification documents of the project.

- Timeline: the timeline is realized exploiting a Wordpress plugin and summarizes the main milestones of the SPECS framework development.
- Workpackages, which reports the task to be performed during the project as reported in the specification of the project.

1.1.4 Events

Events is one of the key sections of the SPECS website, since it provides a continuously up-to-date list of SPECS project events and other international forums strictly related to the SPECS topics.
The Events section is a reference point for anyone is interested to follow or to know more about SPECS activities and about their achieved results. It has a twofold objective: first it provides a summary of the past events related to SPECS or to cloud computing activities partially addressed by SPECS project activities. Second, it provides the announcements of the next events by providing the necessary information about the dates and of the locations in which the international meeting will be held.
This section is online since the beginning of the project and actually contains over 60 events among international conferences, workshops, forums and projects meeting for the advances of the SPECS project activities.
Each of the described events is a way to bring together experts from industries and academies working and interesting in potentialities of the Cloud Computing. Each event discusses principal research enhancement in Cloud Computing and future directions and opportunities for the utilisation of Cloud Computing technologies.
The SPECS events are a way to enable the Cloud communities to meet and discuss on innovative architecture and services and on the research effort needed for the development of new cloud platforms.

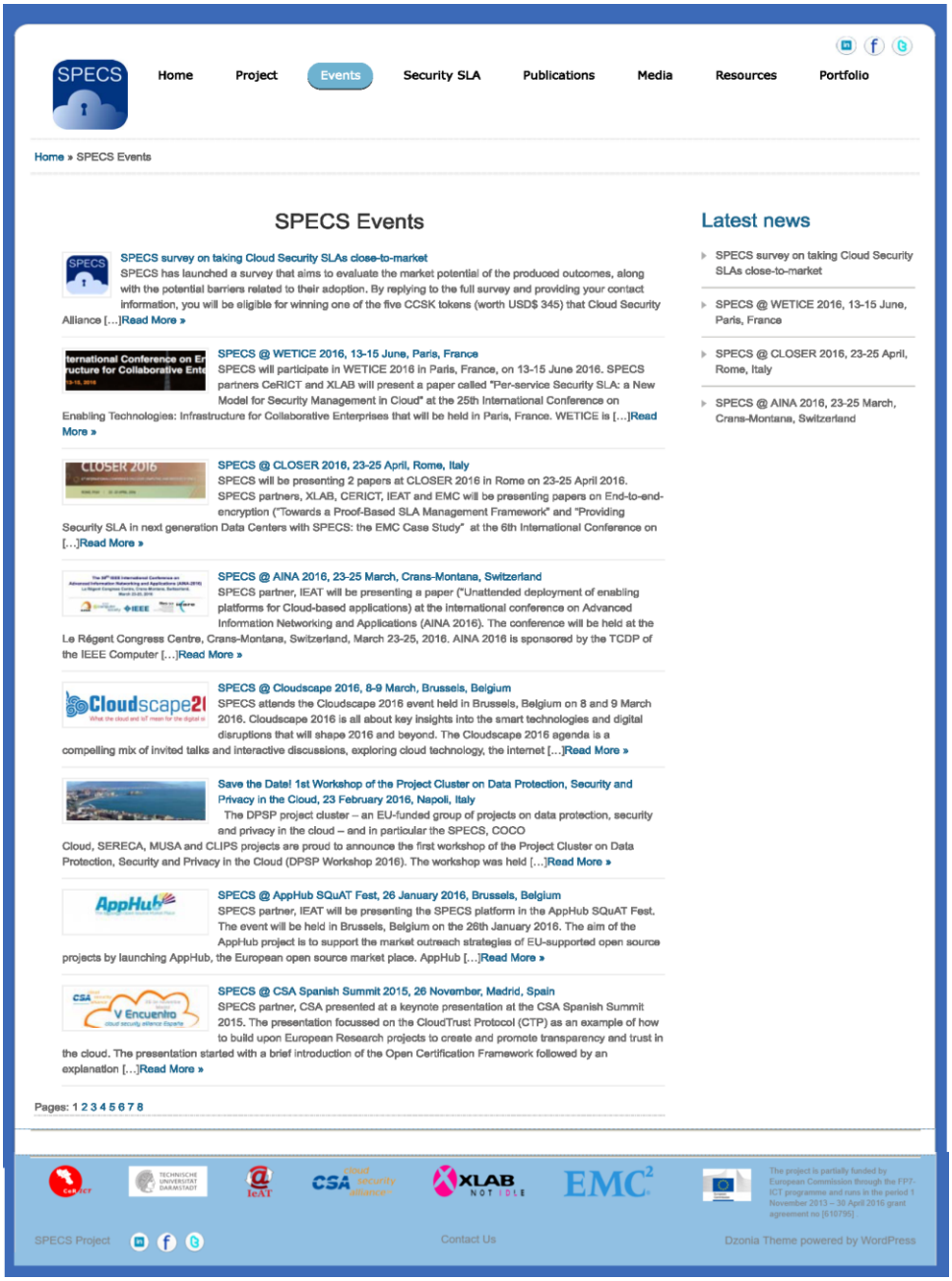

Figure 7- SPECS website Events section

1.1.5 Security SLA

The security SLA is a relevant sections since it provides the international standards to which the SPECS framework is compliant and related initiatives of the European Community strictly related to the SPECS project.
According to this the Security SLA section is organized in three subsections:

- Standard and Best practices
- Related conferences
- Related EC initiatives

As previously stated, The Standard and Best practices section contains the reference standards adopted for the development of the SPECS framework. For each standard, a brief

description and the reference pointers to its definition are provided. To better clarify the relevance of such sections, some of the referred standards are below reported:

- The Cloud Trust Protocol (CTP) which has been designed to provides the user a mechanism by which cloud service customers receive information about the security of the used cloud services;
- The ISO/IEC 19086 "Information Technology (Cloud Computing) Service Level Agreement (SLA) Framework and Terminology", which provides a reference standard for the formulation of a Service Level Agreement in Cloud Computing;
- The ISO/IEC 27017 which gives guidelines for information security controls applicable to the provision and use of cloud services;
- The Privacy Level Agreement (PLA) promulgated by the Cloud Security Alliance (CSA) to provide a reference standard to the privacy management and the data protection practices to be used according to data protection laws.

The Related Conferences section indeed lists the international meetings focusing on the state of the art of the Security SLA. Such international meetings give the opportunity to industrial and academic practitioners to meet and discuss about possible enhancement related to this relevant topic

Finally the section of the EU Initiatives reports a detailed list of the past and current projects, funded by the European Community, and related to the topics of Cloud Computing and Security SLA.

1.1.6 Publications

The Publications section provides a summary of the SPECS project dissemination. It is organized in three sub-sections:

- Public deliverables
- Publications
- Presentation

The first sub-section provides the deliverables related to the work packages described in the SPECS project specifications and reported also in the workpackage subsection, above described.

The second sub-sections provides a detailed list of publications made by the SPECS involved partners starting from 2013, the first year of the projects. The list in this section considers all publication in international conference proceedings, book chapters and international journals.

The third sub-sections, instead, contains the presentations made by the SPECS partners during the events described in the related sections. Each presentation is downloadable in a Pdf format for the consulting.

1.1.7 Media

The media section has been created and inserted in the SPECS website for advertisement purposes. It is, in fact, organized in four sub-sections:

- Flyers and Posters, containing the flyers adopted during the starting phases of the project or created for the SPECS events, in order to advertise interesting people about the event purposes, locations, dates and more.
- Press Releases, reports all the mentions and quotes obtained by the SPECS project during other public events non-strictly related to the project activities

- Videos, containing the screencast of some relevant presentations related to the SPECS activities and to the SPECS framework;
- Photos that provides some photos taken during the SPECS events, detailed in the related section.

A screenshot of this last section is reported in Figure 8.

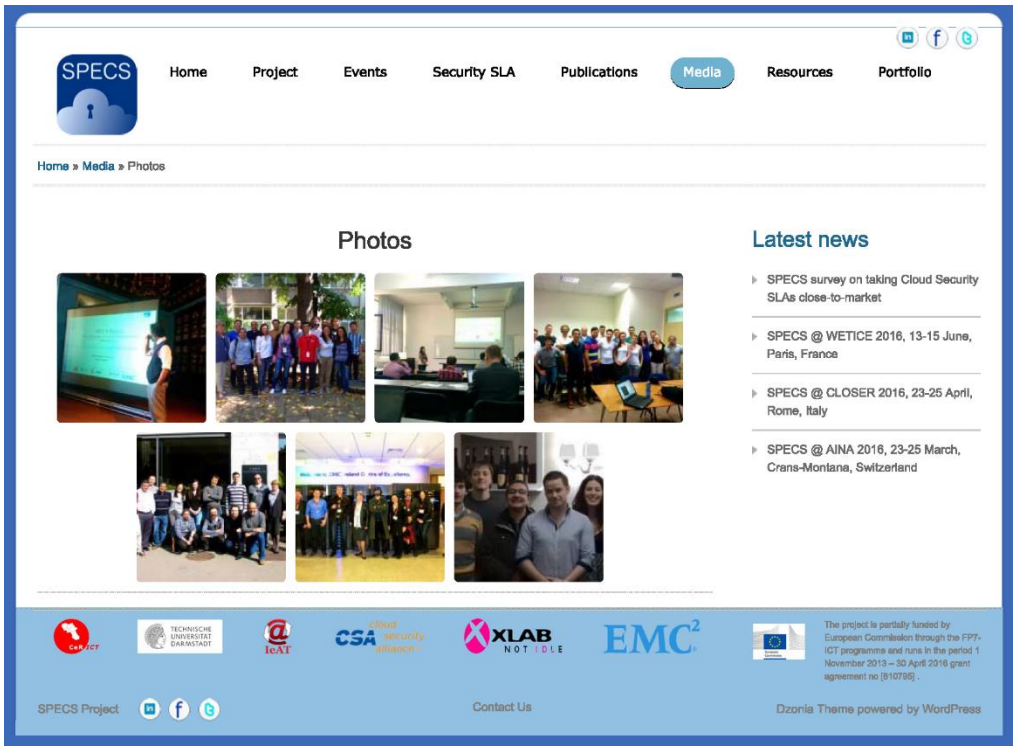

Figure 8- SPECS website Photo section

1.1.8 Resources

A key section of the website is the Resource one, since it not only provides some documents to understand the application domain and the developed framework, but gives also the link to access to the repository containing the components of the SPECS framework, that can be deployed, according to the ways described in the other deliverables.
With the respect to this, the Resources section is organized in four sub-sections:

- The Glossary that is in charge of providing a reference document to drive an user to understand the project activities and the published deliverables. The need for this sections is due to the ambiguous terminology adopted in cloud computing domains, deriving from the presence of several reference standards.
  The glossary is updated every time a new framework component is released.
- The Repository sub-section is maybe the most important since it provides the access to the repositories, containing the developed items and the demo environment that allow their deployment. The Repository subsection is depicted in Figure 9: as reported in the figure, it provides the link to a SPECS Web Container Demo, that gives an example of a SPECS application. Moreover the link to the code repositories are provided: repositories are hosted by the well-know Git platform Bitbucket (https://bitbucket.org/). A deep description of the organization of the code inside the bitbucket platform is provided in Section 3.

- Security SLA Model, which gives an overview of the Security SLA adopted in the SPECS framework. In particular the overview is provided by means of presentations, inserted thanks to the adoption of a Wordpress plugin.
- SPECS Software Navigator that gives the user the possibility to download SPECS applications and provides the necessary guides to enable him/her to deploy such applications. This section will be definitively updated with the last release of the above mentioned software

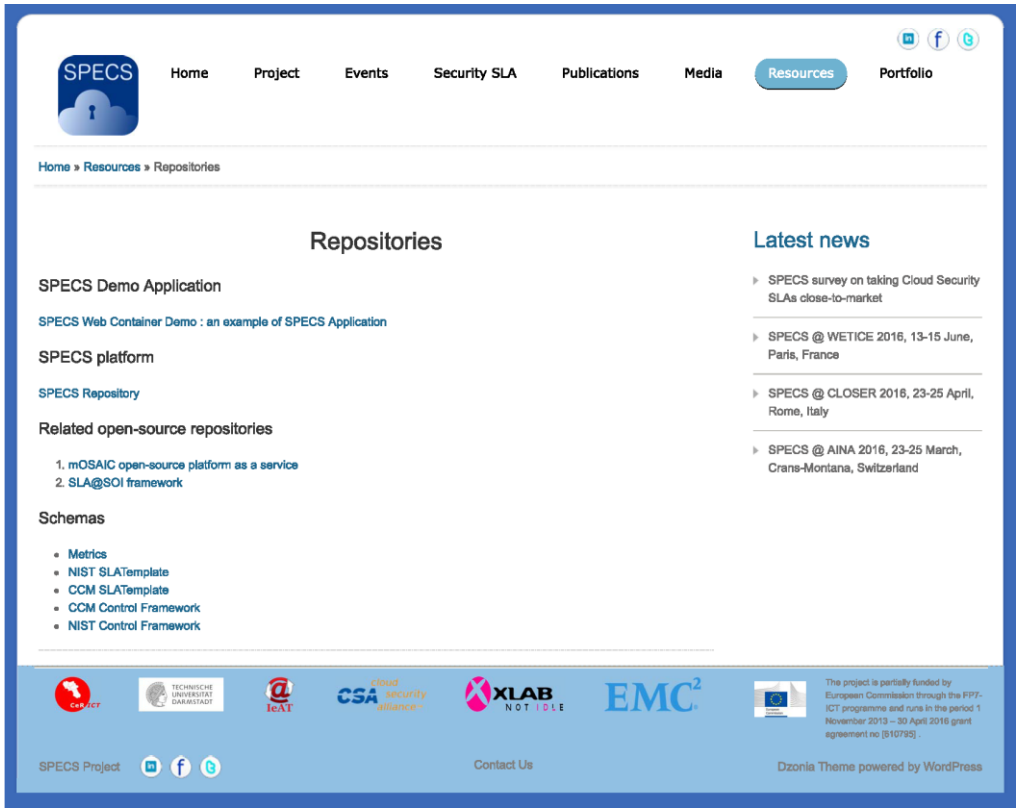

Figure 9- SPECS website Repositories section

1.1.9 Portfolio

The last section has been inserted to provide an overview of the achieved results obtained by the SPECS project activities on real software applications. In particular, each of the composing sub-sections represent a specific SPECS application resulting from the SPECS Project activities.

Each sub-section considering a particular SPECS application is organized in four parts: (1) the description of the considered application and of its provided functionalities; (2) the issues in the state of the art existing before the application of the SPECS results and (3) the advantages achieved on the same considered applications after the exploiting of SPECS outcomes. Finally (4) a summary of the SPECS components enabling such enhancements is provided.

1.2 Social Media

Three different social networks were considered, whose main features and current adoption will be described in the following.
In particular, according to the dissemination plan defined in D6.1.1 we frequently update the SPECS network of stakeholders with available results, news and upcoming events.

Twitter
Twitter is a real-time information network that connects users for many purposes (ideas, news,...). Users follow the accounts of groups and individuals that they find most interesting.
Twitter has over 100million active users, and there is an increasing take-up among individual academics and PhD students, by research projects by university departments and other groups connected to academia, such as research councils and funding bodies, publishers, and government departments.

A twitter account was created for the SPECS project, named **FP7SPECS**, as shown in Figure 10.

Here are some Twitter statistics in the three years of activities:

- Tweets: 442
  Retweeted posts by others: 402
- Favorite posts by others: 66
- Following: 129
- Followers: 200

The adoption of Twitter have boost dissemination for research projects in several ways, such as by spreading the news about new publications, website update or asking for feedbacks, using hashtags to make research material more visible.

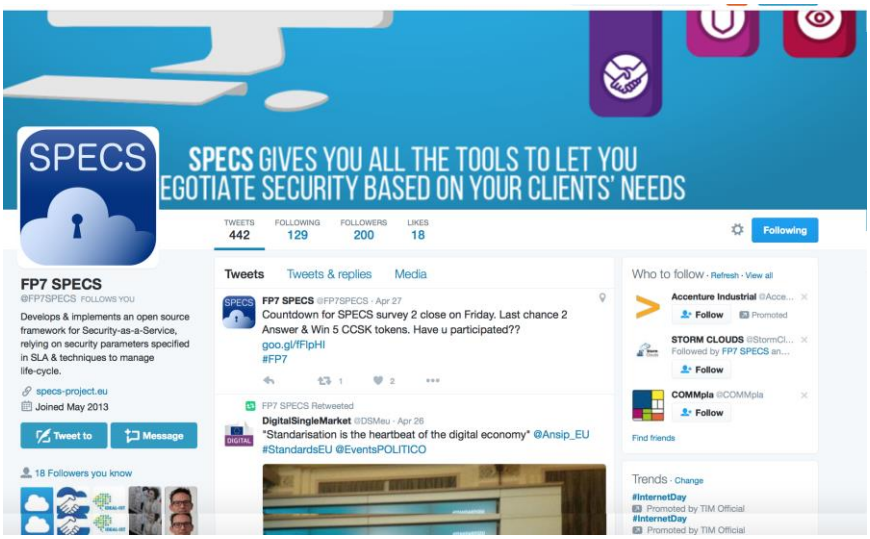

Figure 10. Twitter site of the project

Facebook
Facebook is a free-access social networking website that allows users to join one or more networks, such as a school, place of employment, or a geographic region to easily connect and interact with other people.

A Facebook page was created for the SPECS project, named **Fp-ict Specs**, as shown in Figure 11.

Here are some Facebook statistics in the three years of activities:

- Likes: 101
- Weekly post reach: 120
- Number of posts to date: 115

Although Facebook has encountered some criticism because of privacy concerns and because it can be used for publicity purposes, it has very strong assets that make people join and it was very useful to disseminate both internally and externally, the SPECS consortium activities.

Figure 11. Facebook site of the project

LinkedIn

LinkedIn is a social networking website for people in professional areas. It has more that 175 million users, and is widely used by recruiters. It allows people to show their achievements, work experience, research links etc., and is thus a fundamental tool to disseminate information about a research project to industries willing to acquire projects' outcomes and researchers willing to contribute to and use the technical project achievements.
A LinkedIn profile was created for the SPECS project, called **FP7 SPECS**, as shown in Figure 12.

There are currently 99 members.

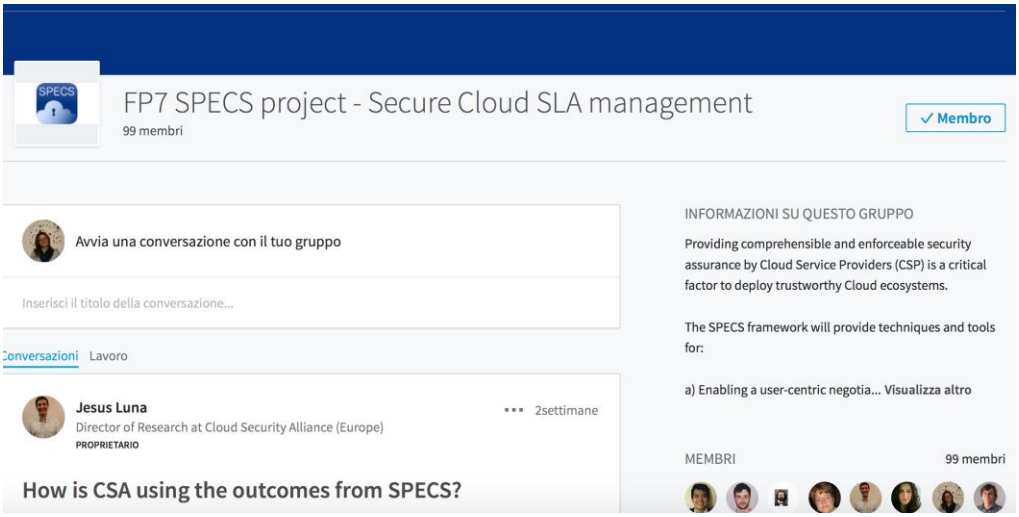

Figure 12. LinkedIn site of the project

1.3 Flyers and Posters

New flyers and posters were printed to disseminate the SPECS results. They were continuously distributed during scientific and industrial events. Figure 13 reports a picture of the last flyer, they all are available on line on the SPECS web site (media section).

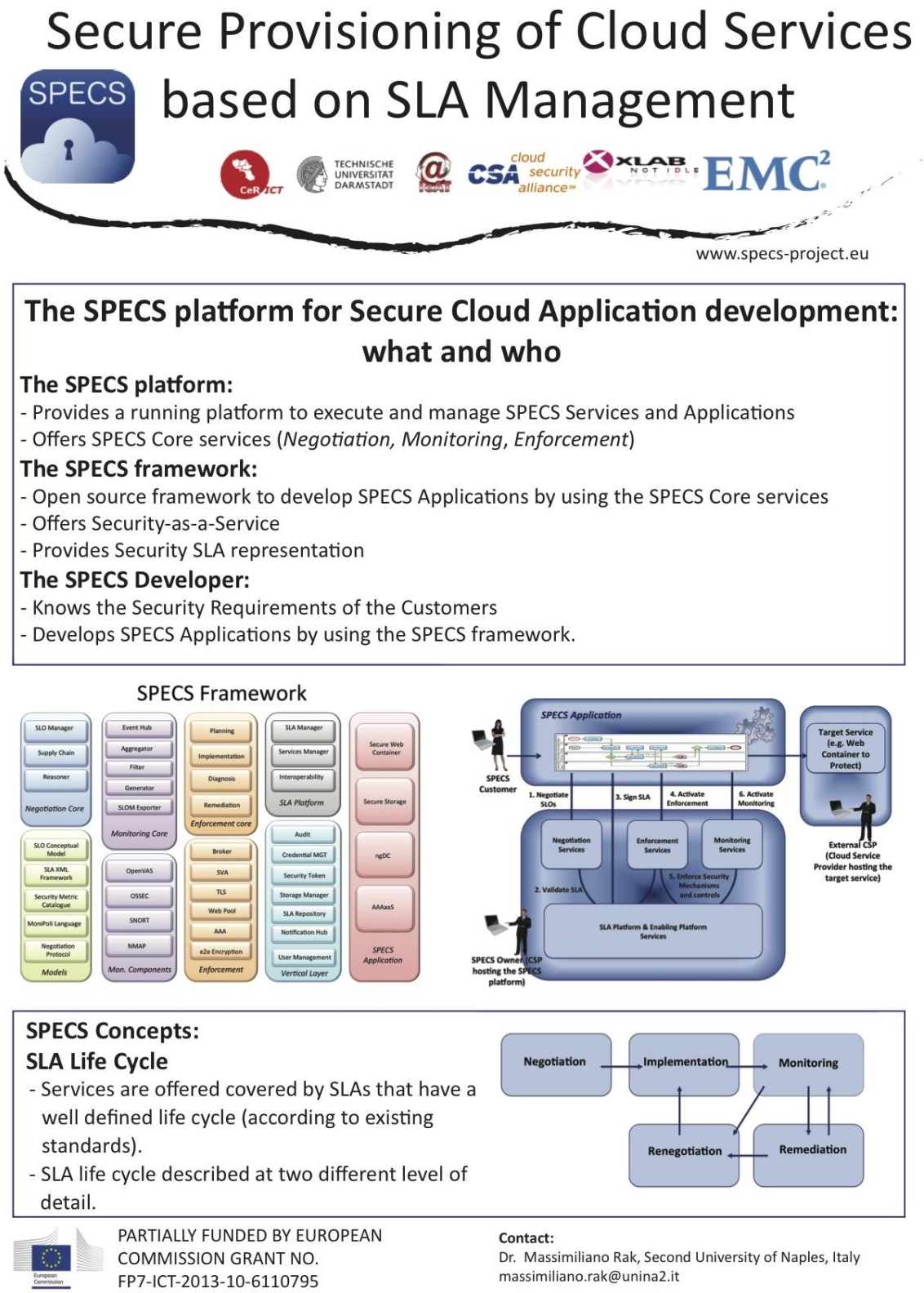

Figure 13. SPECS Flyer

2 The Software repository – developing with SPECS

2.1 The SPECS bitbucket repository

Git repositories are used in the SPECS project for source code management, version control and sharing. SPECS has a dedicated BitBucket account, which is used as centralized point for collecting all the source code repositories. The full SPECS framework, available as a prototype, is released as open source in the SPECS official BitBucket repository https://bitbucket.org/specs-team. All the sub-repositories are publicly available.

Connecting to the indicated link through a whatever web browser, you are connected to the overview page of the BitBucket official repository, as shown in Figure 14. This overview page shows a top menu where you can find general features of your BitBucket (such as language settings, search bar, etc.) and you can also log in.

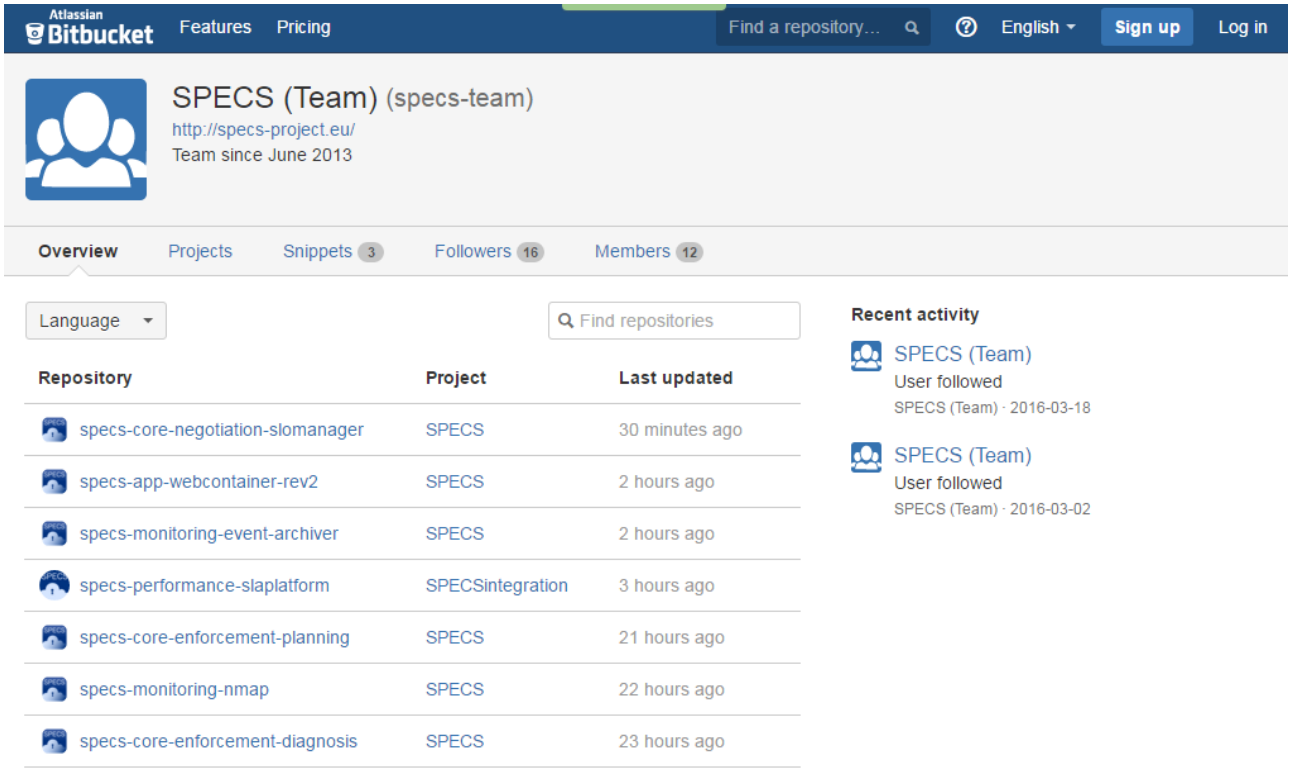

Figure 14. Home page of the official BitBucket repository

In the rest of the overview page you can see a three-columns table, in which all the sub-repositories are listed. For each sub-repository, the second column of the table reports the project to which it belongs, while the last update time is shown in the third column. Above this table, filtering options and search bar are available. Note that this search bar is different from the one reported in the top menu, since it allows to search specifically inside the SPECS BitBucket repository.

All the sub-repositories are organized into the following projects:

1. *SPECS*: collects all the repositories belonging to the full SPECS framework (Source Code Repository);
2. *SPECSintegration*: collects all the repositories related to the integration activities and testing;
3. *SPECSlegacy*: collects all the outdated repositories, no more supported in SPECS, as resulting from significant updates.

2.1.1 BitBucket functionalities

Different functionalities are offered natively by BitBucket. In details, beyond the organization and the sharing of the source code, BitBucket allows for the sharing of code *snippets*, for the management of *members* and *followers* lists. Actually, there are 17 followers and 13 members.

You can register yourself as a follower of the official repository by logging in BitBucket and clicking the *Follow* button. If you are not yet registered on BitBucket, you can freely register an account by clicking on the *Log in* button (on the upper right corner) and following the registration instructions. You can also obtain the access by using your own Google credentials.

As an authenticated BitBucket user (non SPECS-member user) you can read and clone the repositories, but you cannot edit or delete any file. If you edit or delete a file in a specific repository, BitBucket creates for you a clone of the selected repository, and executes your updates on it.

If you want to join the members list, you need to log in into BitBucket and send a private message to the administrators (by clicking on the *Email administrators* button on the upper right corner) with your request or, alternatively, send an email to the project managers by using the following email address: support@specs-project.eu.

Even as a non-authenticated user or as a non-member user, you can select a specific sub-repository (by clicking on the related row in the table), and access to the details page of the specific repository, as shown in Figure 15 for the *specs-app-webcontainer-rev2* repository.

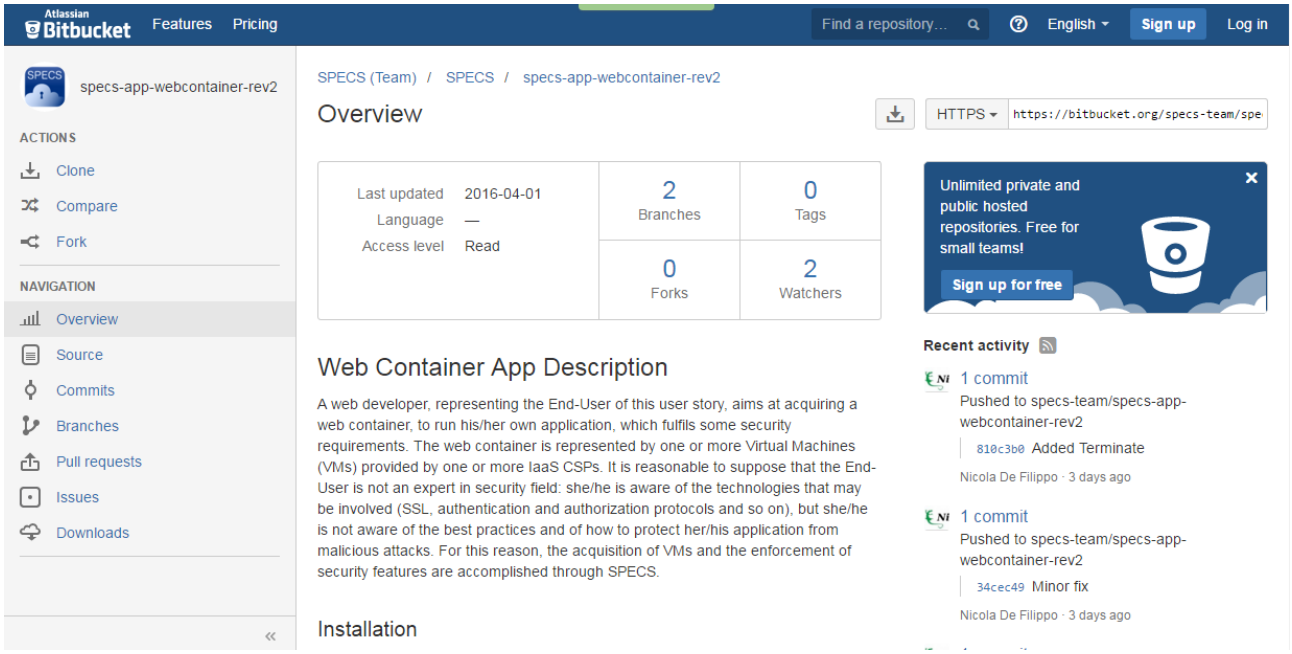

Figure 15. Details page

On this page you can find many information about the selected repository by navigating the left menu. On the *overview* page you find the date of the last update, indications about the source code language, your access level (*read* for non-member users) and the number of branches, tags, forks and watchers. In addition, if a readme file is present, you can find an overall description of the specific sub-repository, installation and compiling instructions, usage instructions and some examples. On the *source* page you can navigate the folder structure of the repository up to the files (of which you can access a preview). You can also access to the different branches and tags and you can execute a clone of the repository in your *Atlassian SourceTree* client. On the *commits* page (respectively, *branches* page) you will find the detailed log of the commits, including author, date and message (respectively, list of branches). On the *pull requests* and *issues* pages you will find, if any, the list of open requests and issues related to the specific repository. At last, you can obtain a copy of the repository (as a zip file) by navigating the *download* page. You can also compare different branches or tags by clicking the *compare* button.

If you are granted as a member, you will be enabled to execute commits on specific repositories, to fork existing repositories and to add additional repositories and new code snippets to the SPECS official repository.

2.1.2 List and organization of sub-repositories

Associated with the SPECS BitBucket official account, there are actually 89 repositories. 15 of them are related to the *SPECSlegacy* project, 5 of them are related to the *SPECSintegration* project, the remaining 69 belong to the *SPECS* main project.

Discarding the *SPECSlegacy* project (where no-more supported repositories are stored) and the *SPECSintegration* project (related to integration testing activites), all the repositories adopt the following dash-separated name convention:

*specs - {app, core, mechanism, utility} - **

The name in second position indicates the name of the macro-package to which the specific repository belongs. In detail:

- *app* indicates repositories containing source code related to the SPECS applications (e.g., platform dashboard, security metric catalogue, web container).
- *core* indicates repositories containing source code related to the core components (negotiation, monitoring and enforcement modules), SLA Platform, Enabling Platform and Vertical Layer;
- *mechanism* indicates repositories containing source code for the security mechanisms, i.e., monitoring and enforcement; the former contains
- *utility* indicates the source code related to the security-tokens and credential service components. Furthermore, the overall data model and specs parent repositories represent general source code needed by many other projects, as well as the XML SLA framework is another repository containing all the xml schema related to the SLA model.

The current number, grouped for macro-package, of repositories is shown in Figure 16.

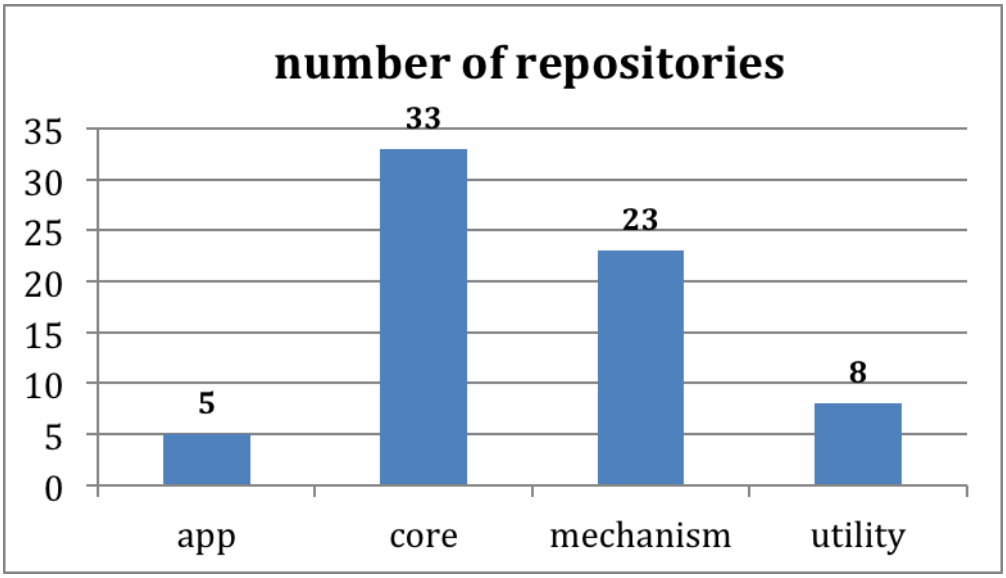

Figure 16. Number of repositories

With respect to the *specs-core* repositories, Figure 17 shows the current number of repositories, grouped for SPECS components (including SLA Platform and Enabling Platform). Analogously, Figure 18 shows the current number of *specs-mechanism* repositories.

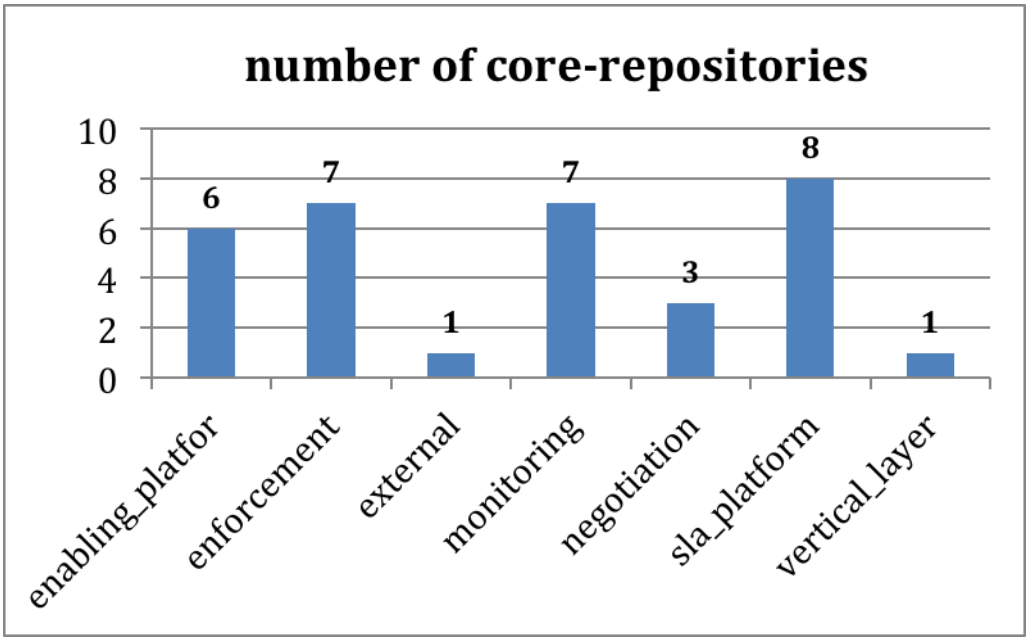

Figure 17. Number of repositories belonging to *core* package

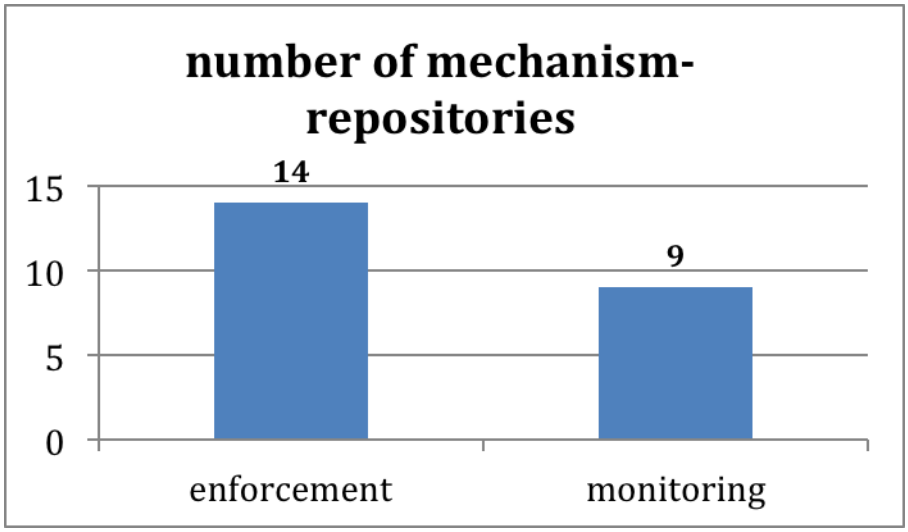

Figure 18. Number of repositories belonging to *mechanism* package

2.1.3 Artifact – repository mapping

The following Table 1 reports the mapping between the list of artifacts and the repositories. In detail, for each artifact we report in the first column the artifact's category, in the second column the specific artifact and in the third column we address for each artifact the list of related repositories which contain implemented code, demo, xml specification files, etc. Let us note that the following three repositories are useful and required by most of the source code developed in SPECS: *specs-utility-data-model*, *specs-utility-interoperable-data-model*, *specs-utility-specs_parent*.

Category	Artifact	Repositories
Negotiation Core	SLO Manager	specs-core-negotiation-slomanager
	Supply Chain Manager	specs-core-negotiation-supply_chain_manager

Category	Artifact	Repositories
	Security Reasoner	specs-core-negotiation-securityreasoner
Monitoring Core	Event Hub	specs-core-monitoring-event-hub specs-core-monitoring-adapter
	Metric Aggregator	specs-core-monitoring-aggregators
	(MoniPoli)Filter	specs-core-monitoring-monipoli
	Archiver	specs-core-monitoring-event-archiver
	CTP Exporter	specs-core-monitoring -cloud-trust-protocol-server specs-core-monitoring -cloud-trust-protocol-adaptor
Enforcement Core	Planning	specs-core-enforcement-planning
	Implementation	specs-core-enforcement-implementation
	Diagnosis	specs-core-enforcement-diagnosis
	Remediation	specs-core-enforcement-rds
	Broker	specs-core-enforcement-broker
Monitoring & Enforcement Mechanisms	TLS	specs-core-enforcement-tls specs-core-enforcement-repository specs-core-external-repository
	DoS Protection	specs-mechanism-monitoring-dos specs-mechanism-monitoring-ossec specs-mechanism-enforcement-dos specs-core-enforcement-repository specs-core-external-repository
	SVA	specs-mechanism-monitoring-sva specs-mechanism-enforcement-sva_core specs-mechanism-enforcement-sva_dashboard specs-mechanism-monitoring-openvas specs-mechanism-enforcement-sva_vulnerability_manager specs-core-enforcement-repository specs-core-external-repository
	NMAP	specs-monitoring-nmap specs-core-enforcement-repository specs-core-external-repository
	Web Pool	specs-mechanism-monitoring-webpool-adapter specs-mechanism-enforcement-webpool specs-core-enforcement-repository specs-core-external-repository
	DBaaS	*offered jointly with e2e Encryption mechanism*
	e2e Encryption	specs-mechanism-enforcement-e2ee-server specs-mechanism-enforcement-e2ee-client specs-mechanism-monitoring-e2ee-auditor specs-mechanism-monitoring-e2ee-adapter specs-mechanism-enforcement-e2ee-koofr-client specs-core-enforcement-repository specs-core-external-repository

Category	Artifact	Repositories
	AAA	specs-mechanism-enforcement-AAA specs-mechanism-enforcement-AAA-client specs-core-enforcement-repository specs-core-external-repository
SLA Platform	SLA Manager	specs-core-sla_platform-sla_manager specs-core-sla_platform-sla_manager-api
	Services Manager	specs-core-sla_platform-service_manager specs-core-sla_platform-service_manager-api
	Interoperability	specs-core-sla_platform-interoperability
	Metric catalogue	specs-core-sla_platform-security_metric_catalogue specs-core-sla_platform-security_metric_catalogue-api
Vertical Layer	Audit	specs-core-sla_platform-auditing
	Credential MGT	specs-utility-credential_manager specs-utility-credential-management-application specs-utility-credential-client
	Security Tokens	specs-utility-security-tokens
	User Management	specs-core-vertical_layer-user_manager
Enabling Platform	Launcher	specs-core-enabling_platform-cluster-launcher specs-core-enabling_platform-bootstrapper specs-core-enabling_platform-cloud-resource-allocator
	Custom OS	specs-core-enabling_platform-custom-os specs-core-enabling_platform-custom-os-packaging
	Core Repository	specs-core-enabling_platform-repository specs-core-external-repository
	Mechanism Repository	specs-core-enforcement-repository specs-core-external-repository
	Testbed Infrastructure	*physical infrastructure – no repository*
SPECS Application	Secure Web Container	specs-app-webcontainer-rev2
	Secure Storage	specs-app
	ngDC	specs-app
	AAAaaS	specs-app
	Platform Dashboard	specs-app-platform_interface
	Security Metric Catalogue app	specs-app-security_metric_catalogue
	Security Reasoner app	specs-app-SecurityReasoner
Models	Security SLA Model	specs-utility-interoperable-data-model
	Machine Readable Model	specs-utility-interoperable-data-model specs-utility-xml-sla-framework
	Security Metric Model	specs-utility-interoperable-data-model specs-utility-xml-sla-framework
	MoniPoli Language	specs-utility-interoperable-data-model
	SPECS REST APIs	*repositories related to all components offering SPECS REST APIs*

Table 1. Mapping between artifacts and repository

As resulting from Table 1, there is not a single repository reporting all the SPECS REST APIs, in fact specific APIs offered by a component are described and maintained in the specific repositories related to the component itself.

3 After the end of the project: the SW tools and repositories

3.1 The SPECS repository

The Bitbucket repositories will be maintained after the end of the project for a period of 3 years. All the open source repositories will be configured to support free cloning by any user that wants to use, change or extend the existent code. After 3 years the content will remain hosted on Bitbucket as long as Bitbucket allows free hosting for it. If before the 3 years limit the repositories needs to be moved from Bitbucket from whatever reason, the new location will be under the SPECS Archives repository, at the address:

- http://www.specs-project.eu/archives/bitbucket/

3.2 SPECS Continuous Integration and Deployment tool

SPECS Continuous Integration uses Atlassian Bamboo tool for managing the devops tasks. The tool is not free but licensed by the IeAT partner based on an open source license that we are entitled to use it in frame of the project until the end of it. Moreover the Bamboo instance uses a considerable amount of hardware that partner IeAT will have to use it in other activities. After the end of the project, within 3 months, the Bamboo instance will be decommissioned. All the partners will be asked to upload the last version of their resulting data into the FTP server. Moreover the current state of Bamboo will be saved, archived and hosted, for 5 years with private access, at the address:

- http://www.specs-project.eu/archives/bamboo-export/

3.3 SPECS Code Quality Management tool

SPECS uses SonarQube for code quality management. This service is collocated and integrated with Bamboo continous integration system. After the end of the project, within 3 months, the SonarQube instance will be decommissioned. The current state of the SonarQube will be saved, archived and hosted, for 5 years with private access, at the address:

- http://www.specs-project.eu/archives/sonarqube-export/

4 After the end of the project: the SPECS archives

During the project lifetime a considerable number of digital data and information was produced or developed. All this information will be archived after the end of the project. Each specific information will be archived and made either public or private (with limited access upon on request). All the archived data will be made available online at the address:

- http://www.specs-project.eu/archives/[archived_data_specific_identifier]

4.1 SPECS Website

The website and the domain name (specs-project.eu) will be maintained at least 5 years after the end of the project. The current content management system that drives the website will be replaced by a static version of the website within 6 months from the end of the project. After this process no modification over the content of the website will be possible.

4.2 SPECS Mailing lists

The mailing lists will be decommissioned within 6 months from the end of the project. All the mailing lists with no exception. The current mailing list data will be archived and hosted, for 3 years, with private access, at the address:

- http://www.specs-project.eu/archives/mailing-list/

A single mail address will be created, as an alias to several personal email address (the list of emails will be agreed by the partners). The email address alias will be: *post-project-support@specs-project.eu*. The new alias will be supported for 3 years from the end of the project.

4.3 SPECS Wiki

The wiki pages will be archived by generating static content from the current wiki system. This operation will be performed in within 6 months from the end of the project. The static content will be archived and hosted, for 5 years, with private access, at the address:

- http://www.specs-project.eu/archives/wiki

4.4 SPECS SVN Repository

The content of the SVN repository will be archived and hosted, with private access, at the address:

- http://www.specs-project.eu/archives/svn-repository

Within 3 months from the end of the project the SVN system will be decommissioned and all the data archived and stored at the address mentioned above.

4.5 SPECS FTP Repository

The content of the FTP server will be archived and stored, for 5 years with public and private access, at the address:

- http://www.specs-project.eu/archives/ftp

The current hostname ftp.specs-project.eu will be redirected to the new address. Within 3 months from the end of the project the FTP server will be decommissioned and the content will be archived and stored at the address mentioned above.

4.6 SPECS Deliverables and papers

After the end of the project all the public deliverables and the papers whose copyright allow open access, will be made available on at least one of the following repositories:

- CORDIS Web Site (http://cordis.europa.eu/)
- Open Aire (https://www.openaire.eu/)
- CeRICT Web Site (www.cerict.it)

5 Conclusions

The Deliverable 7.1.2 (Communication tools, project information package and control procedures: Public project site, flyers, internal and external wiki. Document templates. Organization of the knowledge base. Timetable and checklist. Parameters to be monitored. Includes quality and risk management plan.) is the second deliverable focused on communication tools.
Both deliverables (D7.1.1. and D7.1.2) presented the set of communication tools that can be adopted for both internal and external communications. SPECS, uses interactive communication tools in order to share information, to control and monitor the activities, and to disseminate the project achievements. In this deliverable, we focused the attention on the functionalities of the new web site, on the details on the software repository to manage and use the open source code and, a detailed guide to set up and run the platform has been described. Finally, we have discussed the activities that will be put in place after the end of the project, in order to maintain the code and sustain the project results for the next years.

The tools described in this document outlined the availability of the results gained in the thirty months of the project.
In particular, we would like to outline the availability of:

- an open source framework, available on SPECS bitbucket repository, which contains more than 60 different repositories, that can be reused;
- a set of public deliverables and papers that collect the set of results obtained by the project;
- a project website which offers a clear reference guide to both theoretical results (in terms of the deliverables and papers publicly available) and technical results, (thanks to guides and links to source code repositories)

6 References

[1] Project Web Site, http://www.specs-project.eu
[2] Project Wiki, http://wiki.specs-project.eu
[3] Documents SVN Repository, http://svn.specs-project.eu/documents
[4] Papers SVN Repository, http://svn.specs-project.eu/papers
[5] Source Code SVN Repository, http://svn.specs-project.eu/source
[6] Source Code Bitbucket Repository, http://bitbucket.org/specs-team

7 Annex A – SPECS Manuals

This guide shows the complete installation of a minimum set of SPECS components from scratch, by using the Chef configuration management and SPECS cookbooks. The objective of this guide is to allow an End-User to install SPECS on her/his own Chef Node and, specifically:

- to install the SPECS Platform (i.e., Enabling Platform and SLA Platform);
- to install SPECS Core Components (i.e., Negotiation, Monitoring, Enforcement and Vertical Layers);
- to enable the execution of SPECS Security Mechanisms.

For sake of simplicity, we use the OpenStack cloud platform as basic platform, since it is free and open-source. As shown in **Errore. L'origine riferimento non è stata trovata.**, the installation of SPECS on Chef Nodes is managed by a proper Chef Server and a Chef Workstation. According to the Chef's approach, the Chef Server is in charge of configuring one or more Chef Nodes by executing proper cookbooks, while the Chef Workstation is in charge of uploading cookbooks onto the Chef Server. The cookbooks needed for the installation of the Enabling Platform and the SLA Platform are stored in the following SPECS official BitBucket repositories:

- specs-core-external-repository: it hosts the cookbooks not maintained by SPECS but needed and referenced by the SPECS cookbooks

(https://bitbucket.org/specs-team/specs-core-external-repository)

- specs-core-enabling_platform-repository: it hosts a set of cookbooks needed to deploy, configure and control the SPECS Platform, and specifically the Enabling Platform components (mOS, Chef Server, Chef Clients, etc.), SPECS Core Components, SLA Platform, SPECS Cookbook template (to be used as an example for further SPECS Cookbooks) and the default SPECS Application (to be used for the definition of new SPECS Applications)

(https://bitbucket.org/specs-team/specs-core-enabling_platform-repository)

- specs-core-enforcement-repository: it hosts a set of cookbooks needed to deploy, configure and control the Enforcement components, which offer support for the SPECS Security Mechanisms (i.e., AAA, DBB, DoS protection, E2EE, SVA, TLS, WebPool)

(https://bitbucket.org/specs-team/specs-core-enforcement-repository)

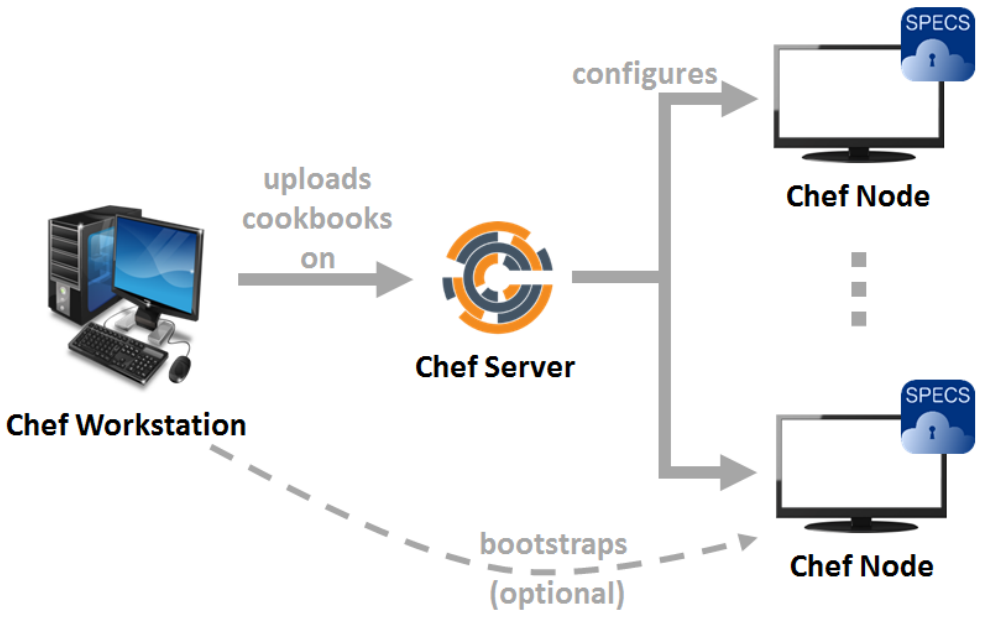

Figure 19. SPECS installation through Chef

This guide covers the complete installation and configuration of both a Chef Server, a Chef Workstation and a Chef Node on OpenStack machines. To acquire these machines, the usage of the Horizon dashboard (which is an implementation of OpenStack's Dashboard) is also shown.

Different steps are needed; in the following subsections we show:

- the installation of a Chef Server on an OpenStack machine, acquired through Horizon (see Section **Errore. L'origine riferimento non è stata trovata.**);
- the installation of a Chef Workstation on an OpenStack machine (see Section **Errore. L'origine riferimento non è stata trovata.**);
- the installation of the SPECS Platform on your Chef Node and the uploading on the Chef Server of the SPECS Security Mechanisms cookbooks (see Section **Errore. L'origine riferimento non è stata trovata.**).

By following these three steps, the End-User will completely install the minimum set of SPECS components on her/his Chef Node and will properly configure the Chef Server to enable the execution of SPECS Security Mechanisms on that specific node, by uploading their related cookbooks on the server. After this installation, the End-User (represented by a developer) is enabled to define, for example, a new application by customizing the default SPECS Application.

Detailed descriptions of SPECS Security Mechanisms (including manual installation and usage guides) can be found in deliverables D3.4.2, D4.3.2, D4.3.3 and D4.4.2; details about the definition of a new SPECS Application are provided in deliverable D5.1.3.

7.1 Installation of a Chef Server on an OpenStack machine

Requirements

- a VPN client
- an RSA key (stored in a file with *.pem* extension)
- a pre-configured network of machines
- a pre-configured router

Steps

1. Connect through the VPN client to Horizon (an implementation of OpenStack's Dashboard) in order to acquire a new VM instance, using the address http://10.10.10.2/horizon/, your own RSA key, your network and your router. Specifically, the overall minimum requirements for a Chef Server are the following:
   - number of virtual CPUs: 2 (preferred 4);
   - RAM capacity: 2GB;
   - HDD capacity: 20 GB.

2. In the *Details* tab (**Errore. L'origine riferimento non è stata trovata.**), populate the fields as follow:
   - *Availability Zone*: as by default, use *nova*;
   - *Instance Name*: choose your preferred instance name (we used *chef-server-cerict*);
   - *Flavor*: as by default, use *linux*;
   - *Instance Count*: as by default, use *1*;
   - *Instance Boot Source*: select *Boot from image*;

- *Image Name*: select *Ubuntu Server*.

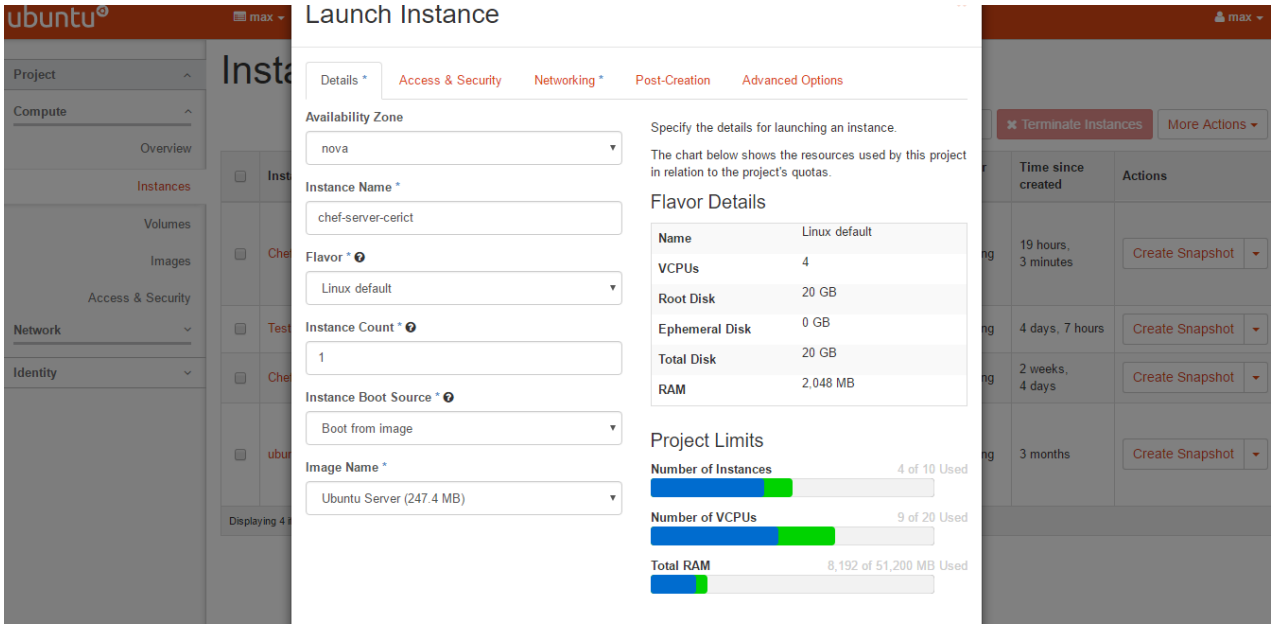

Figure 20. Horizon dashboard – Details tab

3. Select the *Access & Security* tab from the top menu and populate its fields as follows (**Errore. L'origine riferimento non è stata trovata.**):
   - *Key Pair*: select your own key pair (to create a new key pair, follow the instructions reported in point **Errore. L'origine riferimento non è stata trovata.**);
   - *Security Groups*: select your security group (we used *chef-server-security-group*, properly created with options described in point **Errore. L'origine riferimento non è stata trovata.**).

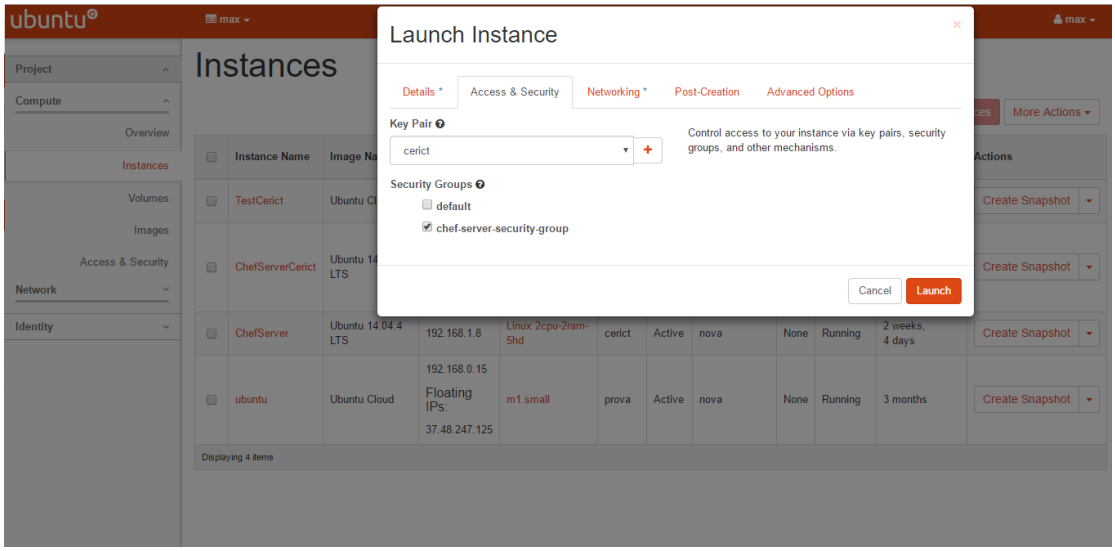

Figure 21. Horizon dashboard – Access & Security tab

- *Create a new key pair*. To create a new key pair, select the + button. It will open a new screen (**Errore. L'origine riferimento non è stata trovata.**), where you have to insert the key pair name and click *Create Keypair*. It will automatically download your private key.

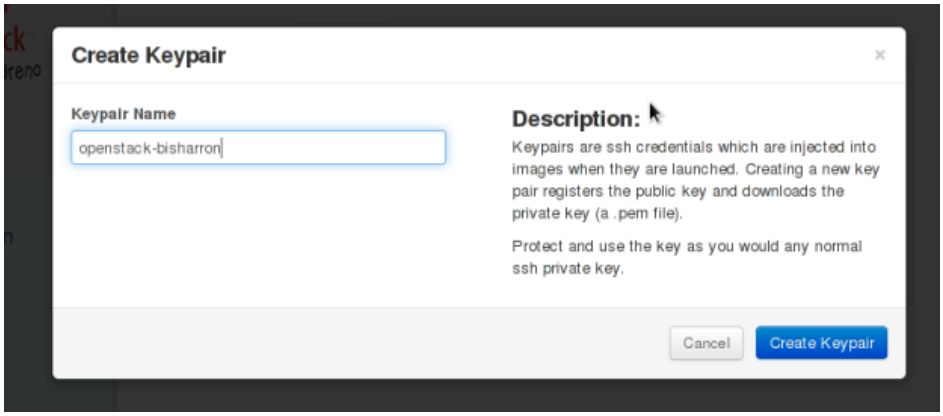

Figure 22. Horizon dashboard – create a new Keypair

- *Security group*. If you need a new security group, create a new one with the rules reported in **Errore. L'origine riferimento non è stata trovata.**. In detail, over all ports, any IP protocol needs to be configured as outgoing and the ICMP as incoming, while the TCP needs to be opened as incoming over ports 22 (SSH), 80 (HTTP), 81 and 443 (HTTPS).

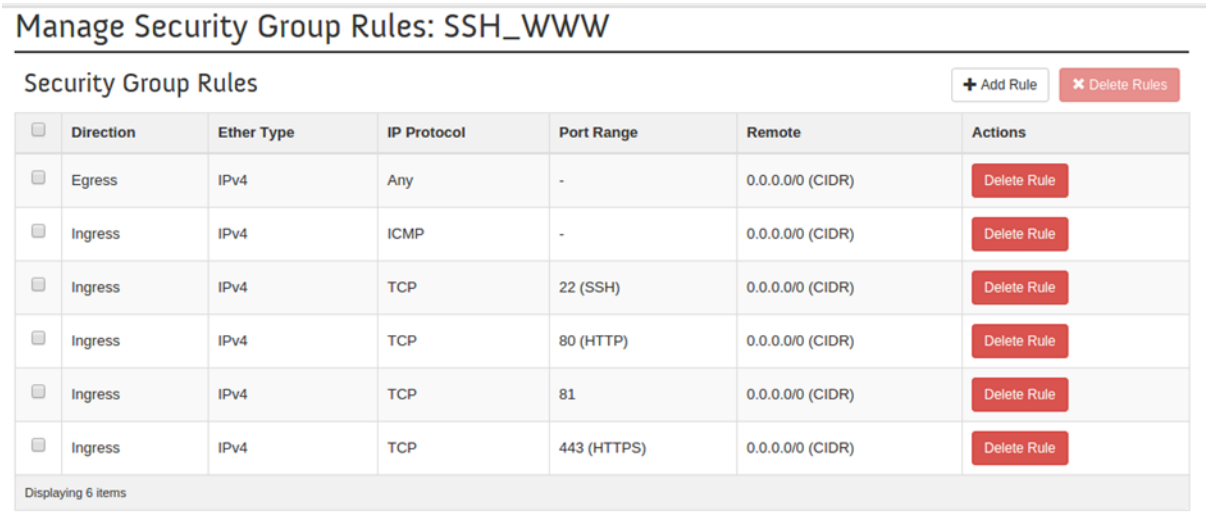

Figure 23. Security Group Rules

4. Select the *Networking* tab from the top menu and select your network (we used *prova-network*) in the field *Selected networks* (**Errore. L'origine riferimento non è stata trovata.**).

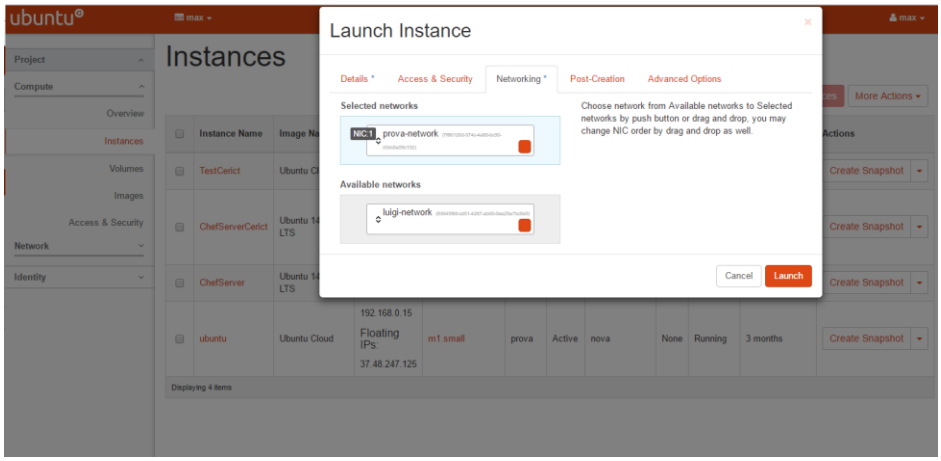

Figure 24. Horizon dashboard – Networking tab

5. Leave all fields as default in tabs *Post-Creation* and *Advanced Options*.

6. Run the new instance and associate a floating IP to it. Hence, come back to the *Instances* list view, select *Associate Floating IP* from the drop down menu on the right, related to the new instance (**Errore. L'origine riferimento non è stata trovata.**). From the *IP Address* list, select an available IP address and click on *Associate*.

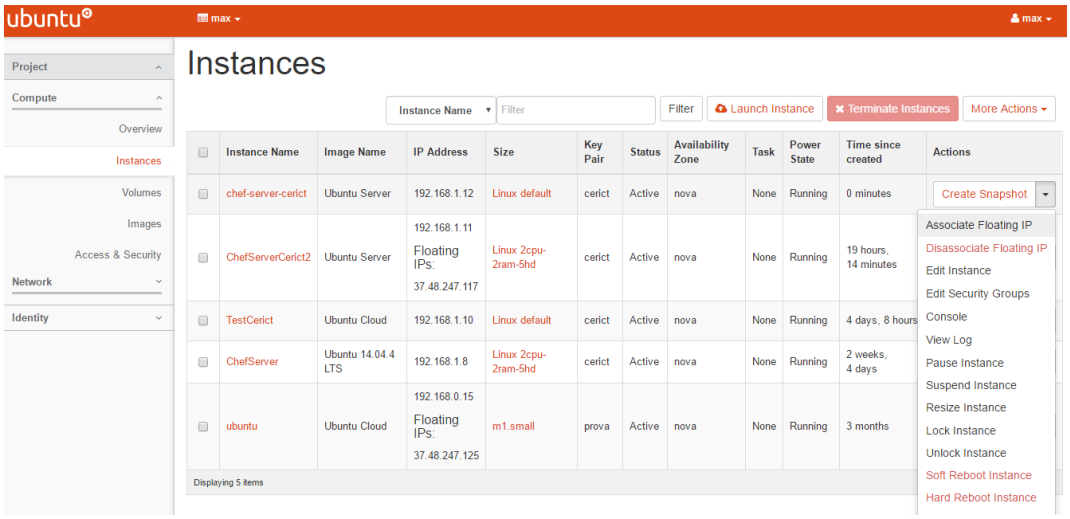

Figure 25. Instances view

7. Open an ssh connection towards the new instance, by using the IP address selected at point **Errore. L'origine riferimento non è stata trovata.**. The first time it will ask if the fingerprint should be stored, click *Yes*. When the shell will be opened, it will ask you to insert your login. Type *ubuntu* and press enter.

8. Verify that your instance has a resolvable name. To do this, type

 # hostname –f

If no name is returned, update the file */etc/hosts* as follows:

127.0.0.1 localhost
127.0.0.1 chefserver.cerict chef-server-cerict

192.168.1.12 chefserver.cerict chef-server-cerict

where *192.168.1.12* is the private IP address (not the Floating IP) of your instance inside the subnet and *chef-server-cerict* is its name. If you re-type

 # hostname –f

you should obtain something like

 # chefserver.cerict

9. Download and install the Chef Server. Update the file /etc/resolv.conf by inserting as first row the following:

 # nameserver 8.8.8.8

Execute the following command (where 12.0.5 is our selected version):

 # wget https://web-dl.packagecloud.io/chef/stable/packages/ubuntu/trusty/chef-server-core_12.0.5-1_amd64.deb

As soon as the download finishes, install it by using the following command:

 # sudo dpkg -i chef-server-core_12.0.5-1_amd64.deb

10. After the installation, reconfigure all by executing the following command:

 # sudo chef-server-ctl reconfigure

11. Configure the users and the organization on the Chef Server.
Execute the following command to create each user:

 # sudo chef-server-ctl user-create [username] [name] [surname] [email] '[password]' –filename [name.pem]

 # sudo chef-server-ctl user-create chefservercerict cerict cerict cerict@gmail.com 'cerict' --filename chefservercerict.pem

where *username = cerict*, *name = cerict*, *surname = cerict*, *email = cerict@gmail.com*, *password = cerict* and *name = chefservercerict*.

Execute the following command to create the organization:

 # sudo chef-server-ctl org-create name "[Long Name]" --association_user [username of the administrator] -f [shortname].pem

 # sudo chef-server-ctl org-create specs "SPECS project" --association_user chefservercerict -f specs-validator.pem

where *name =* specs, *Long Name = SPECS project*, *username of the administrator = chefservercerict* and *shortname = specs-validator*.

Both the credentials are stored in the path */home/ubuntu*

12. We can install the other Chef Server components by executing the following commands

- Console Management

 # sudo chef-server-ctl install opscode-manage
 # sudo chef-server-ctl reconfigure
 # sudo opscode-manage-ctl reconfigure

- Chef Push Jobs

 # sudo chef-server-ctl install opscode-push-jobs-server
 # sudo chef-server-ctl reconfigure
 # sudo opscode-push-jobs-server-ctl reconfigure

- Chef Replication

 # sudo chef-server-ctl install chef-sync
 # sudo chef-server-ctl reconfigure
 # sudo chef-sync-ctl reconfigure

- Reporting

 # sudo chef-server-ctl install opscode-reporting
 # sudo chef-server-ctl reconfigure
 # sudo opscode-reporting-ctl reconfigure

13. You can connect to the management console by typing from a browser the Floating IP of the instance and using the credentials used at point **Errore. L'origine riferimento non è stata trovata.**. In our case, we used:

user: chefservercerict
pass: cerict

You can now use all the functionalities, including the possibility of creating a snapshot of the instance on order to restore it in case of problems. To do this, for example, you can connect to the dashboard and click on *Create Snapshot*.

7.2 Installation of a Chef Workstation on an OpenStack machine

Requirements

- a VPN client
- an RSA key (stored in a file with *.pem* extension)
- a pre-configured network of machines
- a pre-configured router

Steps

1. Reconnect through the VPN client to Horizon, executing a new instance, using the address http://10.10.10.2/horizon/, your own RSA key, your network and your router. Specifically, the overall minimum requirements for a Chef Workstation are the following:
   - number of virtual CPUs: 1;
   - RAM capacity: 2GB;
   - HDD capacity: 20 GB

2. In the *Details* tab (**Errore. L'origine riferimento non è stata trovata.**), populate the fields as follow:
   - *Availability Zone*: as by default, use *nova*;
   - *Instance Name*: choose your preferred instance name (we used *Workstation Ubuntu*);
   - *Flavor*: use *m1.small*;
   - *Instance Count*: as by default, use *1*;
   - *Instance Boot Source*: select *Boot from image*;
   - *Image Name*: select *Ubuntu 14.04 4 LTS*.

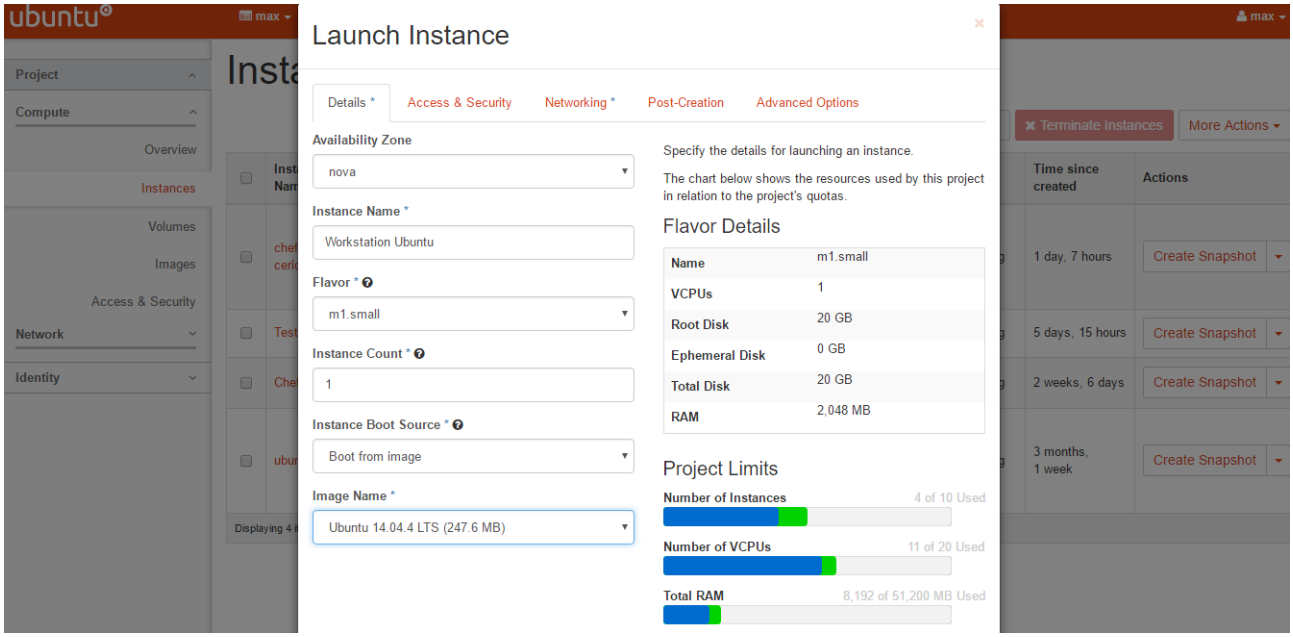

Figure 26. Horizon dashboard – Details tab

3. Select the *Access & Security* tab from the top menu and populate its fields as follows (**Errore. L'origine riferimento non è stata trovata.**):
   - *Key Pair*: select your own key pair (you can reuse the same keys generated/used for the Chef Server);
   - *Security Groups*: select your security group (the same used for the Chef Server).

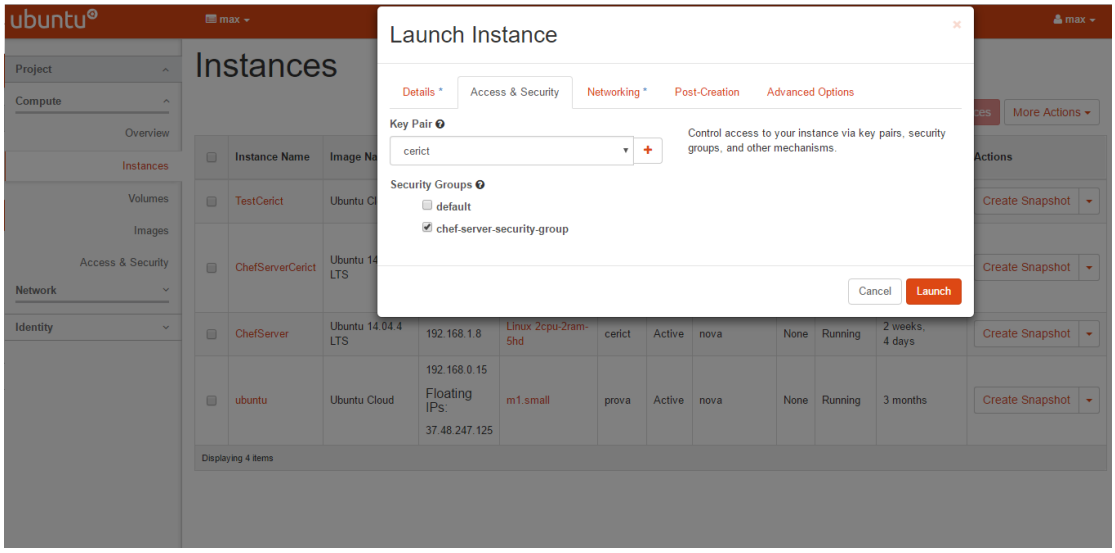

Figure 27. Horizon dashboard – Access & Security tab

4. Select the *Networking* tab from the top menu and select your network (we used *prova-network*) in the field *Selected networks* (**Errore. L'origine riferimento non è stata trovata.**).

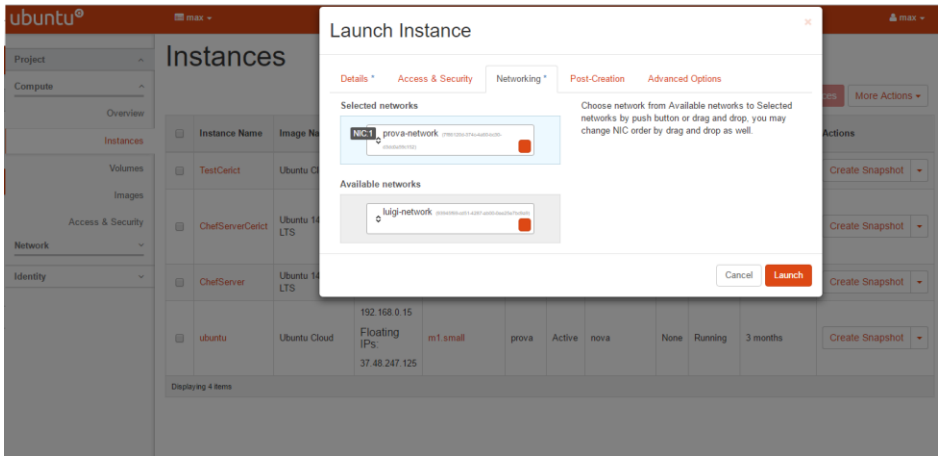

Figure 28. Horizon dashboard – Networking tab

5. Leave all fields as default in tabs *Post-Creation* and *Advanced Options*.

6. Run the new instance and associate a floating IP to it, as previously done for the Chef Server.

7. Open an ssh connection towards the new instance, by using the IP address selected at point **Errore. L'origine riferimento non è stata trovata.**. The first time it will ask if the fingerprint should be stored, click *Yes*. When the shell will be opened, it will ask you to insert your login. Type *ubuntu* and press enter.

8. Install GIT by executing the following commands:

 # sudo apt-get update
 # sudo apt-get install git

9. Clone a base version of a Chef repository by executing:

 # git clone https://github.com/chef/chef-repo.git

This operation will create a local directory named *chef-repo*. Move to this directory and edit the file *.gitignore* by adding the line *.chef* at the end of the file. After this, the file *.gitignore* should be:

```
.rake_test_cache

###
# Ignore Chef key files and secrets
###
.chef/*.pem
.chef/encrypted_data_bag_secret
.chef
```

Execute the commit of the repository.

10. Download the Chef Development Kit. Move to the directory */tmp* and let execute the following commands:

 # wget https://opscode-omnibus-packages.s3.amazonaws.com/ubuntu/12.04/x86_64/chefdk_0.10.0-1_amd64.deb
 # sudo dpkg -i chefdk_0.10.0-1_amd64.deb

Verify the correctness of the installation by typing

 # chef verify

and assure that all components are installed successfully.

11. Edit the file */etc/hosts* as follows, in order to allow the communications with the Chef Server:

```
127.0.0.1       localhost
192.168.1.12   chefserver.cerict chef-server-cerict
```

where *192.168.1.12* is the private address of the Chef Server, *chefserver.cerict* is the domain name of the Chef Server and *chef-server-cerict* is the instance name on which we installed the Chef Server.

12. Download the authentication keys of the Chef Server: move to the directory named *chef-repo* and create a new directory for the configuration and for the keys by executing:

 # mkdir .chef

Copy both the administrator's and user's keys from the Chef Server by opening a new terminal on your pc and executing:

 # sudo scp -3 -i Documents/cerict.pem ubuntu@37.48.247.121:/home/ubuntu/.pem ubuntu@37.48.247.117:/home/ubuntu/chef-repo/.chef/*

where *Documents/cerict.pem* is the path of the pem key (used for both the Chef Server and the Chef Workstation), *37.48.247.121* is the floating IP of the Chef Server and *37.48.247.117* is the floating IP of the Chef Workstation.

13. Let us configure *knife*. Move to the directory *chef-repo/.chef* and let us create a new file named *knife.rb*, which content is the following:

```
current_dir = File.dirname(__FILE__)
log_level                :info
log_location             STDOUT
node_name                "username"
client_key               "#{current_dir}/username.pem"
validation_client_name    "shortname-validator"
validation_key           "#{current_dir}/shortname-validator.pem"
chef_server_url          "https://chefservername/organizations/shortname "
syntax_check_cache_path   "#{ENV['HOME']}/.chef/syntaxcache"
cookbook_path            ["#{current_dir}/../cookbooks"]
```

change the following: *node_name* is the username created above (we used *chefservercerict*), *username.pem* under *client_key* must reflect your .pem file for your user (we used *chefservercerict.pem*), the *validation_client_name* should be your organization's *shortname* followed by –validator (we used *specs-validator*), *shortname.pem* in the *validation_key* path must be set to the shortname that was defined in the steps above (we used *specs-validator.pem*), finally the *chef_server-url* needs to contain the IP address or URL of your Chef Server, with the *shortname* in the file path changed to the *shortname* defined above (we used *https://chefserver.cerict/organizations/specs*).

14. Move to the *chef-repo* and copy the needed SSL certificates from the server by executing:

 # knife ssl fetch

Confirm the correct installation of *knife* by executing

 # knife client list

which should return the client short name defined above (in our case it returns *specs-validator*).

7.3 Installation of SPECS on a Chef Node and uploading of SPECS Security Mechanisms onto the Chef Server

The last step of this guide shows the uploading on the Chef Server of the cookbooks needed to install SPECS on your Chef Node and its basic configuration. Steps 1 and 2 are related to the acquisition of a new OpenStack machine and its bootstrap from the Chef Workstation.

Obviously you can acquire your node from a different provider and/or to bootstrap it in an alternative way.

Steps

1. Acquire a new OpenStack machine using Horizon and configure it as a Chef Node (as already described in the previous subsections). The steps are similar to those described for the Chef Workstation and for the Chef Node, we report the differences with the configuration of the Chef workstation, for sake of completeness:
   - number of virtual CPUs: 2;
   - RAM capacity: 2GB;
   - HDD capacity: 20 GB;
   - *Details* tab – *Flavor* field: select *linux*;
   - *Details* tab – *Image Name* field: select *specs-mos*.

Create and run this new instance and associate a floating IP to it.

2. To bootstrap the new instance from the Chef Workstation you have to execute the following command on the workstation itself:

 # knife bootstrap [IPNode] -x [username] -P [password] --node-name [nodeName] --sudo

where *[IPNode]* and *[nodeName]* shall be substituted respectively by the IP address and the name (as identified by the Chef Server) of the node to bootstrap, while *[username]* and *[password]* are your ssh credentials.
Alternatively, if the instance is not enabled to accept password-based authentication, you can use the key-based authentication, by executing the following command:

 # knife bootstrap [IPNode] -x [username] –identity-file [pemKey] --node-name [nodeName] –sudo

where *[pemKey]* shall be substituted by the path and filename of your pem key. To verify the correct bootstrap you can execute:

#knife node list

which returns the lists of all nodes managed by the Chef Server, where you have to recognize the name of your node.

3. Upload all the necessary cookbooks in the directory *chef-repo/cookbooks/* of the Chef Workstation. These cookbooks shall be downloaded from the BitBucket official repositories listed before. To do this, open an ssh connection towards the Chef Workstation and execute the cloning of the SPECS repositories, by executing:

 # git clone https://bitbucket.org/specs-team/specs-core-external-repository.git
 # git clone https://bitbucket.org/specs-team/specs-core-enabling_platform-repository.git
 # git clone https://bitbucket.org/specs-team/specs-core-enforcement-repository.git

4. Upload the cookbooks contained in the cloned repositories onto the Chef Server, by executing the following commands (one for each repository):

 # knife cookbook upload -a -o specs-core-external-repository
 # knife cookbook upload -a -o specs-core-enabling_platform-repository
 # knife cookbook upload -a -o specs-core-enforcement-repository

Alternatively, you can upload a single cookbook by using the following command:

 # knife cookbook upload [cookbook_name]

where *[cookbook_name]* represents the name of each specific cookbook. Note that you have to respect the same order of repositories.

5. Update the run list of the Chef Node to enable the execution of the cookbooks, by using the Web User Interface (by selecting the node and then clicking on *edit* in the *run list* section) or by executing the command:

 # knife node run_list add [nodeName] 'recipe[COOKBOOK::RECIPE_NAME]'

where *[nodeName]* shall be substituted by the name of the Chef Node.

The correct run list order must be the following:

1) *enabling-platform::apache-tomcat-v7*
2) *monitoring::event-hub*
3) *monitoring::event-archiver*
4) *monitoring::monipoli*
5) *monitoring::ctp-server*
6) *monitoring::ctp-adaptor*
7) *monitoring::wui*
8) *enforcement::planning*
9) *enforcement::diagnosis*
10) *enforcement::implementation*
11) *enforcement::rds*
12) *enforcement::aaa*
13) *sla-negotiation::slo-manager*
14) *sla-platform::default*
15) *sla-platform::service-manager*
16) *sla-platform::sla-manager*
17) *sla-platform::metric-catalogue*
18) *applications::metric-catalogue-app*
19) *applications::web-container-app-rev2*
20) *applications::platform-interface-app*
21) *vault-server*

Let us now open an ssh connection towards the node and execute the following command:

 # chef-client

Your node is now running and the SPECS Platform shall correctly be installed on it. Also the default SPECS Application shall be properly installed and can be customized to create a new SPECS Application.