Innovation Action

D4.2: FIWARE Lab operation guide

Project full title: Bringing FIWARE to the NEXT step
Contract No.: 732851
Strategic Objective: ICT-12-2016 - Net Innovation Initiative
Project Document Number: ICT-12-2016-732851-WP4-D.4.2
Project Document Date: 8 January 2018
Deliverable Type and Security: PU
Author: Fernando López (FF)
Contributors: José Ignacio Carretero (FF), Federico M. Facca (MARTEL), Fernando López (FF), Alfonso Pietropaolo (Engineering), Daniele Santoro (FBK)

Table of Contents

List of Tables

List of figures

Introduction

Executive Summary

This deliverable gives a detailed and up-to-date description of FIWARE Lab operation guide and the different procedures to produce the information requesting from the nodes. It also provides template documents to be used for the different procedures in the same line that it is defined the mail template used to communicate with the FIWARE Lab administrators. A previous variant of this deliverable was partially produced in 30th January 2017 under the deliverable "D.22.1.2 FIWARE Lab coordination operation and support report" and the content is public in the following direction:

https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Lab_Nodes_Handbook

Nevertheless, this is a first time in which we produce a complete deliverable with all the administrative and management operations of the node. Additionally, since, then there have been considerable developments which are documented here:

- Update of the requirements to create a FIWARE Lab node.
- Update of the procedure to become and discontinue a FIWARE Lab node.
- Update the steps to inform about the status of a new node.
- Update legal documents regarding Privacy Policy, Cookies Policy and Terms & Conditions follow the German legislation and detailed activities that we follow in the community.
- Update the steps for joining the FIWARE Lab with a clearer definition with examples of them.
- New section to show the available tools to help in the management of your node, including the previous tool (FI-Health, FIWARE Lab Infographics, Jira and Backlog) and the new ones developed during this period (Deep Log Inspection, Rancher, Mastermind).
- New process to create accounts based on OpenStack Keystone and description to synchronize base images (Ubuntu, CentOS, Debian) on the FIWARE Lab nodes.
- Clear description of the coordination and support procedures together with the way to obtain the FIWARE Lab statistics.
- Definition of the procedure to calculate and obtain the SLA level of the FIWARE Lab nodes besides with the procedure to inform the broken SLA towards the FIWARE Lab nodes.

It is not expected any new publication of this deliverable during the project. Nevertheless, this document is a dynamic one and will keep evolving during next year taking into account the

comments that we receive from the users and administrators of the FIWARE Lab node. After the publication of this Deliverable, the content will be translated to the Wiki page:

https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Lab_Nodes_Handbook

About This Document

This document describes the procedures, tools and how the procedures are materialized using the tools in FIWARE in the scope of the FI-NEXT project. The ultimate goal of both, tools and procedures, is to satisfy needs of all the actors in the FIWARE ecosystem: the FIWARE team, adopters (SMEs, developers & entrepreneurs), testers, infrastructure owners/operators, etc. in terms of use the FIWARE Lab resources.

This document does not describe the result of applying those procedures to get the information about the status of the FIWARE Lab nodes but just a guide to be used in order to provide the corresponding information and resolve the day by day activity related to the management of a FIWARE Lab node.

Procedures and tools are inspired by common practices followed in the developer world, SCRUM and ITIL methodology to deal with ITIL Service Desk, SLAs. This implies a fresher and more dynamic look based on modern platforms such as JIRA, JIRA Service Desk and GitHub.

This document is a snapshot of the current use of the tools as of the date of the document. Procedures and tools may vary dynamically to adapt to the changing nature of the informational or collaboration requirements of the team and the platform. Those changes are oriented to increase the productivity and reduce the lack of knowledge regarding the OpenStack knowledge.

We consider this document a FIWARE Lab reference guide for all the FIWARE Lab administrators and so we will distribute it to them, the current one and any new nodes that may join the federation in order to know how can they proceed.

Intended audience

The target audience of this deliverable is:

- All nodes that have joined or plan to join the FIWARE Lab federation.
- FIWARE experts and technical personnel providing deployment support and end-user support activities.
- Developers and operators of FIWARE Lab tools (also called FIWARE Ops).
- The FIWARE Lab coordination team, in order to evaluate whether a candidate infrastructure meets the minimum technical and operational requirements.
- The FIWARE Foundation as natural new owner and coordinator of FIWARE Lab.
- and, more in general, all the stakeholders of FIWARE community to evaluate status and directions of FIWARE Lab.

Reading recommendation

This document is divided into 10 sections:

- Section 1: introduction, intended audience and reading recommendation
- Section 2: how to setup a FIWARE Lab node
- Section 3: policies definition, update of the privacy policy, cookies policy and terms & conditions.
- Section 4: process for joining FIWARE Lab to help the nodes in the setup of a new node
- Section 5: operating and maintaining a FIWARE Lab node. Set of steps to help in the day-by-day in the operation of the node.
- Section 6: describe two management operations that have to be take into consideration of each of the FIWARE Lab nodes (account management and base image management i.e. Ubuntu, CentOS, Debian images).
- Section 7: overview of FIWARE Lab coordination and support. Set of tools and way of working in order to explain the activities in the FIWARE Lab community.
- Section 8: definition of the service level agreement, how it is measured and definition of the email templates to communicate the FIWARE Lab nodes.
- Section 9: additional document.
- Annexes: set of templates and documents provided to organize the activities of the FIWARE Lab.

1 Essential things to know before joining FIWARE Lab

As a new node of the FIWARE Lab federation, you will increase the capacity of our current facilities and provide the FIWARE services to the users. In the same way, the FIWARE Lab node will take the benefits of the knowledge that the FIWARE community has adquired in the management of this environment. In this guide, you can find the main steps to join the FIWARE Lab federation.

1.1 Requirements

The hardware requirements are associated with the number of end-users (community users) that can be maintained in the node. The minimum number of those customers have to be 20. Based on this, the hardware and networks requirements are specified in the following sections.

1.1.1 Hardware Requirements

The hardware requirements are associated with the number of end-users (community users) is the following:

- Up to 10 vCores per end-user.
- Up to 20 Gb RAM per customer.
- Up to 150 Gb Hard Drive per customer.
- At least 22 Gb for images management.

1.1.2 Network Requirements

The connectivity capacity will be used for two aims: connect to the backbone of the FIWARE Lab to support node management operations, and provide connectivity to deployed services for end-users. You have also to provide a pool of public IPs that can be used during the OpenStack deployment (in order to expose the public API) and at a later time by the FIWARE Lab end-users.

Networking requirements:

- 1 Gbps connectivity for the backbone;
- 100 Mbps Internet connectivity for end-users;
- Firewall to ensure security;

- At least 1 public IPv4 available for each end-user (20 public IPs for 20 community users) plus public IPs for the different OpenStack services.
- IPv6 support is desirable although not required.

1.2 Steps to become a New FIWARE Lab node

In order to actually become a new FIWARE Lab node, several activities are required from both the Node and FIWARE Lab admins. The first is the signature of a Letter of Intent in order to communicate to the node these steps, the SLA levels that they have to fulfil and the FIWARE Lab user data ownership and legislation to be applied in terms of Data Protection. Additionally, it is explained that the no fulfilment of the SLA will produce a disconnection of the node from the FIWARE Lab. The Annex D: Letter of intent to become a new FIWARE Lab node contains the template of this Letter of Intent.

The table below describes, step by step, the whole procedure as it is at the time of writing this document.

Some steps may change in the near future due to the R6.4 of FIWARE Keyrock that includes the migration from Keyrock to Keystone and the Federation of OpenStack Keystone instances, therefore in short time all the nodes can use their own OpenStack Keystone instance and create use on the FIWARE Lab node.

Table 1.1: Steps to become a new FIWARE Lab node

TASK	TASK DESCRIPTION	TASK OWNER
00	Sign the Letter to become a FIWARE Lab node.	Node Admin
01	Provide contacts information.	Node Admin
02	Insert the new node within the agenda of the weekly meetings.	FIWARE Lab Admins
03	Join the weekly meetings.	Node Admin
04	Join the fiware-lab-federation-nodes@lists.fiware.org mail list.	Node Admin + FIWARE Lab Admins
05	Insert the new node in Jira for help-desk, sprint and FLUA management.	FIWARE Lab Admins
06	Sign in for Jira account: https://jira.fiware.org	Node Admin
07	Sign in for a FIWARE Lab account: https://cloud.lab.fiware.org	Node Admin

08	Webinar to explain how to use the main tools and methodology, such as Jira, Sprints, Help tickets, FLUAs.	Node Admin + FIWARE Lab Admins
09	Create sprint activities	FIWARE Lab Admins
10	Install the local OpenStack node	Node Admin
11	Federate the node	Node Admin
12	Update the federation data within the corresponding workitem created for the federation process	Node Admin
13	Install and configure the monitoring system	Node Admin + FIWARE Lab Admins
14	Configure the Sanity Check	Node Admins + FIWARE Lab Admins
15	Insert the new node within Infographic, Health Status and Sanity Check	Node Admins + FIWARE Lab Admins
16	Synchronize GE image list	FIWARE Lab Admins

1.3 Steps to discontinue a FIWARE Lab node

In order to discontinue the operation of a running FIWARE Lab node there are several activities and obligations to be respected. The most important, which is the base of the FIWARE philosophy, is taking care of the migration of the resources toward a persistent node, in order to avoid service outage.

The table below describes all steps, as they are now at the time of writing this document.

Some steps may change in the near future due to the R6.4 of FIWARE Keyrock that includes the migration from Keyrock to Keystone and the Federation of OpenStack Keystone instances, therefore in short time all the nodes can use their own OpenStack Keystone instance and create use on the FIWARE Lab node.

Table 1.2: Steps to discontinue a FIWARE Lab node

TASK	TASK DESCRIPTION	TASK OWNER
01	Inform the FIWARE Lab management about your decision at least 1 month before the shutdown	Node Admin
02	Inform all FIWARE Lab users about the shutdown via the FIWARE Lab notification tool	FIWARE Lab Admins
03	Take care of the migration of all active users toward a persistent node	Node Admin + FIWARE Lab Admins
04	Disconnect the node from centralized Keystone	Node Admin + FIWARE Lab Admins
05	Delete the node from the list of available nodes in JIRA, FLUA, Sanity Check, Infographic, Health Status	FIWARE Lab Admins
06	Delete from the fiware-lab-federation-nodes@lists.fiware.org mail list	FIWARE Lab Admins
07	Delete the node from the weekly meeting agenda/minute	FIWARE Lab Admins
08	Ask the node to send a final report/lessons learnt	FIWARE Lab Admins + Node Admin

1.4 Steps to show the current status of your FIWARE Lab node

The table below represents the template used to check the current status of a node.

Table 1.3: Steps to show the current status of your FIWARE Lab node

Node Name	
OpenStack Version	
Monitoring Version	
Infographic Status	
Health Check Status	
Federation Status	

In order to retrieve all those information is necessary to perform the following operation:

- HOW TO CHECK THE NODE NAME
  ```
  ssh login into one controller node

  grep region_name /etc/nova/nova.conf
  ```

- HOW TO CHECK THE OPENSTACK VERSION
  ```
  ssh login into one controller node

  nova-api --version

  (e.g. 2015.1.1)
  ```

  Take note of 2015.1.1 and go to https://wiki.openstack.org/wiki/Release_Naming to see the corresponding OpenStack release (in this example Kilo).

- HOW TO CHECK THE MONITORING VERSION
  ssh login into one controller node

  search the fiware-check-monitoring.sh script

  run fiware-check-monitoring.sh -v

  Check the text “FIWARE Lab Monitoring System release xxxx”

- HOW TO CHECK THE INFOGRAPHIC STATUS
  Go to http://infographic.lab.fiware.org

  Check the status of the node

- HOW TO CHECK THE HEALTH CHECK STATUS
  Go to https://fi-health.lab.fiware.org and check the Sanity Check status.

  Go to http://status.lab.fiware.org and check the historical performance of the node and available resources.

- HOW TO CHECK THE FEDERATION STATUS
  Go to http://backlog.fiware.org/lab/backlog and check the closing date of the corresponding workitem.

2 Policies definition

2.1 Privacy Policy

The Privacy Policy is changed taking into account the change of ownership of the data from Telefónica I+D, S.A.U. to FIWARE Foundation, e.V.

Additionally, a depth revision of the content of this policy was made in order to update the content and provide a more detailed information about it. The result of this update can be seen in the corresponding Annex E: Private Policy updated and it will be updated at the following URL:

http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Privacy_Policy

2.2 Cookies policy

An update of the current Cookies Policy was driven in order to reflect the changes in the ownership of the data. The Annex F: Cookies Policy updated reflects these changes and in the same way reflected in the following URL:

http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Cookies_Policy_FIWARE

2.3 Terms & Conditions

The Terms & Conditions is also updated following the last updated in the FIWARE ecosystem. This content makes reference to the use of the FIWARE Lab resources and the definition of the rights and obligations both in FIWARE Lab and user of these resources. The termination of these conditions and how to apply it besides with the intellectual property and proprietary rights. This content is reflected in the following URL:

http://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_LAB_Terms_and_Conditions

The update of these Terms & Conditions will be notified to the users via email in order that they will be notified about the changes of them.

3 Process for joining FIWARE Lab

3.1 OpenStack Services Required

FIWARE Lab Nodes are based on the OpenStack distribution. Please take a look to the section OpenStack upgrade version policy in FIWARE Lab (see section 3.2) to understand which version of OpenStack should be running on the nodes. As such and to the time of writing this document, nodes are required to install the following OpenStack services based on the "'OpenStack Newton release"':

- Mandatory:
  - OpenStack Nova (using KVM as hypervisor since image catalogue stores KVM compatible images).
  - OpenStack Glance (Swift as default backend type, the solution may be adopted depending on hardware owned by the specific FIWARE Lab node).
  - OpenStack Cinder (as default solution we suggest LVM, other solution may be adopted depending on hardware owned by the specific FIWARE Lab node).
  - OpenStack Neutron with OVS and GRE or VxLAN tunnels (floating IPs must be made available to users. The minimum number is currently under discussion).
  - OpenStack Ceilometer with MongoDB as backed as default solution.
  - OpenStack Keystone only for initial setup and testing, then FIWARE Lab keystone should be used[1].
  - OpenStack Horizon only for initial setup and testing, then FIWARE Lab Cloud Portal should be used.
- Optional
  - OpenStack Swift with 3 replication factor value. Optionally, could be installed CEPH with the OpenStack Swift APIs.
  - OpenStack Murano with OpenStack Heat for PaaS capabilities.
  - OpenStack Magnum with Swarm for managed docker.

[1] The R6.4 of FIWARE Keyrock includes the migration from Keyrock to Keystone and the Federation of OpenStack Keystone instances, therefore in short time all the nodes can use their own OpenStack Keystone instance and create use on the FIWARE Lab node.

3.2 OpenStack upgrade version policy in FIWARE Lab

FIWARE Lab nodes are based on OpenStack. OpenStack is developed and released around 6-month cycle.

According to FIWARE Lab management rules the upgrade policy of a FIWARE Lab node is two version behind the current official version under development. This is to avoid unsupported-EOL OpenStack release, security and performance issues. This policy secures us that we are almost in line with the community. It is important to notice that the upgrade of the OpenStack version can involve also the upgrade of the Operative System. The recommendation of the FIWARE Lab is the use of Ubuntu like Operating System. The next image shows as detail about the FIWARE Lab policy in use:

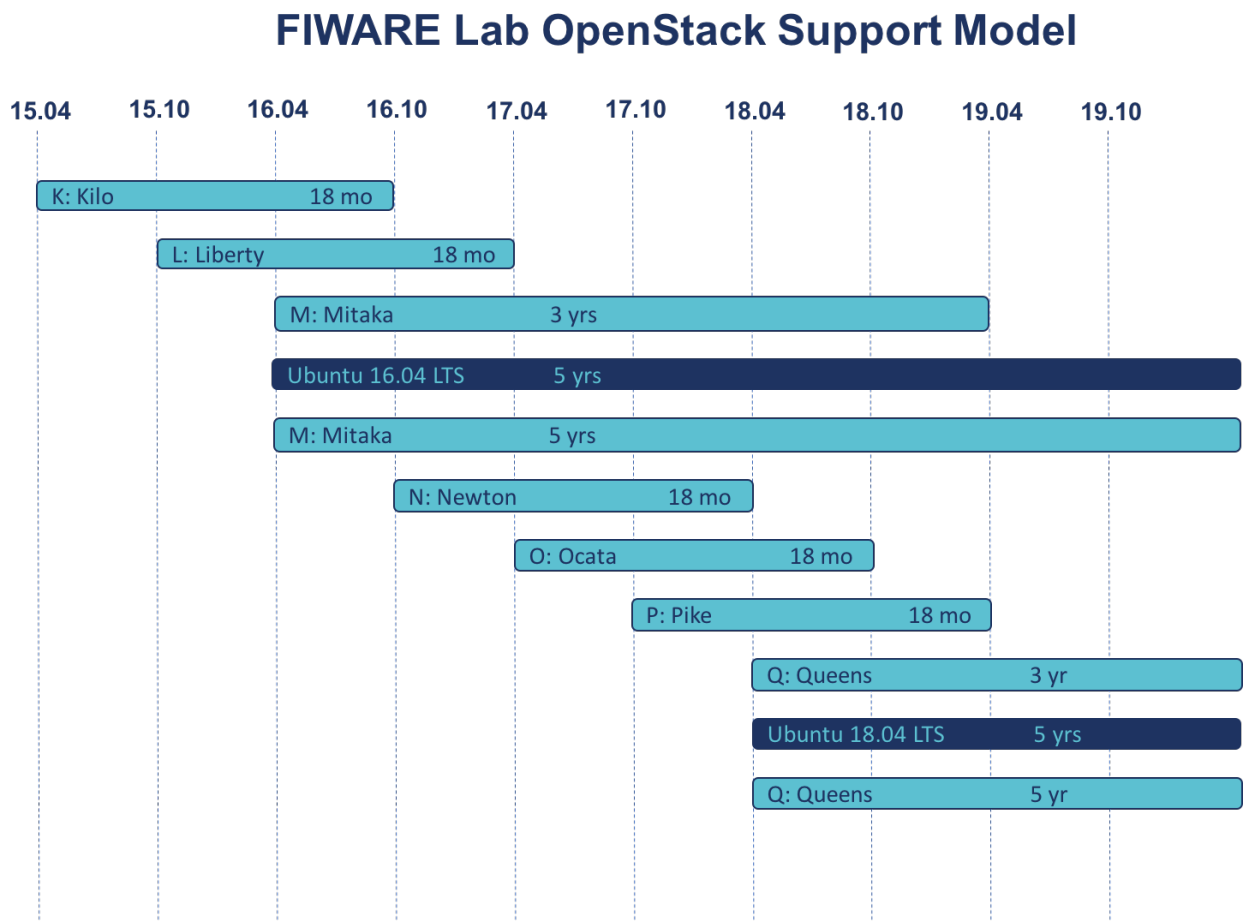


Figure 3.1: FIWARE Lab OpenStack support model

More information about the release series of OpenStack can be found at the following link: https://releases.openstack.org

IMPORTANT:
A FIWARE Lab Node not updated, will stop working properly because not compatible with FIWARE Lab services!

3.3 Installing FIWARE Lab Node

3.3.1 Introduction

In order to install your FIWARE Lab Node you can choose among different options that allows you to deploy an updated version of Vanilla OpenStack compatible with the requirements listed in section "Essential things".

3.3.2 How to install a FIWARE Lab Node

Currently, the OpenStack community offers many ways to install a complete environment and the procedure can be manual or automatic. Thanks to novel DevOps techniques the current trend is to leverage Infrastructure as Code concept and IT Automation tools like Ansible, Puppet or Chef in order to provision and maintain such complex systems.

Moreover Operating System Virtualization (Container-based) helps the management and the upgrade of all services running in the OpenStack based FIWARE Lab Node and it also grants the compatibility and portability of those services across different Operating Systems2.

For the above reasons FIWARE suggests the usage of IT Automation tools and Container-based virtualization in order to setup and maintain FIWARE Lab Nodes. Hereunder are references projects currently supporting the OpenStack installation:

1. **OpenStack-Ansible:** OpenStack services are automatically installed by Ansible and run inside LXC containers.
    a. https://docs.openstack.org/project-deploy-guide/openstack-ansible/
    b. https://docs.openstack.org/openstack-ansible/latest/
    c. https://github.com/openstack/openstack-ansible
2. **Kolla & Kolla-Ansible:** OpenStack services run inside pre-built Docker containers offered as Docker images from the Docker Hub and installed on nodes by Ansible.
    a. https://wiki.openstack.org/wiki/Kolla
    b. https://docs.openstack.org/kolla/latest/
    c. https://github.com/openstack/kolla-ansible

[2] https://developer.ibm.com/opentech/2016/11/21/openstack-in-containers/

Of course, manual installation is still possible, although discouraged as it results in more difficult management primarily due to package dependencies:

3. **Manual Installation & Configuration:**
    a. https://docs.openstack.org/install-guide/
    b. https://docs.openstack.org/install/

3.3.3 Suggested deployment architecture

To join FIWARE Lab no minimal requirement is enforced but the infrastructure must be adequate to support the needs of users who will be hosted on the new nodes. Obviously during the first node setup may not be clear how many users will be active and neither their needs in term of resources. For the above reasons it is strongly recommended, for a production environment, to follow the suggested deployment architecture:

- 3 Controllers in HA (including also Neutron L3 HA solution) with the following services
    - The nova-scheduler service, that allocates VMs on the compute nodes.
    - The cinder-scheduler service, that allocates block storage on the compute nodes.
    - The glance-registry service, that manages the images and VM templates. The backend for the registry maybe the controller node, or the Object Storage.
    - The neutron-server service, that manages the VM networks.
    - The heat-api and engine
    - The swift-proxy service that manages request to the object storage nodes.
    - The nova-api service, that exposes the APIs to interact with the nova-scheduler.
    - The cinder-api service, that exposes the APIs to interact with the cinder-scheduler.
    - The glance-api service, that exposes the APIs to interact with the glance-registry.
    - The keystone service, that manages OpenStack services in a node.
- (Optional) 3+ Object storage nodes with the following services:
    - The swift-account-server service, that handles listing of containers.
    - The swift-container-server service, that handles listing of stored objects.
    - The swift-object-server service, that provides actual object storage capability.
- 6+ Compute nodes (including also Cinder LVM) with the following services
    - The nova-compute service, that manages VMs on the local node.
    - The cinder-volume service, that manages block storage on the local node.
    - The neutron-agent service, that manages VM networks on the local node.
- 3 Ceilometer nodes

3.4 Configuring FIWARE Lab Node

This section provides details about how to make the proper changes in the configuration of your node in order to join FIWARE Lab. Those changes are basically related to the proper configuration of flavours and quotas and more important, related to the common way to define the available networks in a FIWARE Lab node.

3.4.1 Configure Flavors and Quotas

The default flavors should be:

Table 3.1: Default defined flavours

ID	Name	Memory (MB)	Disk (Gb)	Ephemeral	Swap	vCPUs	RXTX Factor	Public
1	m1.tiny	512	1	0		1	1.0	True
2	m1.small	2048	20	0		1	1.0	True
3	m1.medium	4096	40	0		2	1.0	True
4	m1.large	8192	80	0		4	1.0	True

For the nova service, default quotas the values that should be defined are the following:

Table 3.2: Default defined quotas

Quota	Limit
Instances	2
Cores	4
RAM	4096
Floating IPs	1
Fixed IPs	-1
Metadata Items	1024

Injected files	5
Injected file content (bytes)	20240
Injected file path (bytes)	255
Key pairs	10
Security Groups	10
Security Group Rules	20

The neutron default quotas should be:

Table 3.3: Default defined neutron quotas

Field	**Value**
Floating IP	1
Network	5
Port	20
Router	1
Security Group	-1
Security Group Rule	-1
Subnet	5

3.4.2 Configure OpenStack Networks

FIWARE Lab has defined a predefined name of networks to be used by all the nodes. It helps to the different services deployed on top of OpenStack to work with the correct network without any special configuration on it.

- **public-ext-net-01**
  This is the Public External network, a non-shared network providing a floating IP pool (i.e. subnet) that provides public, routable IPv4 addresses. Additionally, nodes can configure IPv6 dual-stack on this network in order to provide IPv6 addresses. This network is not visible to attach directly OpenStack Instances on it. It is only visible to allocate public IPs to be used by tenants.

- **node-int-net-01**
  A shared tenant network providing DHCP IPv4 (and IPv6 in the future) addresses. This network is visible for all tenants and therefore anyone can attach OpenStack instances on it. Any node could choose its own network range since this should not collide with other node's networks.

There is no limitation in the use of networks and every node can configure additional networks in its OpenStack configuration. If we test this information with the CLI tool we obtain the following result if we execute the following command:

```
$ neutron net-list
```

Or using the more recent version of the CLI, the following command:

```
$ openstack network list
```

The output of networks and subnets should be:

Table 3.4: Example returned values of: openstack network list

Id	Name	subnets
3dccc622-7200-40be-b523-0f73674db0e7	public-ext-net-01	44c356e1-53ad-43ce-b3b7-816bbd1d9529 130.206.82.0/22
b99da016-cb02-4556-8d5f-2ce27a9a861d	node-int-net-01	a250c7a4-4d23-4c9a-85be-3e9b367a00a1 172.16.0.0/20

And if we check the sub-network that we have associated to this network, through the following commands

```
$ neutron subnet-list
```

or

```
$ openstack subnet list
```

we will see something like this for the second network:

Table 3.5: Example returned values of: openstack subnet list

Id	Name	CIDR	Allocation pools
a250c7a4-4d23-4c9a-85be-3e9b367a00a1	node-int-subnet-01	172.16.0.0/20	{"start": "172.16.0.2", "end": "172.16.15.254"}

3.5 Check the Status of OpenStack services

The OpenStack administrators can check the status of the OpenStack services in order to make sure that the cloud platform is up and running properly. OpenStack offers a set of CLI commands to obtain the status of these services

3.5.1 OpenStack Nova

Using the Nova CLI, we can execute the command

$ nova service-list

And the output should contain the table with the Nova services list. The services status should be enabled and the state up. The result obtained should be similar to this one:

```
+----+------------------+------------+----------+----------+-------+------------+-----------------+
| Id | Binary           | Host       | Zone     | Status   | State | Updated_at | Disabled Reason |
+----+------------------+------------+----------+----------+-------+------------+-----------------+
| 1  | nova-conductor   | controller | internal | enabled  | up    | 2017-12-2… | -               |
| 2  | nova-consoleauth | controller | internal | enabled  | up    | 2017-12-2… | -               |
| 3  | nova-cert        | controller | internal | enabled  | up    | 2017-12-2… | -               |
| 4  | nova-scheduler   | controller | internal | enabled  | up    | 2017-12-2… | -               |
| 5  | nova-compute     | compute01  | nova     | enabled  | up    | 2017-12-2… | -               |
| 6  | nova-compute     | compute02  | nova     | enabled  | up    | 2017-12-2… | None            |
| 7  | nova-compute     | compute03  | nova     | enabled  | up    | 2017-12-2… | None            |
| 8  | nova-compute     | compute04  | nova     | enabled  | up    | 2017-12-2… | None            |
+----+------------------+------------+----------+----------+-------+------------+-----------------+
```

3.5.2 OpenStack Neutron

OpenStack neutron is one of the more complex service inside the OpenStack ecosystem, it involves lots of agents and obviously all of them have to be working properly. Besides, it has to be checked on compute node and in controller node:

1. For each compute node:[3]
   Verify that the corresponding neutron-openvswitch-agent is running properly. Could be possible that the name of the plugin could be a little different like neutron-plugin-openvswitch-agent.

```
$ service neutron-openvswitch-agent status

neutron-plugin-openvswitch-agent start/running, process 41300
```

2. For each controller node:
   We should verify that the neutron server and the different neutron agents (metadata, dhcp, l3 and openvswitch) and running properly.

```
$ service neutron-server status

neutron-server start/running, process 63925
```

```
$ for srv in metadata dhcp l3 openvswitch; do service
neutron-srv-agent status; done

neutron-metadata-agent start/running, process 3126

neutron-dhcp-agent start/running, process 13261

neutron-l3-agent start/running, process 13423

neutron-openvswitch-agent start/running, process 21127
```

[3] In this document, we use Ubuntu version 14.04 LTS and the systemv process management tools; in case of Ubuntu 16.04 LTS or CentOS 7, these could be replaced by their systemd equivalents.

3. Only in one controller node
   It is not needed that we execute the following command on every controller node, due to the result should be the same, just take one of the controller node and execute:

   ```
   $ neutron agent-list
   ```

   The output table should list all the neutron agents with the value :-) in the alive column and the value True in the admin_state_up column:

Table 3.6: Example returned values of: neutron agent-list

id	agent_type	host	alive	admin_state_up	binary
3f034de3-6fce-4255-b04d-97149d0895ff	Open vSwitch agent	node-30	:-)	True	neutron-openvswitch-agent
9b6dca03-8001-4a34-b0eb-c1e0d9d4990b	L3 agent	node-30	:-)	True	neutron-l3-agent
b60f7605-6b10-4137-83c8-2473aaaa3eb8	DHCP agent	node-30	:-)	True	neutron-dhcp-agent
d92c1377-c51a-4307-8a75-48df8d251e5d	Open vSwitch agent	node-29	:-)	True	neutron-openvswitch-agent
e4e0aa0b-2698-41c4-a36e-48580aba640f	Open vSwitch agent	node-28	:-)	True	neutron-openvswitch-agent
ec05e2e0-bcfe-48cc-ace5-8c612f164604	Metadata agent	node-30	:-)	True	neutron-metadata-agent

3.5.3 OpenStack Cinder

There are 4 services that have to be checked, depending on the configuration of your OpenStack instance. Those services are cinder-api, cinder-scheduler, cinder-volume and cinder-backup (this one sometimes is not installed in the configuration. Therefore, we have to test.

1. On every controller node
   ```
   $ cinder service-list
   ```

   This produce the following content:

Table 3.7: Example returned values of: cinder service-list

Binary	Host	Zone	Status	State	Updated_at	Disabled Reasons
cinder-scheduler	node-30	nova	enabled	up	2015-11-25T16:14:12.00	None
cinder-volume	node-29	nova	enabled	up	2015-11-25T16:14:10.00	None
cinder-volume	node-28	nova	enabled	up	2015-11-25T16:14:11.00	--

   All the services should be enabled in the Status column and up in the State column.

2. On every node with the Cinder role, run:
   ```
   $ for srv in volume backup; do service cinder-$srv status; done

   cinder-volume start/running, process 802

   cinder-backup: process 775
   ```

   It is possible that in your configuration you not have the cinder-backup service and in the other side you have there the cinder-api and cinder-scheduler service up and running.

3.5.4 OpenStack Ceilometer

The installation of ceilometer is specific to every OpenStack instance. Nevertheless, we put here some indications about a common approach to configuring this service in the different nodes.

1. On every instance in which we have running a MongoDB node (we recommend using always a MongoDB and not a MySQL DB), run:

   ```
   $ netstat -nltp | grep mongo
   ```

   The output of the netstat command returns the local IP addresses and ports in the LISTEN status.

2. On every controller node, run:

   ```
   $ for srv in agent-central api agent-notification collector; do
   service ceilometer-$srv status; done
   ```

   It should return that all the services are running and the corresponding PID

   ```
   ceilometer-agent-central start/running, process 35930
   ceilometer-api start/running, process 36014
   ceilometer-agent-notification start/running, process 35955
   ceilometer-collector start/running, process 36061
   ```

3. On every compute node, run:

   ```
   $ service ceilometer-polling status
   ceilometer-polling start/running, process 26435
   ```

3.5.5 OpenStack Glance

For glance, we can execute the following command on every controller node:

```
$ for srv in api registry; do service glance-$srv status; done
```

this should return an indicator that both services are up and running which we can see here:

```
glance-api start/running, process 55927

glance-registry start/running, process 13827
```

3.5.6 (Optional) OpenStack Swift

In case of Swift, it is also similar to OpenStack Ceilometer service. Firstly, it is not a mandatory service to be installed on all the nodes. Even in the case that it is installed, all of the Swift services may not be installed; in some cases, the swift api is installed and integrated with a storage backend and in other cases, just ceph. In this context, we provide the way to check the swift services if you only install OpenStack Swift and the ceph in case that you are using it.

- To check OpenStack services, on every control node, just run:

  ```
  $ for srv in account-auditor account account-reaper account-replicator container-auditor container container-reconciler container-replicator container-sync container-updater object-auditor object object-reconstructor object-replicator object-updater proxy; do service swift-$srv status; done
  ```

  It should provide information about all the services up and running from swift.

- To check ceph processes, just execute on each of the ceph nodes:

  ```
  $ ceph status
  ```

  And the result should be something similar to the following

  ```
    cluster 53aa1403-f5b4-477b-bf7f-c4899c79eaf1
       health HEALTH_OK
       monmap e1: 3 mons at {node-50=172.132.10.278:1234/0,node-51=172.132.10.253:1234/0,node-52=172.132.10.254:1234/0}
              election epoch 938, quorum 0,1,2 node-50,node-51,node-52
       osdmap e787: 4 osds: 4 up, 4 in
        pgmap v1612553: 280 pgs, 13 pools, 30523 MB data, 20057 objects
              130 GB used, 19981 GB / 20111 GB avail
                   280 active+clean
  ```

3.5.7 RabbitMQ

In this section, we want to check the RabbitMQ services for that purpose we have to execute on each of the controller nodes the following command:

```
$ rabbitmqctl cluster_status
```

The output should show the list of running nodes into the field running_nodes with the rabbit@<HOSTNAME> format, besides, the partitions field should be empty. Example output is shown below:

```
Cluster status of node 'rabbit@node-30' ...
[{nodes,[{disc,['rabbit@node-30']}]},
 {running_nodes,['rabbit@node-30']},
 {partitions,[]}]
...done.
```

3.6 Testing FIWARE Lab Node locally

In order to ensure that the OpenStack instance is installed property, a set of manual tests are executed by the FIWARE Lab team in order to guarantee the correct execution of them. Those E2E tests are based on the historical problems that we found in the FIWARE Lab nodes once it was up and ready in the past. Some of these problems include no route to the virtual machine, unable to execute the cloudinit process or unable to delete properly the allocated resources of a user. Therefore, these E2E tests cover all the problematic scenarios that arose during the FIWARE Lab existence.

The tests are divided into Mandatory and Optional. This is because not all FIWARE Lab nodes implement the Object Storage components and hence these tests are not required for all nodes. The language that is used to specify those scenarios is the Gherkin language[4]. Gherkin is plain-text language with well-designed structure to define the behaviour of a system. It is designed to the learn and use in a very easy way not only by programmers. Besides, it can describe concisely examples to describe business rules in most real-world domains. An example of Gherkin document is:

[4] https://cucumber.io/docs/reference

Table 3.8: Example of Gherkin language

```
Feature: Access to the OpenStack instance

  Scenario: Registering a user into OpenStack
          Given  the OpenStack account name of a valid user
          And    the valid password of this user
          When   the user requests access to the OpenStack instance with
those data
          Then   the OpenStack account service (Keystone) authenticates the
user
          And    Keystone redirect to the OpenStack Portal (Horizon).
```

The main keywords In Gherkin are:

- Feature
- Scenario
- Given, When, Then, And, But (Steps)
- Background
- Scenario outline
- Examples

We will take different scenarios in order to complete the lists of tests that we want to apply for each new FIWARE Lab node. Currently those tests are executed manually but It does not mean that it will continue in a manual way, In the future, we will probably adopt a Behaviour Driven Development solution to automatically execute those tests.

3.6.1 Feature 1: Creation and management of Security groups

In this scenario, we want to test the basic operations related to the creation and management of security groups. It is comprised of the following scenarios:

Table 3.9: Feature about the creation and management of Security Groups

```
Feature 1: Creation and management of Security groups

  Scenario 1.1: Create a Security Group
          Given  a registered user in OpenStack environment.
          When   the user requests creation of a Security Group with name
“SGtest”.
          Then   OpenStack creates the corresponding Security Group with
the name “SGtest”.
          And    this Security Group has no rules assigned to it.

  Scenario 1.2: Create a Security Group Rules for a Security Group
```

```
          Given  a registered user in OpenStack environment.
          And    a Security Group previously created with name “SGtest”.
          When   the user requests creation of a Security Group Rule for
SSH access.
          Then   the OpenStack creates the corresponding Security Group
Rules associated to the Security Group “test”.
          And    this Security Group has the corresponding Security Group
Rule associated to it.
```

3.6.2 Feature 2: Creation of key pair

In this scenario, we want to test the basic operations related to the creation and management of key pair content. It is comprised of the following scenario:

Table 3.10: Feature about the creation of key pair

```
Feature 2: Creation of key pair

  Scenario 2.1: Create a Key Pair
          Given  a registered user in OpenStack environment.
          When   the user requests creation of a KeyPair with the name
“testKeyPair”.
          Then   the OpenStack creates the corresponding KeyPair.
          And    the user can download it.
```

3.6.3 Feature 3: Allocate and associate Floating IPs

In this scenario, we want to test the basic operations related to the allocation and association of Floating IPs to a specific user. It is compound the by following scenarios:

Table 3.11: Feature about the allocation and association of floating IPs

```
Feature 3: Allocate and associate Floating IPs

  Scenario 3.1: Allocate an IP to the user project
          Given  a registered user in OpenStack environment.
          When   the user requests the allocation of an IP from the pool
“public-ext-net-01”.
          Then   the OpenStack allocates an IP.
```

```
  Scenario 3.2: Associate a floating IP to a Virtual Machine
          Given  a registered user in OpenStack environment.
          And    a previous deployed virtual machine with name “testVM”.
          And    a previous allocate public IP.
          When   the user requests the association of the floating IP to a
“testVM”.
          Then   the OpenStack associates the floating IP with the
corresponding Virtual Machine.
```

3.6.4 Feature 4: Creation of network and associate to a Virtual Machine

In this scenario, we want to test the basic operations related to the creation of networks and routers and associate them to a virtual machine in order to tests if the neutron service is properly configured. It is comprised of the following scenarios:

Table 3.12: Feature about creation of network and associate to a Virtual Machine

```
Feature 4: Creation of network and associate to a Virtual Machine

  Scenario 4.1: Create a network with its corresponding subnetwork
          Given  a registered user in OpenStack environment.
          When   the user requests the creation of the network
“networktest”.
          And    the addition of subnet with name “subnettest”, network
address “195.134.187.0/10” and DNS Name Server “8.8.8.8”.
          Then   the Neutron service creates the network with the
corresponding subnetwork.

  Scenario 4.2: Create a new router
          Given  a registered user in OpenStack environment.
          When   the user requests the creation of a router “routertest”.
          Then   the OpenStack creates a new router.

  Scenario 4.3: Assign an interface to the previous router
          Given  a registered user in OpenStack environment.
          And    a previous created router “routertest”.
          And    a previous deployed network “networktest” with subnetwork
“subnettest”.
          When   the user requests the addition of a new interface to the
router.
          And    select this network and subnetwork.
          Then   the OpenStack adds the interface to the router.

  Scenario 4.4: Set a gateway to the previous router
```

```
          Given  a registered user in OpenStack environment.
          And    a previous created router “routertest”.
          And    a public access network “public-ext-net-01”.
          When   the user requests to set the Gateway to this external
network “public-ext-net-01”.
          Then   the OpenStack sets the Gateway to this router.

  Scenario 4.5: Check the access to a virtual machine
          Given  a registered user in OpenStack environment.
          And    a previous created network “networktest”.
          And    a router “reoutertest” properly configured in terms of
gateway and interface.
          When   the user requests to create a virtual machine associated
to this network.
          Then   the OpenStack allocates the virtual machine.
          And    the user can access through SSH to the new instance.
```

3.6.5 Feature 5: Working with new Instances

In this scenario, we want to test if a new instance can be deployed and it is correctly managed in terms of network connectivity. We also check access to the OpenStack metadata service in order to receive different actions from it. Last but not least, the test will include the communications that we have to do from the instance to the world in order to be sure that the network interfaces are correctly configured in both directions. It is comprised of the following scenarios:

Table 3.13: Feature about working with instances

```
Feature 5: Working with Virtual Machines

  Scenario 5.1: Create a new Virtual Machine
          Given  a registered user in OpenStack environment.
          And    a base image with the name “base_ubuntu_14.04” exists.
          And    a flavor “m1.small” exists.
          And    a Key pair “testKeyPair” exists.
          And    a Security Group “SGtest” exists.
          And    a network “node-int-net-01” exists.
          When   the user requests the creation of the instance name
“testinstance”.
          And    the previous Flavor, Key pair, Security Group and network
are selected.
          Then   the OpenStack deploy a virtual machine with that name.

  Scenario 5.2: Delete a virtual machine
          Given  a registered user in OpenStack environment.
```

```
          And    a virtual machine with the name “testinstance” exists.
          When   the user requests the deletion of this virtual machine.
          Then   the OpenStack deletes the virtual machine without any
problem.

  Scenario 5.3: Create again a new Virtual Machine
          Given  a registered user in OpenStack environment.
          And    a base image with the name “base_ubuntu_14.04” exists.
          And    a flavor “m1.small” exists.
          And    a Key pair “testKeyPair” exists.
          And    a Security Group “SGtest” exists.
          And    a network “node-int-net-01” exists.
          And    a virtual machine with the name “testinstance” was
previously deleted.
          When   the user requests the creation of the instance name
“testinstance”.
          And    the previous Flavor, Key pair, Security Group and network
are selected.
          Then   the OpenStack deploy a virtual machine with that name.

  Scenario 5.4: Create a Snapshot from a virtual machine
          Given  a registered user in OpenStack environment.
          And    a virtual machine with the name “testinstance” exists.
          When   the user requests the creation of a snapshot of that image
with name “demo-instance-snapshot”.
          Then   the OpenStack creates the snapshot with this name.

  Scenario 5.5: Create a virtual machine from a snapshot image
          Given  a registered user in OpenStack environment.
          And    a previous snapshot with the name “demo-instance-
snapshot”.
          And    a flavor “m1.small” exists.
          And    a Key pair “testKeyPair” exists.
          And    a Security Group “SGtest” exists.
          And    a network “node-int-net-01” exists.
          And    a virtual machine with the name “testinstance” does not
exist.
          When   the user requests the creation of the instance name
“testinstance2”.
          And    the previous Flavor, Key pair, Security Group and network
are selected.
          Then   the OpenStack deploy a virtual machine with that name.

  Scenario 5.6: Check the access to a virtual machine created from an image
          Given  a registered user in OpenStack environment.
          And    a previously created virtual machine “testinstance2”
exists.
          And    a public IP is available from the pool “public-ext-net-
01”.
```

```
          When   the user requests to associate this public IP to the
virtual machine.
          Then   the OpenStack associates them.
          And    the user can access through SSH to the new instance.

  Scenario 5.7: Check the access to the metadata service from a virtual
machine
          Given  a registered user in OpenStack environment.
          And    a previously created virtual machine “testinstance2”
exists.
          And    this virtual machine is accessible using SSH.
          When   the user accesses to the virtual machine using SSH.
          And    request the access to the metadata service through the
command “curl http://169.254.169.254/1.0/meta-data”.
          Then   command line responses with the corresponding information
about the metadata service.

  Scenario 5.8: Check the access to the world from a deployed virtual
machine
          Given  a registered user in OpenStack environment.
          And    a previously created virtual machine “testinstance2”
exists.
          And    this virtual machine is accessible using SSH.
          When   the user accesses to the virtual machine using SSH.
          And    the user executes a simple ping command “ping
www.google.com”.
          Then   ping command responses correctly with the information of
this server.
```

3.6.6 Feature 6: Working with volumes

In this scenario, we want to test the creation and association of volumes work properly on the new OpenStack environment. It is comprised of the following scenarios:

Table 3.14: Feature about working with volumes

```
Feature 6: Check volume management

  Scenario 6.1: Check the creation of a volume
          Given  a registered user in OpenStack environment.
          When   the user requests the creation of a volume with the name
“testvolume”.
          And    the description “a test volume to be deleted”.
          And    a size of 1 Gb.
          Then   the OpenStack creates the volume properly with status
```

```
“available”.

  Scenario 6.2: Delete a volume
          Given  a registered user in OpenStack environment.
          And    a volume with name “testvolume” exists.
          When   the user requests the deletion of this volume.
          Then   the OpenStack deletes the volume properly.

  Scenario 6.3: Attach a volume to an instance
          Given  a registered user in OpenStack environment.
          And    a virtual machine with the name “testinstance2” exists.
          And    a volume with the name “testvolume” is created properly.
          When   the user tries to attach the volume to the instance.
          Then   the OpenStack attaches the volume to the instance.
          And    the status is “in-use”.
          And    the attachment is “1”.

  Scenario 6.4: Check a volume to an instance
          Given  a registered user in OpenStack environment.
          And    a virtual machine with the name “testinstance2” exists.
          And    a volume with the name “testvolume” is attached to this
virtual machine.
          When   the user accesses to the virtual machine through SSH
client.
          And    execute the command “sudo fdisk -l”.
          Then   the commands response with the current partition table.
          And    it contains the description of a disk “Disk /dev/vdb: 1073
MB, 1073741824 bytes”.
          And    it says us “Disk /dev/vdb doesn't contain a valid
partition table” which means that it is ready to mount this new partition.
```

3.6.7 Feature 7: Check list of images

In this scenario, we want to test if the node has installed the list of base images. They are the only base images that are allowed to be used in the FIWARE Lab environment for security reasons. It is comprised of the following scenario:

Table 3.15: Feature about checking the list of images

```
Feature 7: Check image list

  Scenario 7.1: Check the current list of image list
          Given  a registered user in OpenStack environment.
          When   the user requests the list of available images.
```

```
          Then   the OpenStack response with the list of available images.
          And    it should contain “base_ubuntu_16.04”, “base_ubuntu_14.04
     “, “base_debian_8” and “base_centos_7”.
```

3.6.8 Feature 8: Object Storage management

In this scenario, we want to test if we can work with object storage service. This is an optional test and only will be executed if the FIWARE Lab node currently install the corresponding service. It is comprised of the following scenario:

Table 3.16: Feature about object storage management

```
Feature 8: Object Storage management

  Scenario 8.1: Create a container
          Given  a registered user in OpenStack environment.
          When   the user requests the creation of a new container with the
name “testcontainer”.
          Then   the OpenStack creates the container.
          And    the number of Objects is “0”.
          And    the Size is “0 bytes”.

  Scenario 8.2: Upload a text file on a container
          Given  a registered user in OpenStack environment.
          And    a container with the name “testcontainer” exists.
          And    this container is empty.
          When   the user requests the upload of a simple text file on this
container.
          And    the name is “testfile”.
          Then   the OpenStack uploads the file into the container.
          And    the after 1 minute, the Objects number change to “1”.
          And    the Size is not “0 bytes”.

  Scenario 8.3: Download an object from a Container
          Given  a registered user in OpenStack environment.
          And    a container with the name “testcontainer” exists.
          And    an object with the name “testfile” exists.
          When   the user requests the download of the object.
          Then   the OpenStack downloads the object.
          And    the user can open properly and see the content of the
file.

  Scenario 8.4: Delete an object from a Container
          Given  a registered user in OpenStack environment.
```

```
        And    a container with the name “testcontainer” exists.
        And    an object with the name “testfile” exists.
        And    this is the only object on that container.
        When   the user requests the deletion of the object.
        Then   the OpenStack deletes the object.
        And    the after 1 minute, the Objects number change to “0”.
        And    the Size is “0 bytes”.

Scenario 8.5: Delete a Container
        Given  a registered user in OpenStack environment.
        And    a container with the name “testcontainer” exists.
        And    this container is empty.
        When   the user requests the deletion of the container.
        Then   the OpenStack deletes the container.
```

3.6.9 Report the results of the execution of the tests

The report consists of a detailed description of the manual tests provided to the node together with the table of the results of the execution of those tests. As it is possible that there may be several errors in different iterative steps, the report will keep the previous result of the execution of the tests.

Sometimes if the number of errors is very high, we can decide to stop the execution of the tests and inform to the future FIWARE Lab node that they have to resolve the current errors in order to follow with the manual execution of the tests.

Additionally, the results will include some comments about the problem that was found and in same case, if the reporter knows the solution, instructions to correct the problem and pass the scenario’s test.

You can see in the Annex A: Template of report of the Testing FIWARE Lab Node locally, the format of this report.

3.7 Connecting your Node to FIWARE Lab

This section contains a guide of the necessary steps to connect your Node to FIWARE Lab.

3.7.1 Prerequisites

Before starting the process of connecting your node to FIWARE Lab you have to accomplish the following:

- To be an authorized node to join FIWARE Lab, you can get this authorization by filling the corresponding document “Become a new FIWARE Lab node” see Annex D and send it to FIWARE Lab administrator (fernando dot lopez at fiware dot org).
- To have every OpenStack service properly working locally (using your local Keystone)
- To have the public endpoints of your OpenStack services opened to the Internet.

3.7.2 Connection process

Creating your "admin" and "services" accounts

The first thing you need to start registering your node is to have the following accounts registered in FIWARE Lab Keystone:

- An "admin" account: it will be used for administrative tasks such as communicating with the users of your node or managing the users accounts status.
- A "service" account for each OpenStack service you are providing: these accounts are used to configure the OpenStack services and allow them to validate tokens with the FIWARE Lab Keystone.

To create all these accounts, you have to contact FIWARE Lab Keystone administrators providing the following data:

- Name of your node
- List of services you wish to register
- Email address of the node administrator

FIWARE Lab administrators will provide you the usernames/passwords for all the service accounts. They will also provide you the '"authorization endpoint"' you have to use in the step "Configuring your services".

Registering your services endpoints

The second step is to register your node endpoints in the FIWARE Lab Service Catalogue. To do that you have to contact FIWARE Lab Keystone administrators providing them the list of endpoints of all your OpenStack services.

The list of the endpoints has to follow the following template:

- Nova: (service type: compute, service name: nova)

```
"adminURL": "http://<IP_ADRRESS>:8774/v2/$(tenant_id)s"

"internalURL": "http://<IP_ADRRESS>:8774/v2/$(tenant_id)s"

"publicURL": "http://<IP_ADRRESS>:8774/v2/$(tenant_id)s"
```

- Glance: (service type: image, service name: image)

```
"adminURL": "http://<IP_ADRRESS>:9292/v1"

"internalURL": "http://<IP_ADRRESS>:9292/v1"

"publicURL": "http://<IP_ADRRESS>:9292/v1"
```

- Volume: (service type: volume, service name: cinder)

```
"adminURL": "http://<IP_ADRRESS>:8776/v1/$(tenant_id)s"

"internalURL": "http://<IP_ADRRESS>:8776/v1/$(tenant_id)s"

"publicURL": "http://<IP_ADRRESS>:8776/v1/$(tenant_id)s"
```

- Network: (service type: network, service name: neutron)

```
"adminURL": "http://<IP_ADRRESS>:9696/"

"internalURL": "http://<IP_ADRRESS>:9696/"

"publicURL": "http://<IP_ADRRESS>:9696/"
```

- Object Store: (service type: object-store, service name: swift)

```
"adminURL": "http://<IP_ADRRESS>:8090/v1"

“internalURL": "http://<IP_ADRRESS>:8090/v1/AUTH_$(tenant_id)s"

"publicURL": "http://<IP_ADRRESS>:8090/v1/AUTH_$(tenant_id)s"
```

Configuring your services

Once you have your service accounts and passwords you can proceed with the service configuration. What you have to do is to change the Keystone service to which they are validating the tokens. Currently your OpenStack services are connected to your local Keystone. You have to connect them to the FIWARE Lab Keystone.

To do so, you have to modify the configuration files of each service changing the following parameters:

- **`auth_uri`**: You have to update it with the authorization endpoint administrators have provided you when registering your account (in the step "Creating your admin and services accounts")
- **`admin_tenant_name`**: "service"
- **`admin_user`**: the username of this service account
- **`admin_password`**: the password of this service account

Below is a list of configuration files that need to be changed:

- `/etc/nova/nova.conf`
- `/etc/neutron/dhcp_agent.ini`
- `/etc/neutron/metadata_agent.ini`
- `/etc/neutron/neutron.conf`
- `/etc/neutron/l3_agent.ini`
- `/etc/neutron/api-paste.ini`
- `/etc/glance/glance-api.conf`
- `/etc/glance/glance-registry.conf`
- `/etc/glance/glance-cache.conf`
- `/etc/cinder/cinder.conf`
- `/etc/swift/proxy-server.conf`

Some of these files can be also hosted in the Controller and in the Compute nodes. It depends on the OpenStack configuration adopted by each Infrastructure Owner. For example, you can install

OpenStack Networking in the Controller server or in a dedicated one. Usually in Compute nodes you have only to change the nova configuration file.

Once these files are updated you should restart the corresponding services.

Validating the registration

Once all the steps are done, you have to check that the connection with the new Keystone is working properly. To do that you have to use the CLI tools to check:

- you can deploy a virtual machine
- you can deploy an image (deploy an image is equal to register as image in glance)
- you can deploy a network
- you can attach a floating IP to a VM and it is reachable from outside

Keep in mind that if we want to use CLI tools, some environment variables must be first exported:

```
export OS_REGION_NAME="you_region_name"
export OS_USERNAME=”your_user_name”
export OS_PASSWORD=”your_password”
export OS_AUTH_URL=http://cloud.lab.fiware.org:4730/v3
export OS_PROJECT_NAME=admin
export OS_PROJECT_DOMAIN_ID=default
export OS_USER_DOMAIN_ID=default
export OS_IDENTITY_API_VERSION=3
```

3.7.3 Node publication in FIWARE Lab

Once the registration process is finished and you have correctly validated it using the CLI tools, your node is ready for publication in FIWARE Lab.

To publish it you have to contact the FIWARE Lab Keystone administrators telling them you have finished the process. They will check everything is ok and the will activate your node in the FIWARE Lab Cloud Portal and in the FIWARE Lab Account Portal (for administrators).

3.7.4 Synchronization of Glance Images

Once your node is ready and working, the Glance Images need to be synchronized. These images include some base Operating System images (CentOS, Ubuntu, Debian) with some security enhancements and some images containing the Basic Generic Enablers needed to build applications based on FIWARE (i.e. Orion Context Broker, CKAN, Keyrock IDM, etc).

To do that you have to contact FIWARE Lab Spain2 node administrators providing them the credentials that should be used to synchronize the images. The first synchronization is a long process since it has to upload many large files.

3.8 Registering your Node in FI-Health

Registering a new FIWARE Lab node in the FI-Health tool is a straightforward operation. You only need to provide the following data:

- Name of the FIWARE Lab node, which should be the same as that provided in the FIWARE Lab Keystone service (see section 3.7, Connecting your Node to FIWARE Lab).
- Name of the public external network that it is configured in the FIWARE Lab node.
- Name of the shared network.
- If you have installed Object Storage in order to activate those tests.

The best way to proceed with it is creating a FI-Health Issue[5] inside the GitHub project this can be managed inside the code repository.

Once the FI-Health tool executes the tests for the new FIWARE Lab node, you can access the application using your admin account. This admin account will be created in the Connecting your Node to FIWARE Lab phase (see previous section) and will have the format "**admin-<name of the FIWARE Lab node>**". This account gives you the possibility to relaunch the tests if you need to check if some problems was resolved after administrator intervention in the FIWARE Lab node.

3.9 Registering your Node in the Infographics

The current FIWARE monitoring system is based on four stacked macro components, some of them run on every FIWARE Lab node:

- **Ceilometer:** https://wiki.openstack.org/wiki/Ceilometer
- **Ceilometer FIWARE plugin:** https://github.com/SmartInfrastructures/ceilometer-plugin-fiware

others run centrally on FIWARE Lab premises:

[5] https://github.com/telefonicaid/fiware-health/issues

- **Monasca:** https://wiki.openstack.org/wiki/Monasca
- **MonitoringAPI:** https://github.com/SmartInfrastructures/FIWARELab-monitoringAPI
- **Infographic GUI:** https://github.com/SmartInfrastructures/fi-lab-infographics

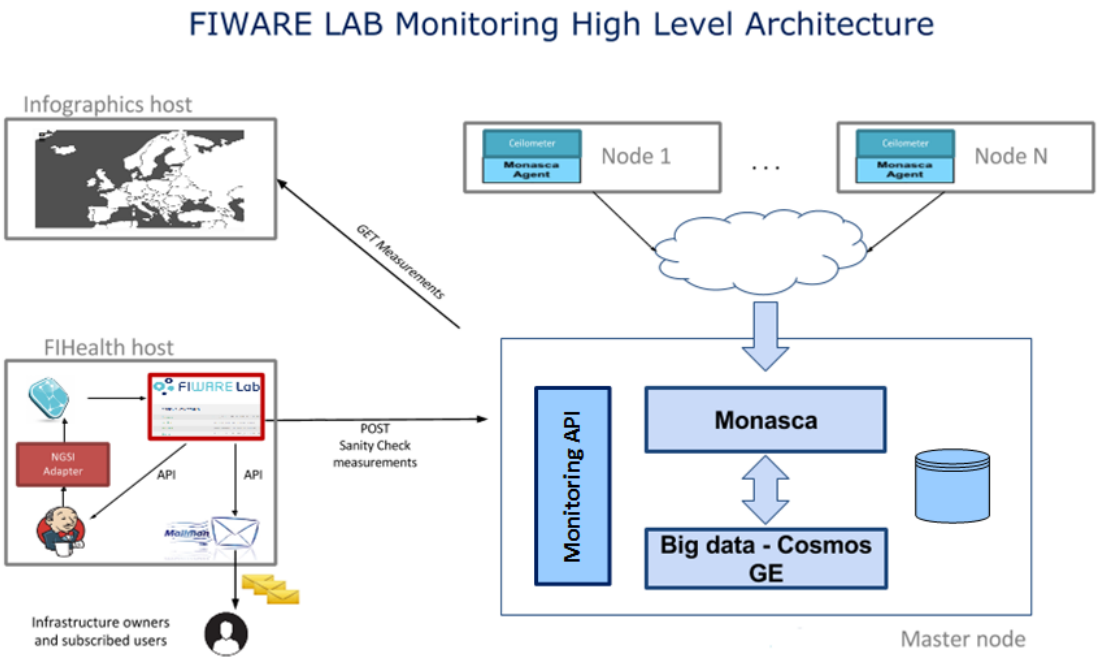


Figure 3.2: FIWARE Lab Monitoring high level architecture.

Monitoring information collected by Ceilometer on each FIWARE Lab node flows to the central Monasca API server for persistency, they are then retrieved by the FIWARE Infographics GUI and FIWARE Lab Status page using the FIWARE monitoring API that, in turn, provide aggregation and authentication over the collected measurements.

In order to register a new FIWARE Lab node in the Infographic GUI some required steps must be respected by Infrastructure Owner of the node and must be successfully completed in the following order, as there are dependencies:

1) Ceilometer must be installed and federated as explained in section [How to setup a FIWARE Lab Node]. Moreover, it should be able to collect OpenStack default monitoring data locally.

2) Ceilometer FIWARE plugin must be properly installed and configured on the node. It should be able to push all the FIWARE custom measurements to the Monasca API server. Detailed and constantly updated instruction can be found on the GitHub repository under the branch that correspond with the running OpenStack version: https://github.com/SmartInfrastructures/ceilometer-plugin-fiware.

3) Intention to publish the node on Infographic GUI must be notified during the FIWARE Lab node weekly meeting in order to enable the new node on a testing Infographic GUI. This allows to check that collected measurements are coherent with the real node status.

4) Once the testing Infographic GUI reports correct data from the node, it will be published on the main GUI and the new FIWARE Lab node will be visible on both FIWARE Infographic and Status page GUI respectively at the following addresses: http://infographic.lab.fiware.org and http://status.lab.fiware.org.

3.10 Registering your Node in Deep Log Inspection

FIWARE Lab offers a centralised solution for Log Inspection to help node admins detect anomalies in the behaviour of their services or of the users. The solution is based on ElasticSearch ecosystem. The adoption of the deep log inspection is optional, but it is highly recommended to help you in the management of your node.

Node admins will need only to configure a syslog server and configure OpenStack services to forward their logs to it.

The syslog server used as default solution is the Monasca Log Agent (https://github.com/logstash-plugins/logstash-output-monasca_log_api). The server can be installed on an existing server (such as the OpenStack node controllers) or a Virtual Machine provided inside the node.

Detailed instructions are provided in this guide:

http://deep-log-inspection.readthedocs.io/en/latest/install/monasca-log-agent/

The actual user and password to connect your log agent will be provided by FIWARE Lab team, you will need to open an issue requesting it in Jira.

4 Operating and Maintaining a FIWARE Lab node

In this section, we include the different tools and procedures that can be used in order to operate your FIWARE Lab node and try to resolve or find any possible problem on it. Those tools are used to monitor FIWARE Lab usage and availability as well as provide statistics and evidences of FIWARE Lab activities. Last but not least, we introduce the tool that we use to get the status of the Sprint Planning and Backlog status of your node.

4.1 Deep Log Inspection

Deep Log Inspection, if you completed the registration of your logs in the FIWARE Lab service, allows you to analyse the logs. The service is not yet in production, but it will be available soon at: https://kibana.deeplogmanager.lab.fiware.org.

At this URL admins will be able to search in their logs any type of issue or event (see interface) related to their node.

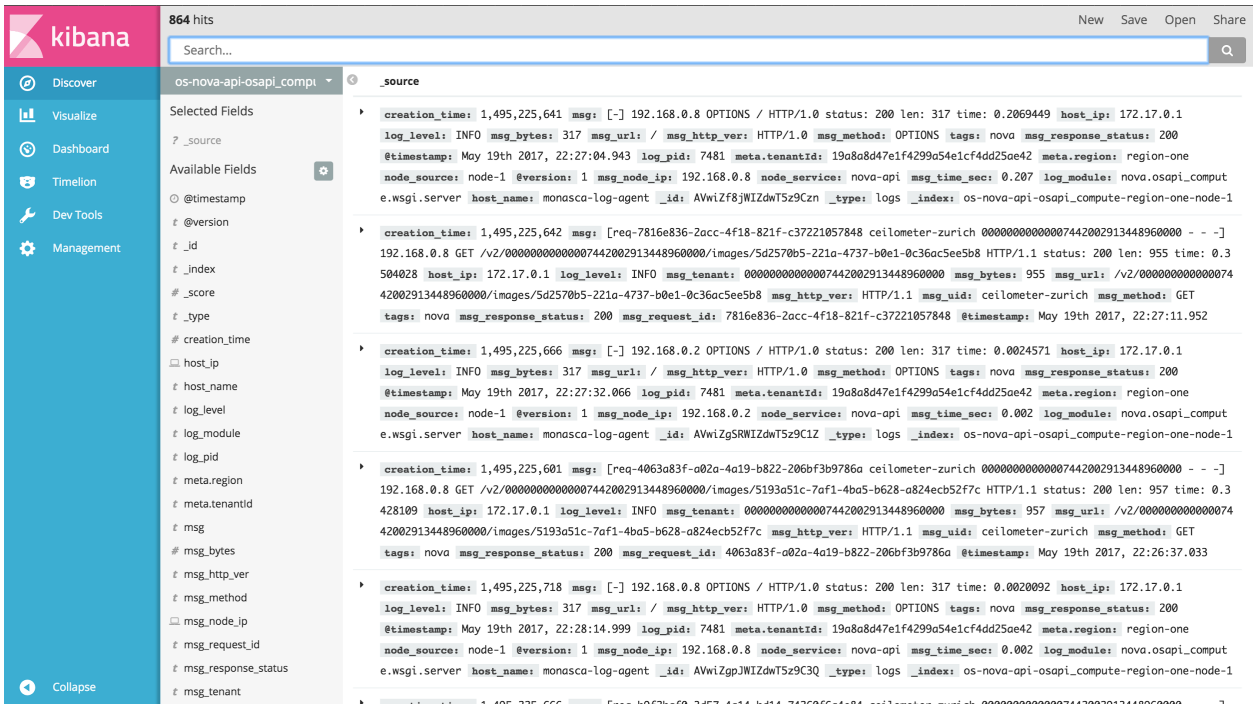


Figure 4.1: Deep Log Inspection tool.

Beyond that a number of predefined queries and visualizations are available to simplify admins routine analysis over the logs. You can find a user manual at this link: http://deep-log-inspection.readthedocs.io/en/latest/

4.2 FI-Health

FI-Health is an end-to-end testing tool that ensures that the OpenStack integrated components is working as expected. Each of the FIWARE Lab node is tested in a real-world scenario using the OpenStack Rest API provided by the federated Keystone service endpoints. The idea behind these tests is to provide end-to-end testing of the FIWARE Cloud Portal to test whether the flow of the application is performing as designed from start to finish. The purpose of carrying out these end-to-end tests is to identify OpenStack services problems before or at the same time that the users can observe those problems in the FIWARE Cloud Portal. Therefore, FI-Health is the key component to provide quality of service in the FIWARE Cloud environment.

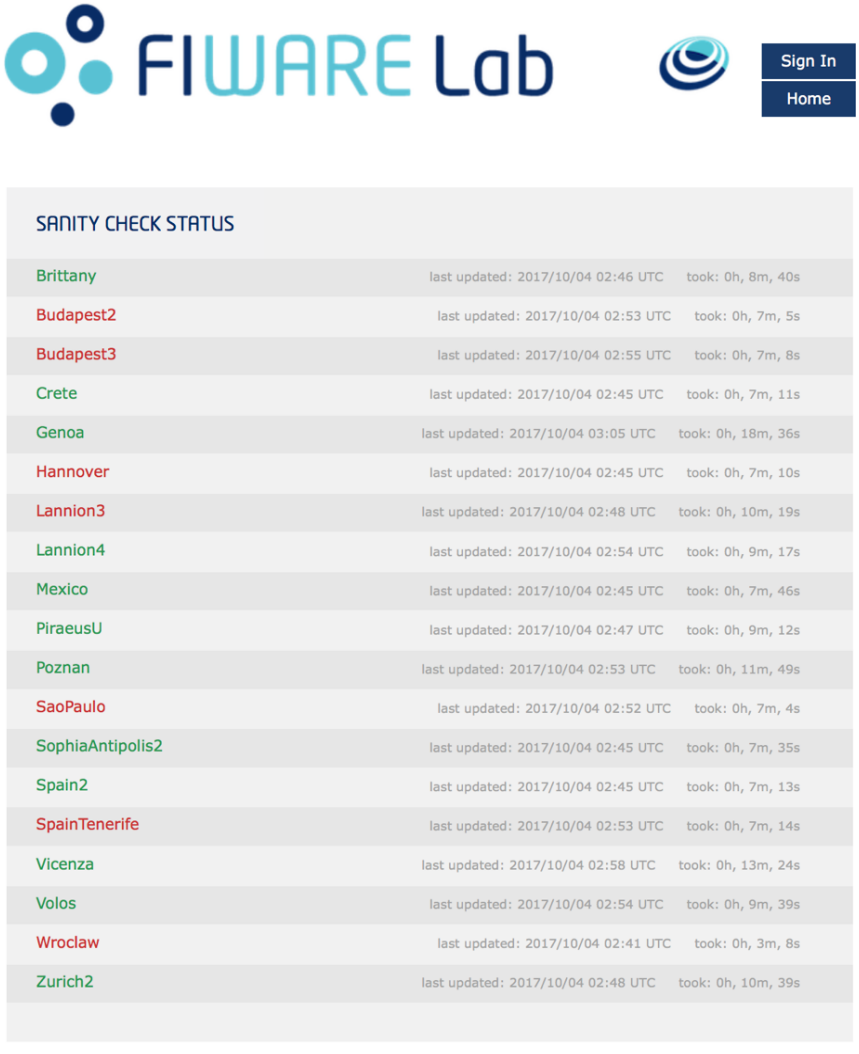


Figure 4.2: Sanity Check tool.

You can obtain more information about the component by accessing the FIWARE Health GitHub repository. Note that if you want to increase or improve this tool you can suggest new issues in this GitHub repository.

Additionally, you can install and check this tool to check locally your FIWARE Lab node. In that case, you should follow the indications in the FIWARE Health Dashboard readme and FIWARE Health Sanity Checks readme.

4.3 FIWARE Lab Infographic

FIWARE Lab Infographics and Status Pages are simple but important services to allow users to:

- Know in an intuitive way the infrastructure capacities made available by FIWARE Lab infrastructure;
- Monitor current status of infrastructure services and know about any issue in any node of FIWARE Lab infrastructure.

While the information on infrastructure capacities is more related to marketing, the one on services status is extremely important to support Developers and Federation Managers operations. The production URL of the Infographics is the following:

- http://infographic.lab.fiware.org/
- http://status.lab.fi-ware.org/

This component provides capacity information about the different OpenStack regions and status information of infrastructure services. The Federation Monitor component exposes data on the capacity and status of FIWARE Lab infrastructure through a set of RESTful API that Infographics and Status Pages calls in order to get that data.

For more details, it is recommended to refer to the main documentation included in the software repository: https://github.com/SmartInfrastructures/fi-lab-infographics.

Figure 4.3 shows the main page of the Infographics tool. It is comprised of:

- a user's section: it shows the total number of users as sum of Basic Users, Trial Users and Community Users;
- a map that shows all nodes;
- six interactive tabs that display different data;
- the list of FIWARE Lab Capacity supporters;

When data about a specific node are obsolete, the node in the map is grey and the user can check the timestamp of its last update by passing the cursor over it.

Figure 4.3: FIWARE Infographics main page.

The Status page (Figure 4.4) depicted below is composed by:

- a big map that shows all the nodes and their overall status (green, yellow or red);
- a table showing the current status of all the FIWARE Lab nodes services together with the FIHealth Sanity Check status (FIHealth Sanity Check executes periodically a set of tests on the nodes in order to verify if the basic functionalities are guaranteed);
- a histogram that shows the average on the last selected months of the Sanity Check status for each node.

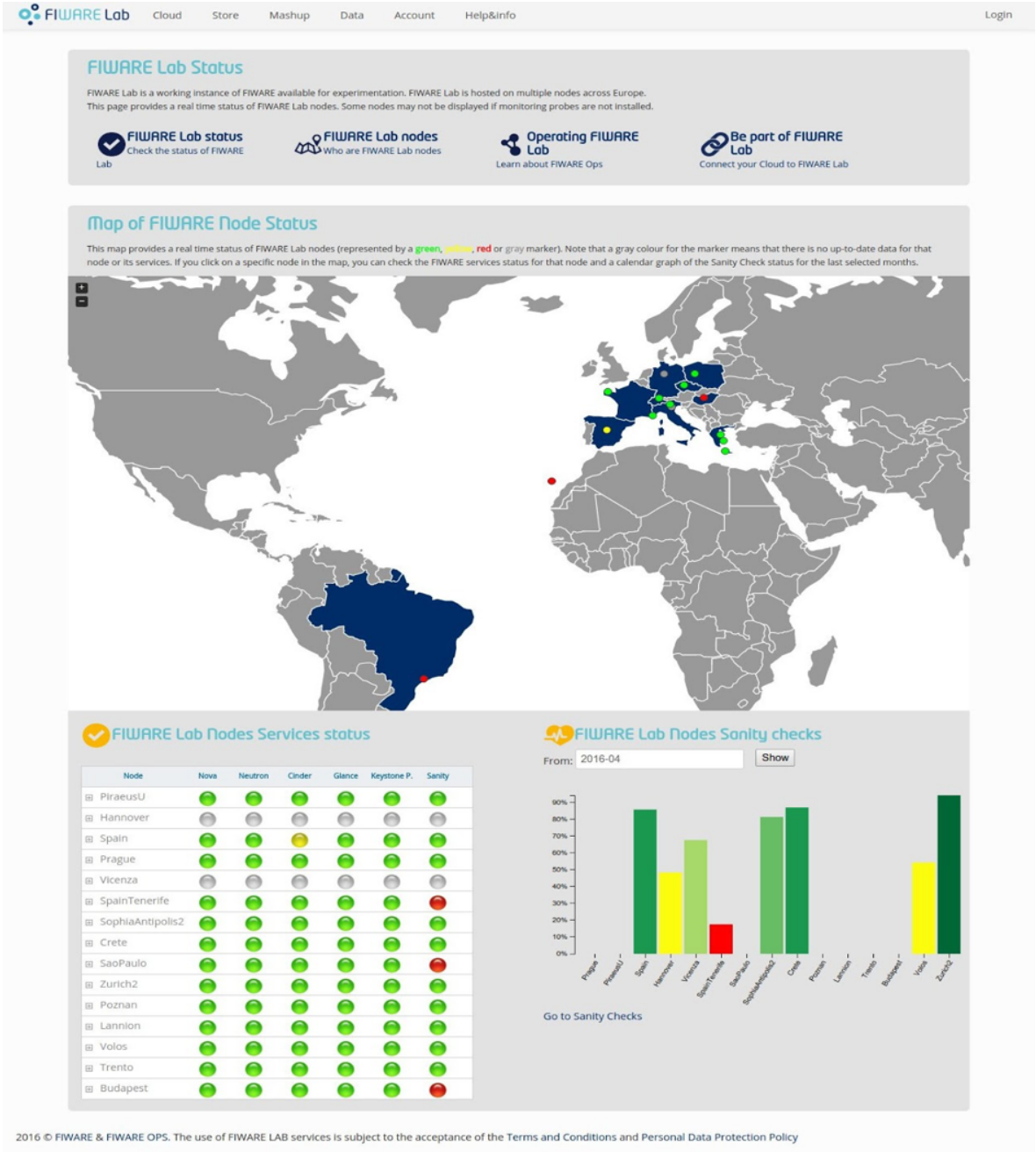


Figure 4.4: FIWARE Status main page

4.4 Rancher

FIWARE Ops team is currently deploying a customised version of Rancher 1.6 (https://rancher.com/docs/rancher/v1.6/en/) on top of FIWARE Lab. This tool allows users with a Cloud account in FIWARE Lab to deploy and manage a Docker Swarm cluster on FIWARE Lab. Through which they can deploy their FIWARE applications on the Lab using docker containers. This is a service offered to the users, and as such FIWARE Lab Node Administrators do not have much to do. They only need to be sure that:

- Their OpenStack APIs are reachable from FIWARE Lab servers (which should be the default case).
- Provide the configuration of MTU within their node to users interested into using the service (or even installing manually Docker Swarm)[6].

For more details, we recommend you to refer to the documentation of the Rancher set-up within FIWARE Lab: https://github.com/SmartInfrastructures/rancher-setup-notes

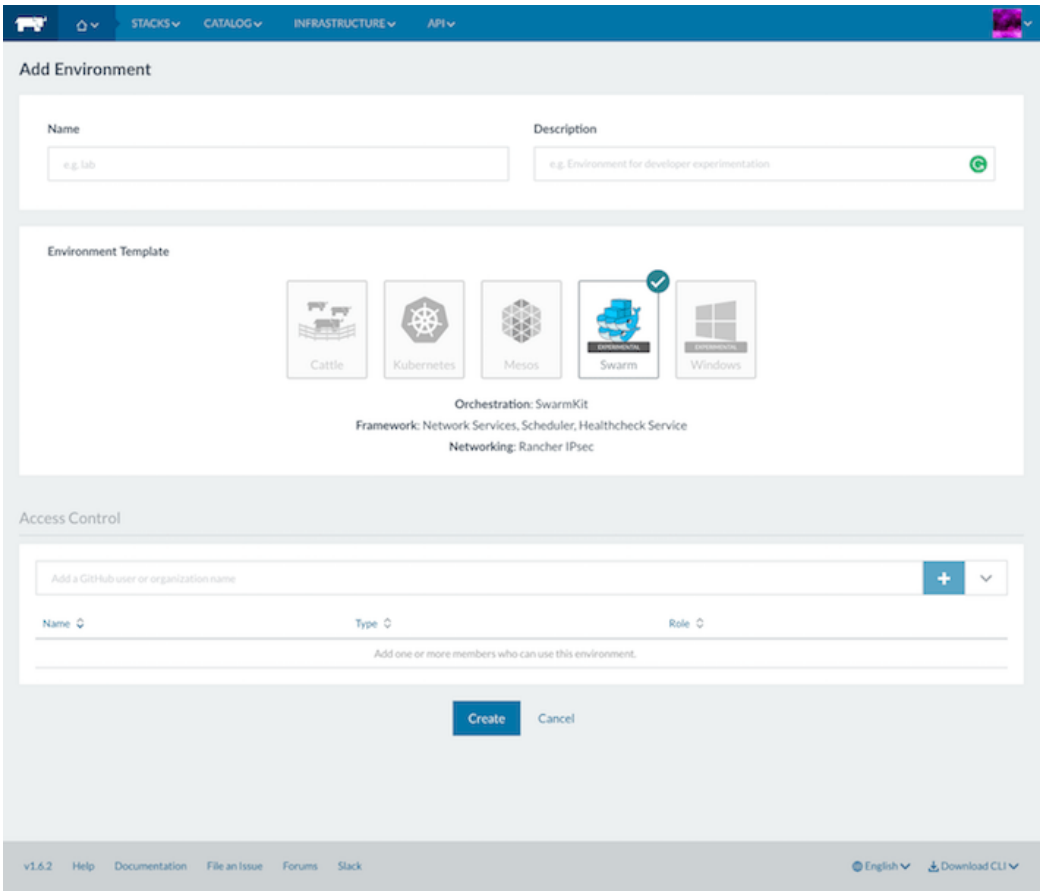


Figure 4.5: FIWARE Rancher tool.

[6] MTU value is an important aspect to enable Swarm nodes to work correctly: https://github.com/SmartInfrastructures/rancher-setup-notes#72-environment-templates

4.5 FIWARE Mastermind

MasterMind is an ongoing development effort to provide a solution to easily configure and orchestrate FIWARE NGSI-based services. The service is provided to FIWARE Lab end-users as a centralised service. No specific action is needed by single FIWARE Lab Node Administrators. Currently the service in under development, so it is not yet deployed in FIWARE Lab. Current status can be found at: https://github.com/martel-innovate/MasterMind-API

The following is a screenshot of the Work in Progress MasterMind UI.

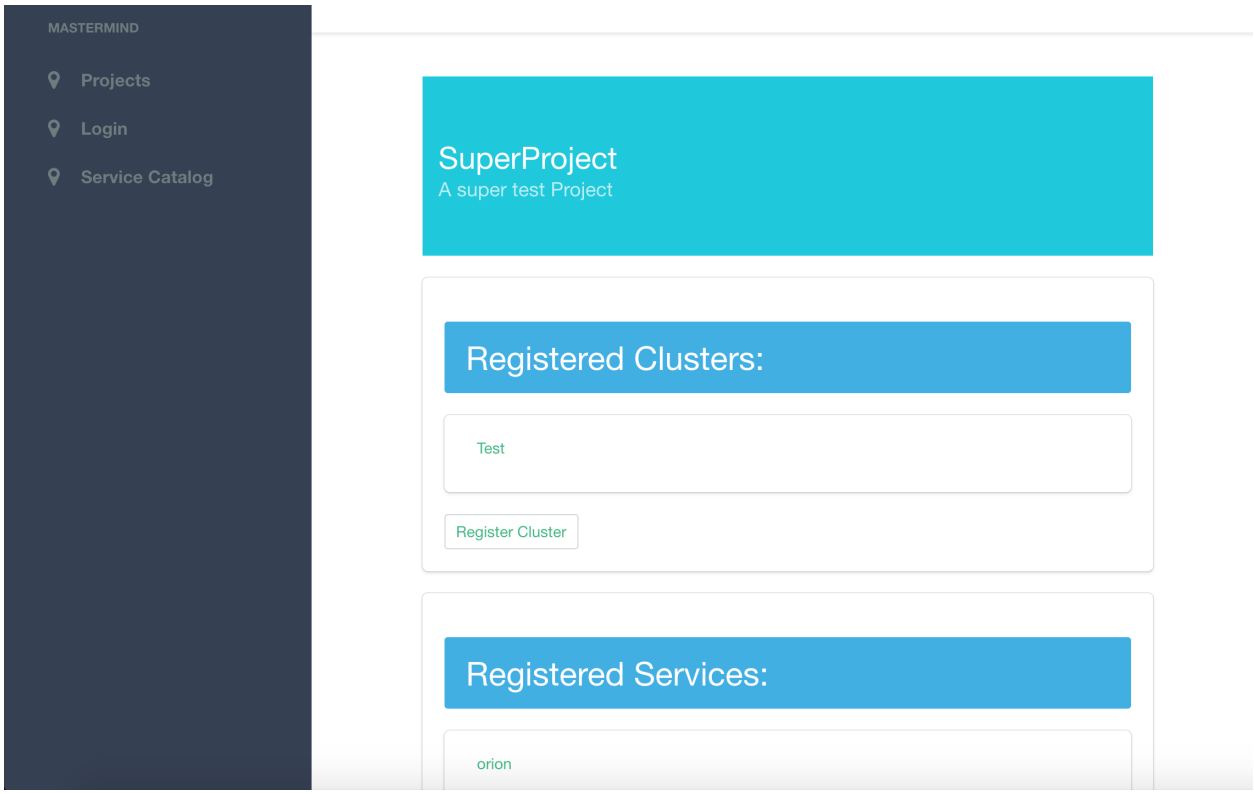


Figure 4.6: FIWARE Mastermind tool.

4.6 JIRA

Jira is the ticket manager tool used from internal and external users.

The main internal usage of Jira is to keep track of all activities carried out within FIWARE Lab and to assign specific tasks to the different stakeholders. Jira, combined to Agile dynamics is used to create monthly Sprint useful to plan upcoming tasks as well as to fix and reach targets.

The following is a picture of Jira which is available at https://jira.fiware.org/

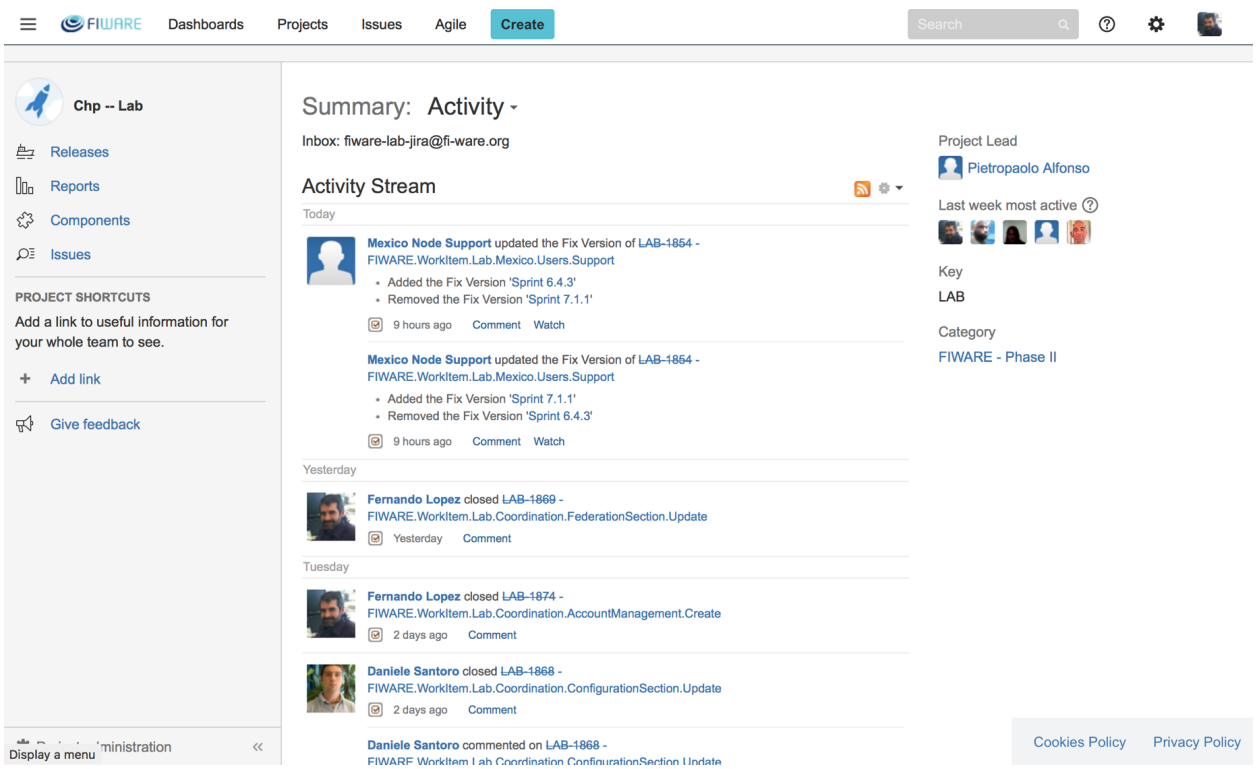


Figure 4.7: FIWARE Jira instance.

4.7 Backlog

The backlog management web site is a tool implemented with the purpose to provide easy understanding of the activity recorded on the trackers. Access to FIWARE Backlog Management Web Site at http://backlog.fiware.org

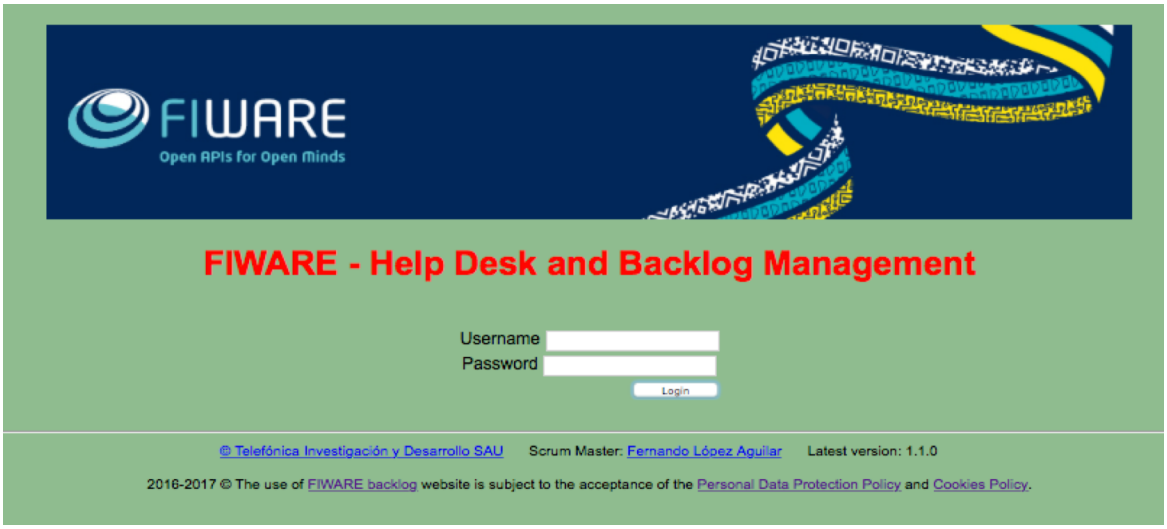


Figure 4.8: FIWARE Backlog tool.

At FIWARE we are looking to be sensitive to several sources demanding both reactive and proactive working modes. Therefore, the Backlog Management web site has been arranged accordingly.

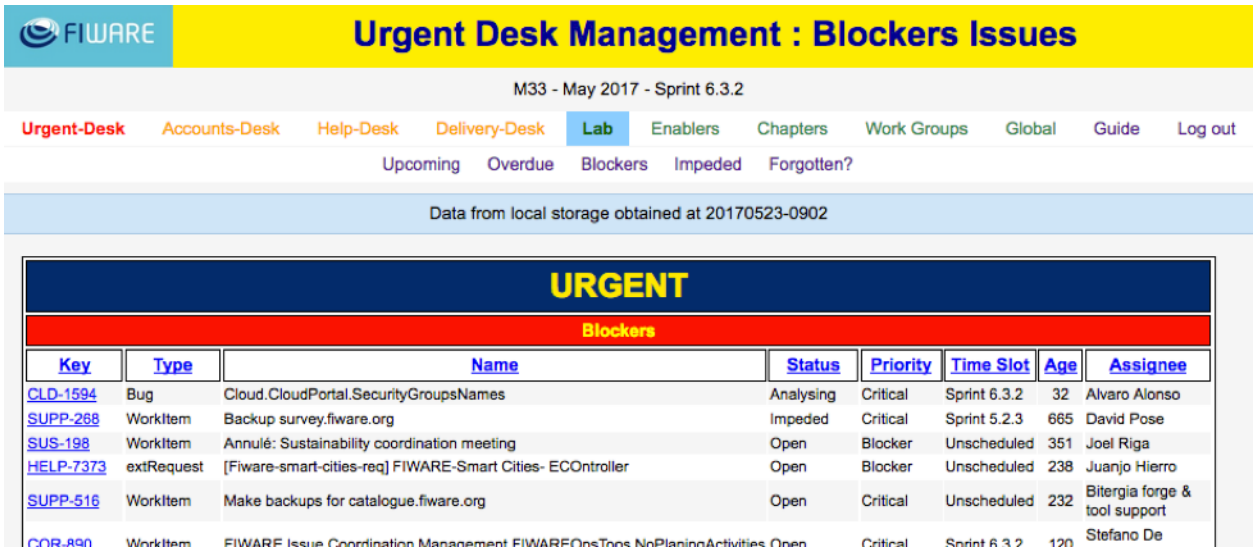


Figure 4.9: FIWARE Backlog, urgent desk management.

Important thing in the Backlog tool is the Urgent Desk and Help Desk option in the menu. The Urgent Desk is an important tool since it allows to create common awareness on issues with time-sensitive fields such as issues' deadlines, priorities and status. To meet this objective, it collects all items in the trackers with relevant deadlines, priorities and status to display them in the desk. Additionally, the Help Desk is also a key tool in supporting the end users. It is implemented by adding email lists connected to JIRA. The Backlog Web Site also helps by providing meaningful views for the different channels and actors.

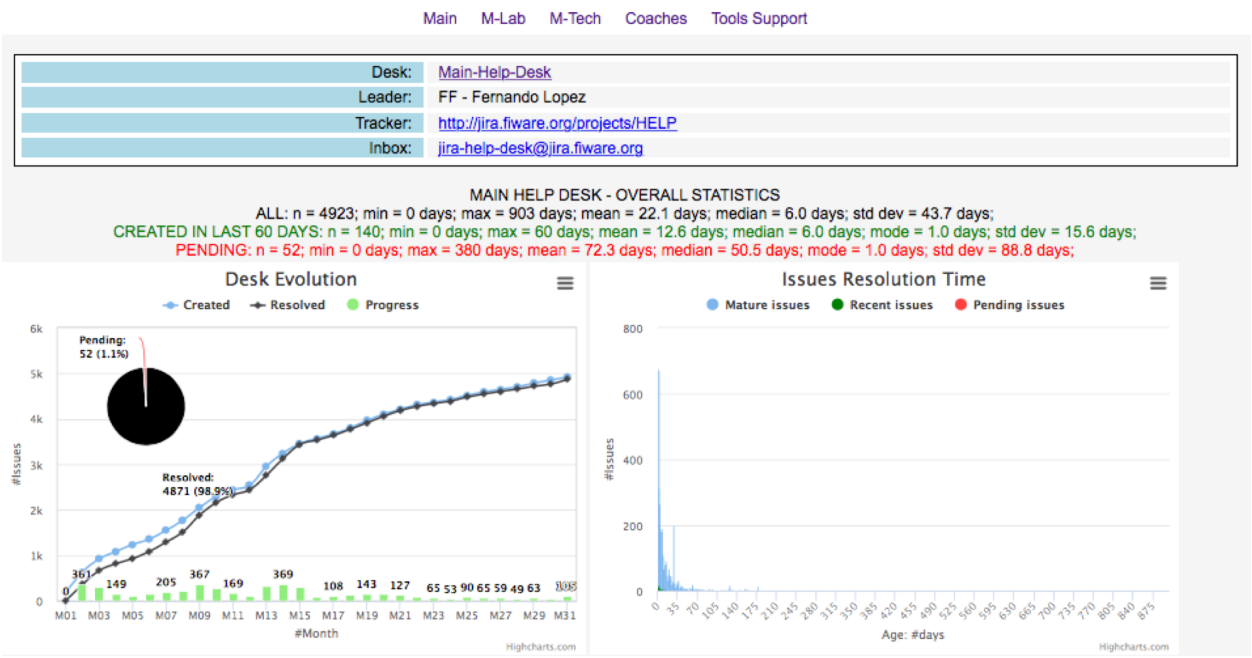


Figure 4.10: FIWARE Backlog, Help-Desk evolution.

Last but not least, each channel has its corresponding view in the tracker and backlog web site. Two key channels from the technical point of view are:

- fiware-lab-help help desk view of the channel devoted to incoming issues to be solved in FIWARE Lab.
- fiware-tech-help help desk view of the channel devoted to incoming issues to be solved in Generic Enabler Implementations.

Both channels are analysed every day by the L1 HELP-Desk support team in order to assign correctly the tickets that we receive to the owners and resolve it in time in order to complete the requirements of the FIWARE Lab node SLA levels.

5 Management operations on a FIWARE Lab node

5.1 Account Management

This section contains the different actions to manage a user account. The most important issue relating to customer accounts is the process to upgrade from a FIWARE Trial Account to a FIWARE Community Account, what are their available resources, how can apply and so on and what have to do every FIWARE Lab node administrator to proceed with it.

5.1.1 Request for Community Account Upgrade

To request an account upgrade, FIWARE Lab users need to fill in an Upgrade Account request.

- Who can apply?

  Participants to the FIWARE Accelerator programme

  Individuals and Companies willing to develop innovative applications based on FIWARE, and to disclose the usage of FIWARE they will make.

- How much does it cost?

  It's free! We are enacting this requirement to ensure that start-ups and SMEs willing to develop applications and demonstrators using FIWARE have the access to the proper resources.

- How long do I retain the Community Status?

  The typical duration for the community status is 9 months. We believe this is enough to allow a team to develop a complete solution based on FIWARE. If needed, accounts can be extended. A procedure will be made available in due time.

- How can I request for a Community Account upgrade?

  Users can apply through the help page (http://help.lab.fi-ware.org) clicking on the "Request Community Account upgrade" button.

- How much should it take to complete the procedure?

  The support team is available 9 am to 5 pm during working days. Except for complex request where negotiation of resources is required, a request should be fulfilled in 1 working day.

5.1.2 Quotas and Flavours

What is a Flavour and what are the available flavours?

Flavours represents the size of virtual machines. FIWARE Lab flavours are based on OpenStack flavours. Available flavours in the FIWARE Lab are listed in this table.

Table 5.1: Available flavours

ID	Flavour	Memory	Disk	Virtual CPUs
1	m1.tiny	512	1	1
2	m1.small	2048	20	1
3	m1.medium	4096	40	2
4	m1.large	8192	80	4

Keep in mind that it is only a recommendation to optimize the physical resources available in a node. Each of the FIWARE Lab administrators is free to create new flavour or change resources assigned to users but in that case, they could break the homogeneity between different FIWARE Lab nodes and could be difficult (if not impossible in some cases) to make a migration from one node to another.

What are the default quotas?

By default, each user has assigned a minimum number of resources that can be used in the development of its solution. This quota is comprised of the following values:

Table 5.2: Default quotas

VM Instances	vCPUs	Hard Disk	Memory	Public IPs	Routers	Networks
2	2	40 Gb	4096 Mb	1	0	0

What are the default maximum quotas?

If the project required more resources, the users have the possibility to explain it and the default quotas can be extended to a maximum default quotas that is comprised of the following values:

Table 5.3: Maximum quotas

VM Instances	vCPUs	Hard Disk	Memory	Public IPs	Routers	Networks
5	10	100 Gb	10240 Mb	1	1	1

Keep in mind that in order that users can obtain more resources, it is mandatory that they provide the corresponding justification why they need more resources. Exceptionally, if users need more resources, after a clear justification the FIWARE Lab nodes can provide extra resources to cover them.

5.1.3 Resource assignment and account upgrade by FIWARE Lab administrators

This section describes how FIWARE Lab Node operators should process incoming requests. The process is simple and depicted below.

Once, for example, a coach assigns you a ticket, you start progressing it, each comment will be notified to the end-user that will be in this way able to provide you further information. For resource assignment/negotiation the guide in Section Quotas and Flavours should be considered.

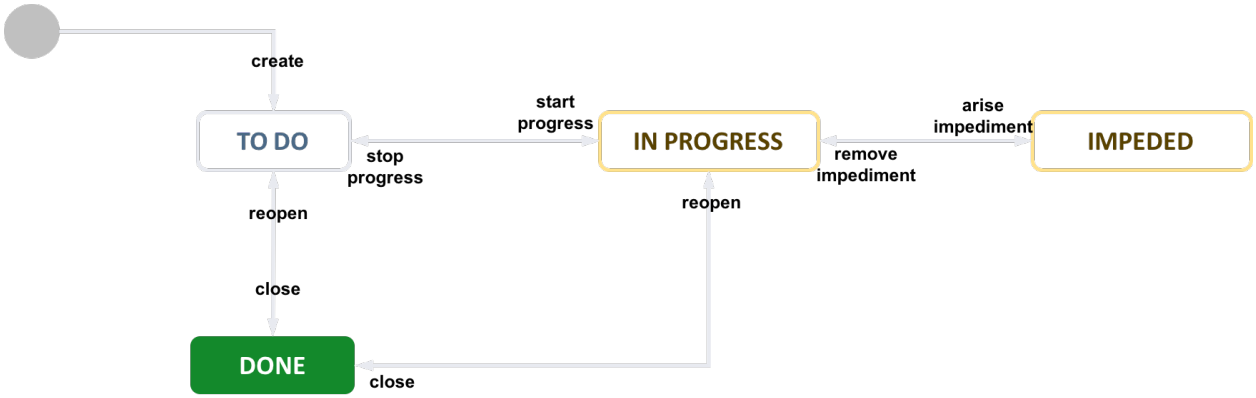


Figure 5.1: Account upgrade workflow.

If your node is not able to complete the assignment, we suggest negotiating the assignment of those resources in other FIWARE Lab node, taking into consideration user's requirements and predefined accelerator's program to FIWARE Lab nodes mapping. Of course, you should inform users about this migration assignment to confirm that it is ok for them.

How to assign resources to a new community user

To manage user information, you have to use the Horizon-based Cloud Portal interface at https://cloud.lab.fiware.org. In this web site, you can to click on the button "Request Community Account" and you will see a dialog box requesting the information required to set up a community account. Once the user has filled all the information and submitted it, the process will start with the creation of the corresponding Jira ticket. The next steps that every FIWARE Lab node has to follow is the following:

1. Start progressing the ticket as soon as you receive it.
2. Check the requested resources and comment to inform the user about decision on resources, or additional information you need.
3. If the user is not created you have to create it.
4. If the user doesn't have an associated project to create cloud resources, you have to create it and to assign the user as a member of the project.
5. Assign the role "Community" in the domain "default" to the user. The user can have several roles for different projects (owner, member, ...), however, it is a Community or a Trial user for the whole domain.
6. Include in the metadata of the user ("description" field) the date in which the user has been upgraded to Community (with the format "community_started_at": "YYYY-MM-DD").
7. Assign the desired quotas in your node to the user
8. Assign the endpoint group of your node to the user's project.
9. If you need to contact the user you can use the email address included in the user detail.
10. Progress to "done" when completed.
11. Alternatively, if any impediment arises, move to impeded, and inform the user by commenting the ticket.

5.1.4 Procedure to create new FIWARE Lab user account

The process to create a new FIWARE Lab user account, includes a procedure to check that the email introduced is a valid email. During this process, FIWARE requested the information to the user to be completed in a Google Form. This information includes:

- User name to be kept in the database.
- User email to communicate any issue directly to the user.
- Preferred FIWARE Lab regions to be used.
- Acceptance of the FIWARE Lab Terms & Conditions.

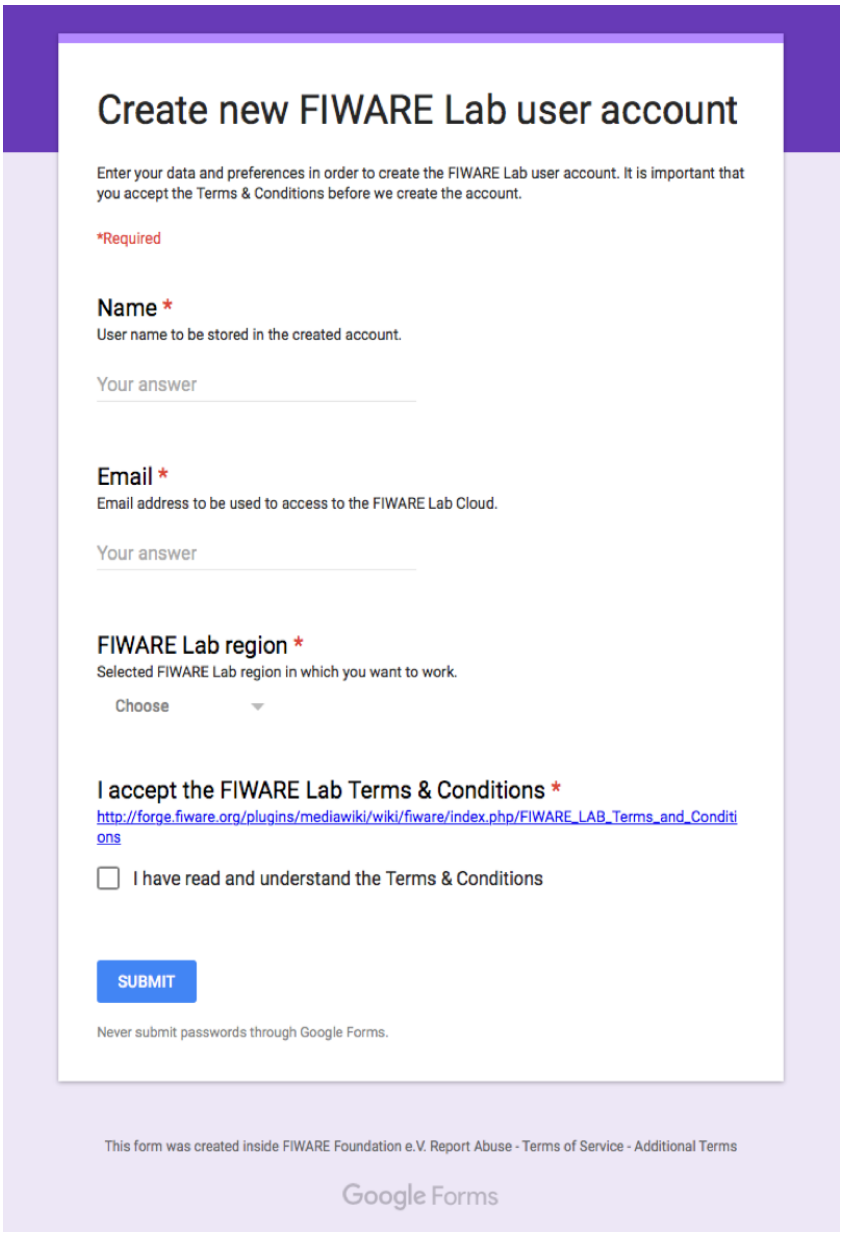


Figure 5.2: FIWARE Lab user creation form.

All these data are mandatory to be fulfilled. This form activates a specific procedure to manage the creation of the FIWARE Lab user account.

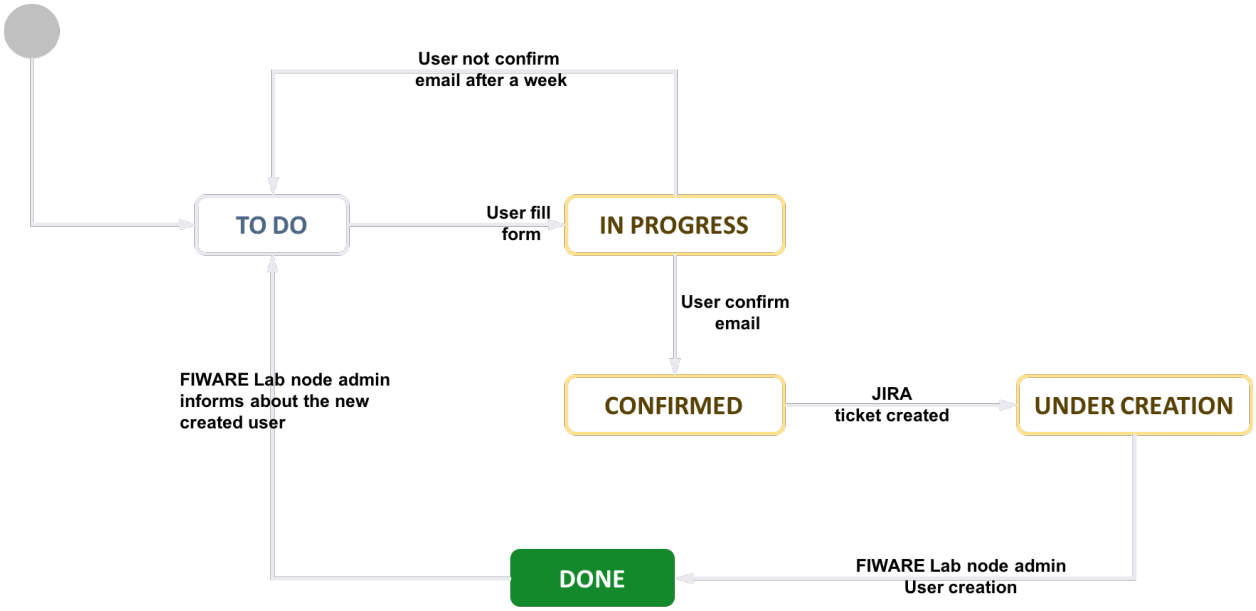
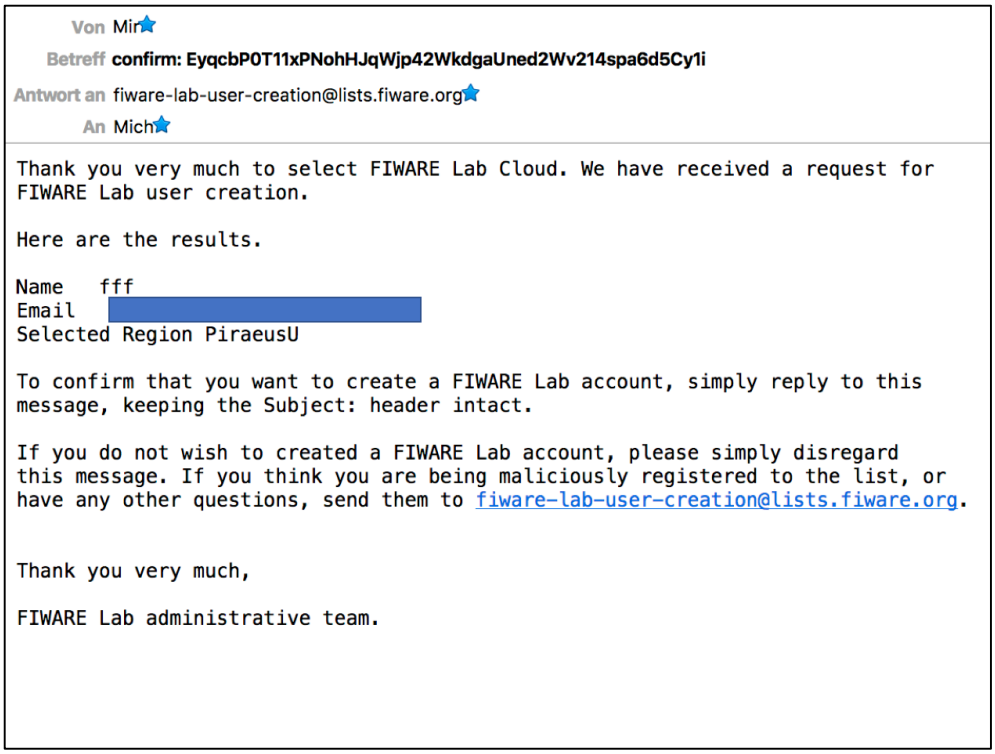


Figure 5.3: FIWARE Lab user creation workflow.

Once that the user submits this information an email will automatically be sent to the user using the email provided. This email includes in the subject (Betreff in German in the following picture) the confirmation code to be submitted. The user must reply to this email in order to confirm that the email is correct.



Von Mir
Betreff confirm: EyqcbP0T11xPNohHJqWjp42WkdgaUned2Wv214spa6d5Cy1i
Antwort an fiware-lab-user-creation@lists.fiware.org
An Mich

Thank you very much to select FIWARE Lab Cloud. We have received a request for FIWARE Lab user creation.

Here are the results.

Name fff
Email [redacted]
Selected Region PiraeusU

To confirm that you want to create a FIWARE Lab account, simply reply to this message, keeping the Subject: header intact.

If you do not wish to created a FIWARE Lab account, please simply disregard this message. If you think you are being maliciously registered to the list, or have any other questions, send them to fiware-lab-user-creation@lists.fiware.org.

Thank you very much,

FIWARE Lab administrative team.

Figure 5.4: FIWARE Lab user creation, confirmation email template.

During this procedure, it is requested to the new user the confirmation to create a new user. This confirmation is obtained through the response of this received email leaving intact the Subject or the email. The received responded email with the Subject intact is enough to confirm and activate the JIRA ticket creation with the corresponding FIWARE Lab node administrator in order to create the corresponding user account. After the creation, the FIWARE Lab node administrator informs the users that the new account has been created and the recommendation to change the password by themselves.

The information of the user created (and eventually deleted) will be kept under Data Protection control for statistical purposes and evaluation of the creation and deletion of user accounts in the FIWARE Lab ecosystem.

5.2 Base-images Management

The creation of base images is a very important operation mainly due the security updates and configuration of them. We download the official images from the different OS supported in FIWARE Lab for OpenStack. There are the three options that we can manage:

- CentOS 6 and 7,
- Ubuntu 14.04 and 16.04 (LTS releases) and
- Debian 7 and 8.

However, we try to modify these images in order to make the default image a little more secure doing some operations on the Base Image. For this purpose, we follow the recommendations of the Centre for Internet Security (CIS). CIS is a forward-thinking, non-profit entity that harnesses the power of a global IT community to safeguard private and public organizations against cyber threats. CIS Benchmarks is the global standard and recognized best practices for securing IT systems and data against the most pervasive attacks. It provides a very exhaustive guideline, continuously refined and verified, to configure Operating System in a secure way. The recommendations that we adopt in the configuration of the virtual machines are the following:

- We remove the default password for the default user. Additionally, the only valid method to login on the Instances is through public-private key.
- Root user is disabled to be used to access to the Instance through SSH.
- We remove the less secure ciphers from the valid ciphers and the less secure Key exchange methods.
- We add a Warning banner explaining that an authorization is needed to access to these Instances.
- We add some IPTables rules to ensure that by default, only some ports (ssh, http and https can be used).
- By default, we enable only automatic security updates.
- The administrative access to the instances is using a specific user with both password and public-private key. Every FIWARE Lab node has assigned the corresponding administrator who contact us to provide details about this access.

All the FIWARE GEis, that are deployed using these base images, inheriting those security configurations options. Sometimes, under the requirements of the FIWARE GEis owners, we need to modify IPTables rules in order to allow the use of other ports.

6 Overview of FIWARE Lab node coordination and support

This section offers an overview of:

- FIWARE Lab coordination and support procedures (including community accounts management, adopted tools, and Help Desk procedures);
- statistics related to overall resources and usage of FIWARE Lab;
- a log of General Events (e.g. failures, maintenance) pertaining to the overall FIWARE Lab.

6.1 Coordination and support procedures

6.1.1 Coordination Approach

FIWARE Lab activities are managed by weekly scrum meetings and monthly Agile sprints. Every Tuesday at 15:00, all Nodes Administrators report their status and discuss different topics relating to the activities carried out within the Lab.

The main topics covered within the meeting are:

- Status of the node

  Each Node Administrator, provides a snapshot of the current status of its node. He/She provides information about the presence within the Infographic page (http://infographic.lab.fiware.org), about the Sanity Checks status (http://status.lab.fiware.org), as well as useful updates during specific activities like hardware maintenance, users migration or OpenStack upgrade version.

- Issues occurred during the previous week

  Each Node Administrator, discusses issues encountered during the past week. This is a crucial point in order to identify and solve possible weaknesses or bugs within the FIWARE Lab architecture (e.g. common connectivity errors toward the centralized keystone) as well as possible weakness in the FIWARE Lab documentation.

- Instructions from the FIWARE Lab Task Force or Technical Steering Committee

  This task is to inform all Nodes Administrators about decisions taken by the FIWARE Lab Task Force or by the Technical Steering Committee and to design a roadmap of the future activities.

- Help Desk pending requests

  All Nodes Administrators are asked to verify all pending (not closed) Help Desk requests, and it is discussed how to resolve them as soon as possible in order to meet agreed SLAs.

- Share suggestions

  This task is to share suggestions among all Node Administrators and FIWARE Lab technical experts. It is an important aspect of the meeting because it allows those who have found a solution or workaround for a specific problem to share that experience among the community to facilitate the expansion and stability of FIWARE Lab.

- Topics of the day

  This is an open window within the meeting to discuss about topics (even off-topics) not covered during a standard meeting.

- Agile Sprint meeting

  Twice a month, in addition to the topics covered above, a Planning Sprint and a Closing Sprint activities is inserted within the meeting. All node administrators are asked to create a "work-item" for every single activity they are going to carry out so that to have a trace of the whole FIWARE Lab activities and effort spent.

Beside all points above, a specific mailing list (fiware-lab-federation-nodes@lists.fiware.org) is constantly used from Node Administrators and FIWARE Lab experts to exchange each other doubts, information, tips and any kind of communication useful to the growth and stability of FIWARE Lab.

6.1.2 Community Account Requests

A Community User is allowed to experiment with FIWARE technology for a period of more than 9 months. Typical examples are SMEs/start-ups under the FIWARE Accelerator Programme.

Trial Users can always apply to for upgrading their accounts to become Community Users. This is granted to everybody if it is understood that the application they aim to developing, is considered a relevant reference example for the development of the FIWARE Community.

In order to apply to become a Community User it is necessary to compile an application form accessible through the main page of the FIWARE Lab portal (https://account.lab.fiware.org) – click the Request Community Account Upgrade - button.

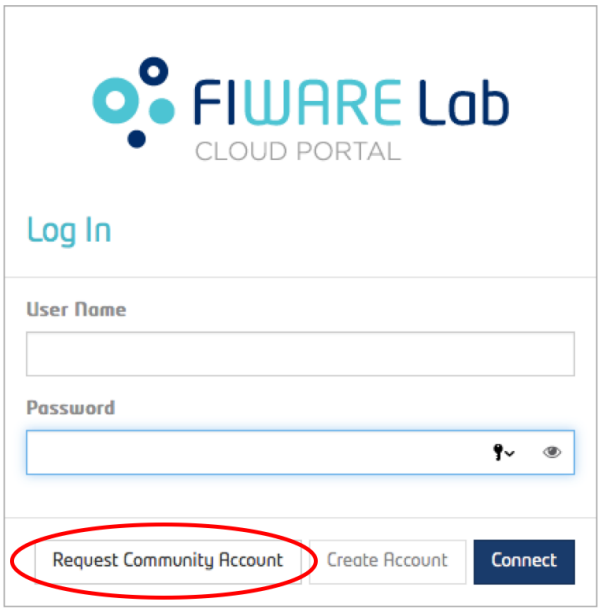


Figure 6.1: FIWARE Lab request community account upgrade.

This will open a window in which information about application is requested from the user, this information will be used to understand exactly what is the planned use of the resources, why they are required and in which FIWARE Lab node you plan to use those resources.

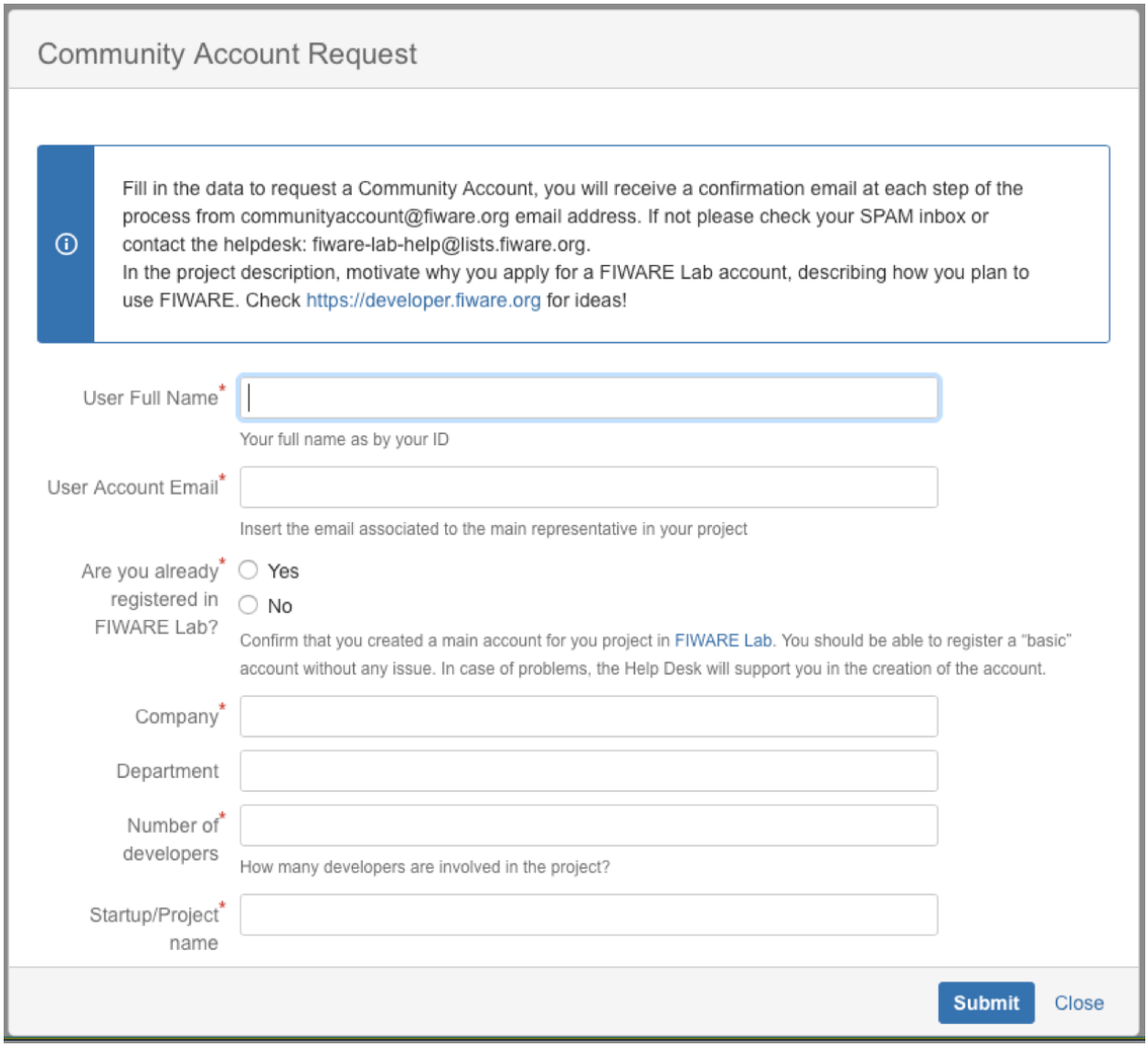


Figure 6.2: FIWARE Lab community account request.

It will create a ticket inside the FIWARE Lab Upgrade Account in Jira to be response by the L1 Support team and assigned accordingly to the corresponding FIWARE Lab administrator node in order to resolve it.

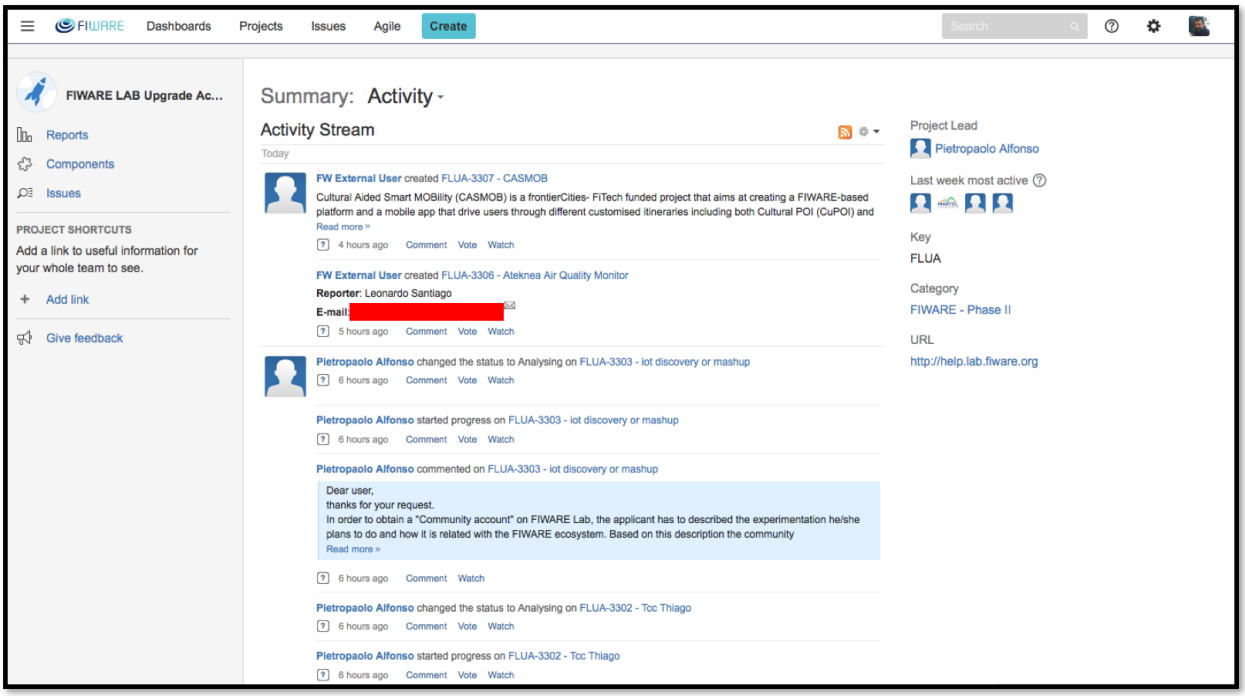


Figure 6.3: FIWARE Lab Upgrade Account (FLUA) Jira project.

From here, the FIWARE Lab node administrators will take those tickets and apply the corresponding activities described in section 7.1.3 to provide the corresponding resources to the user.

6.1.3 Help Desk Support

The Help Desk activities are daily part of FIWARE Lab operations. The Help Desk is the support that FIWARE Lab experts, Nodes Administrators and GEs Owners, give to external and internal users. The Help Desk activities are structured in 2 main Level of support in order to guarantee that agreed SLAs are achieved.

The first level of support is comprised of a team which is in charge of managing all FIWARE Lab incoming tickets. This team is also responsible for categorizing all incoming tickets in order to guarantee that Backlog statistics (http://backlog.fiware.org/helpdesk/main) are always properly updated. If is not possible for a member of the first level support to solve a problem (e.g. a specific Node failure), he/she will assign that ticket to a second level of support consisting of Node Administrators and GEs Owners. The Level 2 Help Desk team is in charge of providing specific high-level support for the corresponding FIWARE Lab Node or FIWARE Generic Enabler.

In the following, we describe in more details the procedures for Level 1 support.

The Level 1 Help Desk team is the initial contact point for all incoming tickets, it provides support for general issues that can be easily solved by pointing to the FAQ, Stack Overflow, groups or other documentation. Moreover, its responsibility is to filter, categorise and forward to Level 2 support (Node Owners or GE Owners) all those tickets that it is not able to answer. It is also responsible for managing Community Account requests.

The Level 1 Help Desk team is organized in 8x5 (Mon to Fri) scheduled shifts from 08:00 to 17:00. The team is composed by people from Engineering, FBK, FIWARE Foundation, Martel that are on duty on a respective week according to the calendar available at: https://docs.google.com/spreadsheets/d/1X9iLY9Znd3Rh-GM4qffGEHRNKqKcHZqdm-dP0rq8OVo/edit#gid=1383004526. Regarding INFOTEC and ITSM they are in charge of the management of the FIWARE Lab Mexico node, providing support to the corresponding Level 1 Help Desk team.

At the end of each day, there should be no tickets remaining in status “unassigned” within the FIWARE Lab Help Desk queue: http://backlog.fiware.org/lab/helpdesk.

Level1 Support Requests Management

The activity that each member of the L1 Support Team should be assigned the corresponding tickets to the proper person, assigning the corresponding Component and for statistical reasons assign the corresponding HD-Chapter, HD-Enabler and HD-Node. They are Jira issues attributes that need to be assigned manually when a new ticket is received. Usually, it is made by the FIWARE Scrum Master together with the process to assign the ticket to the corresponding owner in order to be resolved.

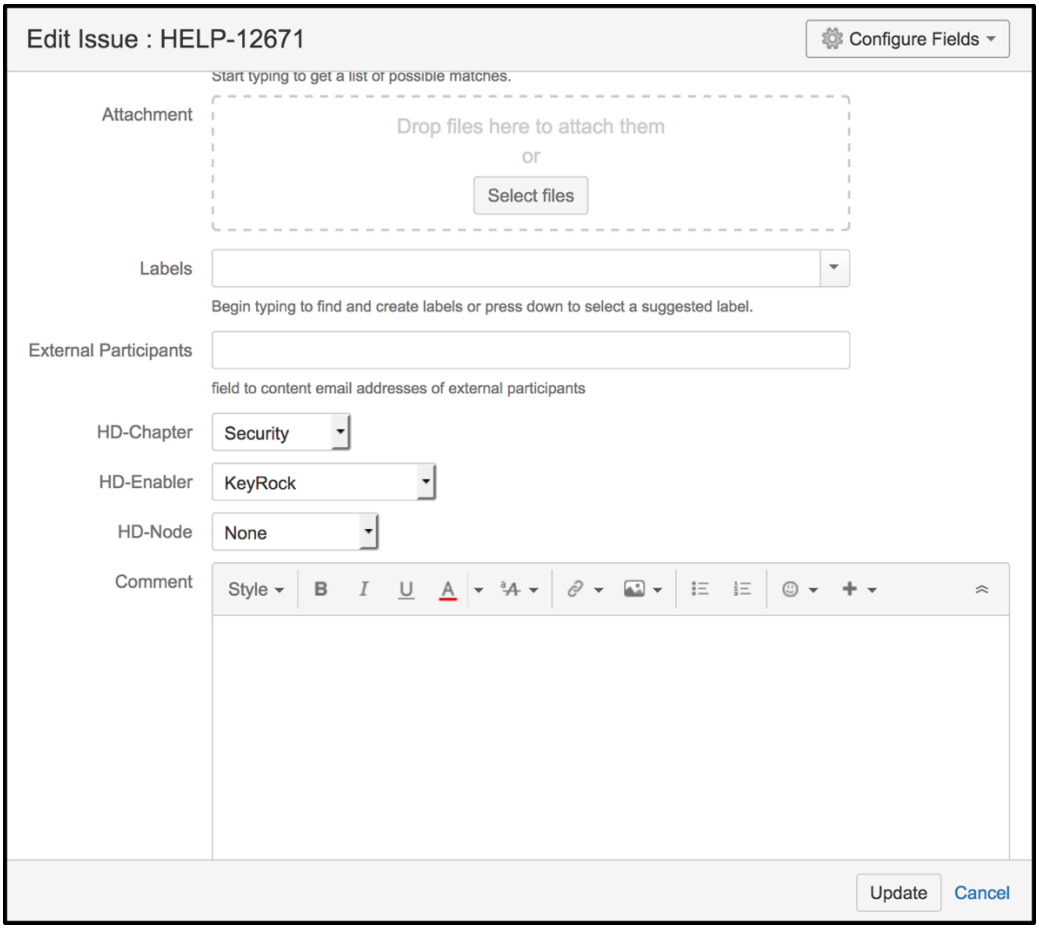


Figure 6.4: HD-Chapter, HD-Enabler and HD-Node attributes in issues.

For the assignee of the JIRA ticket, we have to differentiate between Generic Enablers owners and FIWARE Lab administrators' nodes. For the first one, the list of owners can be obtained from the following table:

https://docs.google.com/spreadsheets/d/1X9iLY9Znd3Rh-GM4qffGEHRNKqKcHZqdm-dP0rq8OVo/edit#gid=694707434

It is a working document of FIWARE and it is available for the FIWARE Lab administrators. If a FIWARE Lab node administrators want to get access to it, they can request access to the owner of the file (fernando dot lopez at fiware dot org). In case of the FIWARE Lab administrator owners this information should be found in the following table:

https://docs.google.com/spreadsheets/d/1X9iLY9Znd3Rh-GM4qffGEHRNKqKcHZqdm-dP0rq8OVo/edit#gid=744561338

Regarding the components, usually they are automatically selected by the different tools that are behind JIRA, but in case that we found a JIRA ticket without the component the available values are the following:

Table 6.1: List of components in Help-Desk Jira project.

Components	Description
FIWARE-COLLABORATION-REQ	Issues related to the request associated to some type of collaboration with FIWARE in terms of participation in some events or in terms of improvement some FIWARE GEri
FIWARE-FEEDBACK	General issues related to feedback recover from the users.
FIWARE-GENERAL-HELP	Issues in general not classified in the other components.
FIWARE-LAB-HELP	Issues related to some of the FIWARE Lab nodes
FIWARE-MUNDUS-REQ	Issues regarding the activities of FIWARE associated to the extension of FIWARE beyond Europe (FI-GLOBAL project).
FIWARE-OPEN-DATA-REQ	(Deprecated) Issues related to the management of Open Data inside CKAN tool or related to the possibility to contribute with Open Data inside FIWARE ecosystem.
FIWARE-OPS-HELP	Issues related to the use of the different FIWARE Ops tools.
FIWARE-SMART-CITIES-REQ	(Deprecated) Issues related to the collaboration in the SmartCities
FIWARE-SPEAKERS-REQ	Issues related to the request to get some FIWARE expert to provide some presentation or speech in some events or just in some Summer School or so on.
FIWARE-TECH-HELP	Issues related to some of the FIWARE GEri.
FIWARE-TRAINING-REQ	(Deprecated) Issues related to the request of training for some of the accelerator program. Currently, it is not needed due to there is no request for training from accelerator programs.

In case of HD-Chapter and HD-Enabler, they make reference to the corresponding Chapter and Enabler that have to be assigned the corresponding JIRA ticket, how it was described previously (see Figure 6.4: HD-Chapter, HD-Enabler and HD-Node attributes in issues.). It is something that have to be selected once we edit the corresponding JIRA issue.

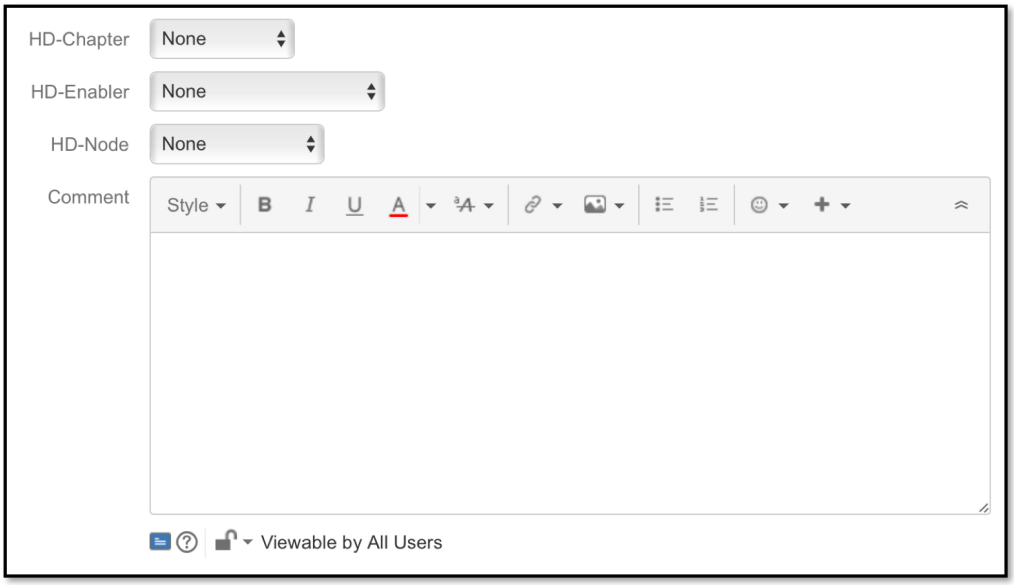


Figure 6.5: Jira Help Desk issue.

Last but not least, HD-Node makes reference to the corresponding FIWARE Lab node in which this issue should be resolved. Usually, the issues are related to FIWARE GEis or FIWARE Lab nodes. The first one is associated to the email fiware-tech-help@lists.fiware.org and the second is associated to fiware-lab-help@lists.fiware.org. This means:

- FIWARE GEis have associated the attribute of the HD-Enabler in order to identity which one is it and the HD-Chapter to know on which FIWARE Chapter is included this enabler.
- FIWARE Lab nodes has associated the attribute HD-Node, in order to know to which FIWARE Lab node is assigned this issue or Jira ticket.
- Due to these tickets come from different source it is needed to complete only HD-Chapter and HD-Enabler in case of tickets associated to FIWARE GEis or HD-Node is case of tickets associated to FIWARE Lab nodes.

How it was mentioned before, the reason of it, is just to allow further analytical analysis of the tickets that we receive in the different channels.

Community Account Requests Management

The Level 1 team is also responsible for the process of approval of the Community Account requests.

Each time a FIWARE Lab user asks for a Community Account, a "FLUA" ticket is generated and the Level 1 team is responsible for verifying the eligibility of users and approve the requests of the user as trusted.

This is done by examining the scope of the account request, e.g. the project for which it was requested and how it adopts FIWARE technologies. To be eligible, a request should: i) make relevant use of key FIWARE technologies (i.e. not just using the Lab as mean to host other technologies); ii) have an experimentation or educational purpose (i.e. commercial services cannot be hosted on FIWARE Lab).

Only if an account is eligible the user is upgraded to Community and the resources assigned accordingly.

Based on the preferences of the user and resource available, the Level 1 team assigns the approved accounts to a Node. The Level 1 team will keep open the initial ticket until the node completes the assignment and eventually follows up with the assigned node (and the user) to ensure that the procedure completes correctly.

6.2 FIWARE Lab Statistics

6.2.1 Available Resources

Physical resources

According to OpenStack documentation the default overcommit for RAM is 1.5:1, the default overcommit for CPU is 16:1 and disk space for the Virtual Machines should not be overcommitted (please, notice that Volumes provisioned by Cinder is not physically where the Virtual machines are) -- That is shown in this page of the documentation:

https://docs.openstack.org/arch-design/design-compute/design-compute-overcommit.html

The available physical resources can be easily calculated when the hardware is bought and installed. However, we could ask nova about the resources (available and used) with a small and simple script like this one:

```
( nova --timeout 15 --insecure list > /dev/null || exit 1
  h=$(($(nova --timeout 60 --insecure list  --all-tenants | wc -l) - 4))
  echo hosts $h
  for a in `nova --timeout 60 --insecure host-list | awk '/compute/ {print $2}'`; do
   nova --timeout 60 --insecure host-describe $a | awk '
       /total/ {printf ("cpu %s\nmem %s\ndisk %s\n",$6,$8,$10)}
       /used_now/ {printf ("used_cpu %s\nused_mem %s\nused_disk %s\n",$6,$8,$10)}
       ' &
done ) | awk '{sum[$1] += $2 } ; END {for (a in sum) {print a,sum[a]}}'
```

The output for this script is something like this:

```
mem 6767440
used_mem 7213056
disk 512859
used_disk 69913
hosts 1864
cpu 1632
used_cpu 3509
```

Where mem, disk and cpu are the physical resources installed in all the compute nodes, and used_mem, used_disk and used_cpu are the virtual resources used. The other field, hosts, is the number of VMs deployed in the node.

As a caveat, the command "nova host-describe" will perform something like a Unix "df" command. So, the returned value for disk is the free space in /var/lib/nova/instances. If a NFS share is mounted in several compute nodes (this would make easier some administrative tasks), the nova host-describe is going to return the free space in the NFS Share multiplied for the number of hosts where it is mounted as well as the used_disk. So, the disk information is wrong.

Floating IPs

The default external network providing floating IPs, how it was described in the previous section is named public-ext-net-01.

```
neutron net-show public-ext-net-01
```

We will see that the subnet associated with that network has ID 4430b64a-85d8-4933-ae79-9a76ff1e2aa9, therefore, querying the subnet:

```
neutron subnet-show 4430b64a-85d8-4933-ae79-9a76ff1e2aa9

…
| allocation_pools  | {"start": "130.206.112.16", "end": "130.206.127.254"} |
| cidr              | 130.206.112.0/20                                      |
```

We have a /20 CIDR which is 2^(32-20)=4096 IPs, however, the 1st one is 130.206.112.16 (16 IPs not in the pool from 130.206.112.0 to 130.206.112.15) and the last IP in the pool is 130.206.127.254 (1 IP not in the pool at the end) --- So There are 17 IPs not in the pool, this means 4096-17=4079 floating IPs. To know how many floating IPs there are in use, we can check it with Neutron as administrator:

```
neutron floatingip-list | grep " 130.206" | wc -l
```

6.2.2 Get summary statistics

In order to get the summary statistics on one node during a period of time, you can execute the command:

```
$ openstack usage list
```

It will return the list of resources consumed for each tenant/project as shown below:

```
Usage from 2013-06-25 to 2013-07-24:
```

Project	Servers	RAM MB-Hours	CPU Hours	Disk GB-Hours
demo	1	344064.44	672.00	0.00
stack	3	671626.76	327.94	6558.86

6.2.3 Hosted Users

In order to get the Users, there are several ways of doing this, the way we usually do is using a script made in Python which retrieves all the information about users stored in Keystone (groups, users, roles, role_assignments, etc.) and produces a big json.

Using the Command Line tool jq, we query this json file:

```
jq -r '.role_assignments[] |
       select (.role.id == "<role_id basic,  trial or community") |
       .user.id' <fichero_json_usuarios> | wc -l
```

It is worth noting that this information is also visible at http://infographic.lab.fiware.org

7 Service Level Agreement

In this section, we describe the Service Level Agreement offered by the FIWARE Lab nodes, what is the procedure to notify node owners about a SLA breach and what should be the procedure to notify that a node is discontinued from the FIWARE Lab. All SLAs have been defined in a measurable way and based on measurable threshold to make the SLA breach detection as objective as possible.

7.1 Definition of the SLA

The nodes that are funded by the FI-NEXT project, have to assure a certain level of quality defined through a Service Level Agreement (SLA). Nevertheless, the downtime of cloud computing relative to SLAs is very difficult to calculate as the root cause of this service interruption involves an evaluation of a complex environment. This is the reason why SLAs focus on the overall service provided by a node, including the response time to resolve issues reported from FIWARE Lab users. Therefore, the FIWARE Lab SLAs defined for each node funded under FI-NEXT are:

- Services availability on the node above 95% threshold
- Level 1 and Level 2 support, Mon to Fri, 9 am to 5 pm CET
- Ticket response time before EOB of the following work day for 95% of requests.
- Ticket resolution time within 2 working days (EOB of the second one) for 95% of requests. Tickets transferred and accepted by level 3 support as tickets that will require a software patch or workaround to be developed by the level 3 support team will be considered resolved at level 1 and 2 support levels.

In the case of nodes financially supported by FI-NEXT, failure to meet the above SLAs during the execution of the project may lead to adjustments in the financial contributions to be provided to the responsible beneficiary to the extent that, in the worst case, the corresponding node may be terminated and related financial contributions can be stopped. This will be definitively the case if for a period of 2 consecutive weeks the node is not operational.

In the case of nodes not financially supported by FI-NEXT, failures to meet the above SLAs, may lead, if not promptly fixed, to the termination of the rights of the node to use the FIWARE brand and its disconnection from the FIWARE Lab federation.

7.2 Procedure to inform failure in SLA

Every month the FIWARE Lab Management team compute the SLAs for the last period. According to the computed SLA the management team will contact nodes that failed their SLA and notify them (see template in Annex G) about the violation and potential implications. The notification includes the violated SLAs and associated values as computed by the FIWARE Lab Management team. The month time frame has been selected to avoid that one single maintenance day at the beginning of the monitoring, influence too negatively the SLA computation.

Once notified, nodes have one week time to check the claims from the FIWARE Lab Management team, and proof that the SLA failure notification is not justified (e.g. the failure was due to central keystone malfunctioning, or the failure was due to network issues of the monitoring services, …).

The SLA is computed through the monitoring services availability and the HELP-DESK tickets handling, as defined below.

7.3 Procedure to monitor SLA

7.3.1 Services availability SLA Monitoring

FIWARE Lab node services are the set of OpenStack services that are running on each of the FIWARE Lab nodes. The measurement of availability of those services is managed through the use of the FIWARE Lab monitoring system tool. This FIWARE Lab monitoring system collects all the monitoring measurements from each FIWARE Lab node. This information is centrally stored, aggregated and available through the FIWARE monitoring API for both real-time raw data and aggregated historical data.

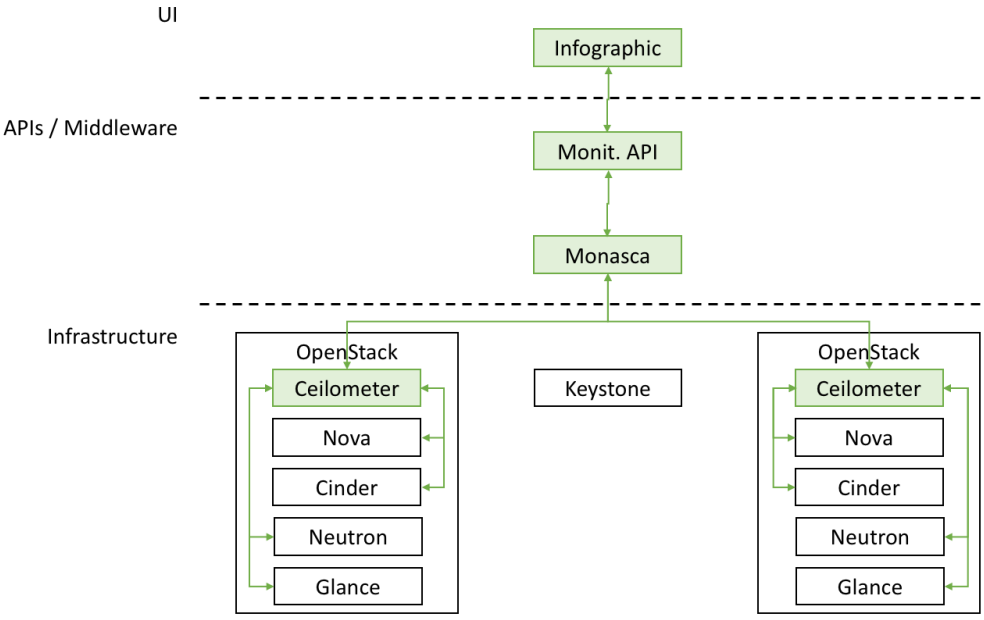


Figure 7.1: FIWARE Lab Monitoring Architecture.

FIWARE relies on the above pipeline to calculate the FIWARE Lab node overall service availability analysing the uptime status of:

1. The OpenStack services (Computing, Networking, Storage and Authentication) using the Monasca agent component installed on each FIWARE Lab node
2. The FIWARE Lab node Sanity Check using FIWARE Lab Health

In particular the monthly averaged uptime of such services is stored in the historical monitoring database as a unique `OverallStatus` metric and its value should not be lower than 95%, in order to respect the given SLA. To retrieve the aforementioned monthly uptime, a specific FIWARE monitoring API (Services4Region) is used:

In order to get aggregated and averaged information, a specific monitoring API (Services4Region) will be used:

http://docs.federationmonitoring.apiary.io/#reference/service/services4region/list-all-services-running-on-a-given-region.

An example of SLA check performed on day 2017-11-01 and considering the services availability of October follows:

1) `curl http://HOST:PORT/monitoring/regions/REGION_ID/services?since=2017-10-01&aggregate=m`
2) A JSON summarizing the daily uptime status of each service will be downloaded
3) The field `OverallStatus` will be considered for the SLA check:

```
OverallStatus:
  {
    value: "yellow",
    value_clean: 0.6789216470588236,
    description: "description"
  }
```

4) Should be this number below 95% over the last SLA monitoring period (a month considered in the example), it will be considered a violation.

Of course, each FIWARE Lab node responsible could check its own availability using the aforementioned API and aggregating historical data by hours, day and month.

7.3.2 HELP-Desk tickets handling SLA Monitoring

As we have mentioned previously, the HELP-Desk tickets are managed through a JIRA instance. This allows us to maintain control and operate a ticketing system to monitor the evolution of tickets with the possibility to fix resolution times and service policies. In summary, JIRA is our single point of contact between FIWARE Lab service providers and the FIWARE Lab users. It is the typical definition of a Service Desk which manages incidents and service requests in the same way that it handles communication with the users. JIRA can offer a plugin called JIRA Service Desk (https://www.atlassian.com/software/jira/service-desk) to provide a fully featured service desk with self-service, automation, SLAs, and Customer Satisfaction (CSAT) reporting. More important for our purposes is that Jira Service Desk provides built-in Service Level Agreement (or SLA) management. It is a time metric, which lets you define how time will be measured for this SLA; and, last but not least, a goal for selected issues, which lets you define a target for the time metric. Different sets of issues can have different goals.

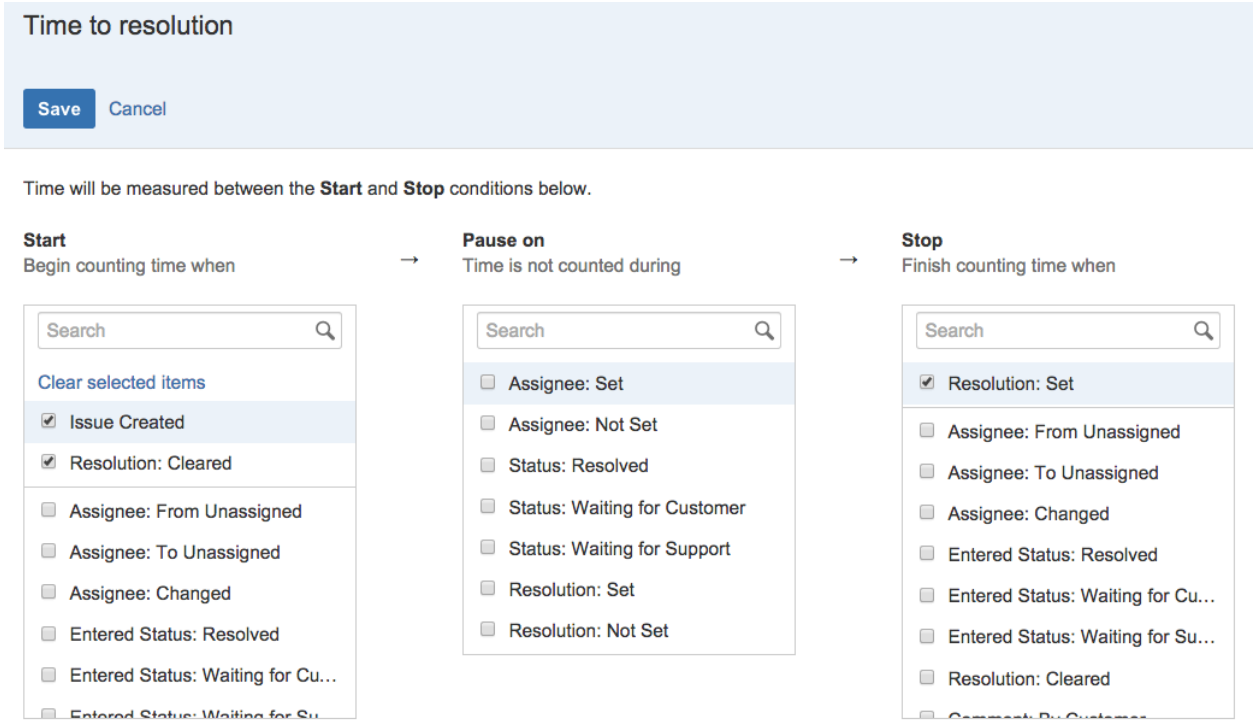


Figure 7.2: Jira time resolution.

While the time conditions on an SLA specify what your team considers to be trackable time, the goal section of the SLA metric lets you set the amount of time that's allowed for different scenarios.

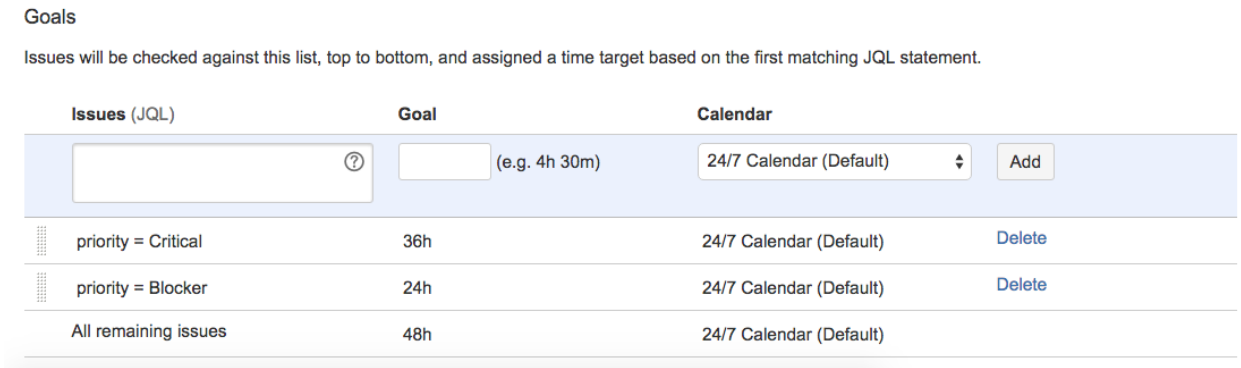


Figure 7.3: Jira goal specification for issues.

We can define SLA Calendars in order to specify team's working hours. For example, SLA calendars let you exclude night-time, holidays, or weekends from the time that affects the SLA

metrics. It is important to take into account that for computing the SLA we exclude the weekend days.

Once that the SLA is configured by the JIRA administrators, team members can see a read-only version of the SLA tab so they can view how the SLA is configured. In the detailed view, the SLA section lists even more detail about the SLA that the issue is being measured against.

Figure 7.4: Jira issue SLA evolution.

The SLA tracker uses colours to indicate the urgency of a given SLA for an issue based on the time remaining. In the following figure, you can see the description of the colours that are offered by the JIRA Service Desk.

Table 7.1: SLA tracker colours, urgency of a given SLA.

10:09	SLA has greater than 1 hour remaining.
0:48	SLA has less than 1 hour remaining. If the SLA goal is one hour, the SLA has 30 minutes remaining.
0:23	SLA has less than 30 minutes remaining. If the SLA goal is one hour, the SLA has 15 minutes remaining.
-17	SLA has breached the target. The amount of time past the goal is shown as a negative number.
0:37	SLA has been paused.

A completed SLA displays the time remaining when the SLA was completed (or the amount of time breached) and an icon to indicate whether the SLA was completed successfully or unsuccessfully.

Table 7.2: SLA completed status.

17 ✔	SLA completed successfully.
-24 ✘	SLA completed unsuccessfully (it breached the target)

As an overall result, we can access this information through JIRA API and in that case, we can, through a script, compute the number of tickets in progress by the end of the day and evaluate if it is below 95% over the SLA monitoring period, which will be considered a violation.

As regards ticket resolution, the operation will be the same. A script computes the resolution time for each ticket assigned to a node over the SLA monitoring period, and compute the number of tickets that are closed within 48 hours (the script will consider properly also weekends or holidays), should be this number below 95% over the last SLA monitoring period, it will be considered a violation.

7.4 Procedure to stop a FIWARE Lab node due to SLA failures

If during a month (as defined above) SLA breach has been detected for a node. The FIWARE Lab Management team will proceed as follows:

1. Inform the node about the SLA breach detection (see Annex G) and ask for clarification about the SLA breach.
2. If clarifications on SLA breaches are not satisfactory and node is not back to normal operations in 30 days (i.e. SLA is violated twice in a row), the node will be notified about the decision of disconnecting it from FIWARE Lab (see Annex G).
3. If clarifications on SLA breaches are satisfactory and node is back to normal operations in 30 days, no other action will be taken.

8 Additional Documentation

FIWARE Mundus released a white paper with a summary of knowledge related to Joining FIWARE Lab. The White paper includes an analysis of CAPEX and OPEX. You can find the last updated version here:

https://www.dropbox.com/s/yz3kic91wgcabsc/JoinFIWARELab_v1.5.pdf?dl=0.

9 Annex A: Template of report of the Testing FIWARE Lab Node locally

Tester: <Name of the person who executes the tests (with email)>
Node: <Name of the node in which these tests are executed>
Node administrator: <Contact person of the administrator of the node (with email)>
Date: <timestamp of the execution of the current tests>

Feature	Scenario	Status	Comments
F1	S1.1	[OK | NOK | --]	
	S1.2	[OK | NOK | --]	
F2	S2.1	[OK | NOK | --]	
F3	S3.1	[OK | NOK | --]	
	S3.2	[OK | NOK | --]	
F4	S4.1	[OK | NOK | --]	
...	...	...	...
F8	S8.1	[OK | NOK | --]	
	S8.2	[OK | NOK | --]	
	S8.3	[OK | NOK | --]	
	S8.4	[OK | NOK | --]	
	S8.5	[OK | NOK | --]	

Legend:

- F means Feature.
- S means Scenario.
- OK means that the tests were executed without any problem
- NOK means that some problem was found during the execution of this test. Additional details could be provided in the comments cells of this error.
- "--" means that the tests were not executed due to a previous scenario with error.

10 Annex B: Template to become a new FIWARE Lab node

STEPS TO BECOME A NEW FIWARE LAB NODE

New FIWARE Lab node: ...
FIWARE Lab node name: ...
Administrative contact: ...
Support email contact: ...
JIRA user: ...

Task	Task Description	Task Owner	Status
0	Sign the letter to become a FIWARE Lab node	Node Admin	[OK | NOK]
1	Provide contacts information.	Node Admin	[OK | NOK]
2	Insert the new node within the agenda of the weekly meeting.	FIWARE Lab Admins	[OK | NOK]
3	Join the weekly meeting.	Node Admin	[OK | NOK]
4	Join the fiware-lab-federation-nodes@lists.fiware.org mail list.	Node Admin + FIWARE Lab Admins	[OK | NOK]
5	Insert the new node in JIRA for help-desk, sprint and FLUA management.	FIWARE Lab Admins	[OK | NOK]
6	Sign in for a JIRA account: https://jira.fiware.org/	Node Admin	[OK | NOK]
7	Sign in for a FIWARE Lab account: https://account.lab.fiware.org	Node Admin	[OK | NOK]
8	Webinar to explain how to use the main tools, such as Jira, sprint, help-tickets, FLUA upgrade.	Node Admin + FIWARE Lab Admins	[OK | NOK]
9	Create sprint activities.	FIWARE Lab Admins	[OK | NOK]
10	Install the local OpenStack node.	Node Admin	[OK | NOK]
11	Federate the node.	Node Admin	[OK | NOK]
12	Update the federation date within the corresponding workitem created for the federation process.	Node Admin	[OK | NOK]
13	Install and configure the monitoring system.	Node Admin + FIWARE Lab Admins	[OK | NOK]
14	Configure the Sanity Check.	Node Admin + FIWARE Lab Admins	[OK | NOK]
15	Insert the new node within Infographic, Health Status, Sanity Check.	Node Admin + FIWARE Lab Admins	[OK | NOK]
16	Synchronize GE images list.	FIWARE Lab Admins	[OK | NOK]

11 Annex C: Template to discontinue a FIWARE Lab node

Steps to discontinue a FIWARE Lab node

Task	Task Description	Task Owner	Status	Date
1	Inform the FIWARE Lab management about your decision at least 1 month before the shutdown	Node Admin	[OK | NOK]	
2	Inform all FIWARE Lab users about the shutdown via the FIWARE Lab notification tool	FIWARE Lab Admins	[OK | NOK]	
3	Take care of the migration of all active users toward a persistent node	Node Admin + FIWARE Lab Admins	[OK | NOK]	
4	Disconnect the node from centralized Keystone	Node Admin + FIWARE Lab Admins	[OK | NOK]	
5	Delete the node from the list of available nodes in JIRA, FLUA, Sanity Check, Infographic, Health Status	FIWARE Lab Admins	[OK | NOK]	
6	Delete from the fiware-lab-federation-nodes@lists.fiware.org mail list	FIWARE Lab Admins	[OK | NOK]	
7	Delete the node from the weekly meeting agenda/minute	FIWARE Lab Admins	[OK | NOK]	
8	Ask the node to send a final report/lessons learnt	Node Admin + FIWARE Lab Admins	[OK | NOK]	

12 Annex D: Letter of intent to become a new FIWARE Lab node

Innovation Action

Letter of intent to become a new FIWARE Lab node

……
(name of the organisation)

1 Letter of intent to become a new FIWARE Lab node

"The FIWARE Foundation, e.V. (hereinafter "FIWARE") representing the FIWARE Community is very happy to receive your intention to be part of the FIWARE Lab ecosystem. FIWARE as such is acting to ensure SLA levels of FIWARE Lab for the benefit of all its worldwide users.

As a new applicant to FIWARE Lab, several activities are required from your side in order to configure the node and put it in the FIWARE Lab ecosystem. These activities are described in the FIWARE Lab Operation Guide, section "Steps to become a New FIWARE Lab node". Please take a look to it.

Although, FIWARE Lab is an environment where experimental activities are carried out, we are requesting you to offer a certain level of SLA in terms of resolution of users' problems while keeping the FIWARE Lab operations up and running with a **SLA of the 95% of operational time**. It means that issues should be solved in an average of **2 working days**.

Accordingly, in the event that the node would not be operational for more than **4 weeks**, the management of the FIWARE Lab might unilaterally disconnect the node without possibility to challenge the decision. The operations of the node will be evaluated according to the automatic tests performed by the FI-Health service of the FIWARE Lab as well as the resolution of issues in of **2 working days**. Please note that each issue is automatically associated with a ticket in our JIRA instance, so proper tracking and monitoring can be performed.

Additionally, we inform you that FIWARE is the owner of the data and therefore responsible for protecting this information. Consequently, FIWARE is the data controller and you can access for notification to Franklinstrasse 13A, 10587 Berlin. We advise you that FIWARE complies with the current German legislation related to Personal Data Protection, Users' Privacy and the Secrecy and Security of Personal Data, as established in the Federal Data Protection Act in the version promulgated on 14 January 2003 (Federal Law Gazette I p. 66, BGBl. I S. 66 in German), as most recently amended by Article 7 of the Act of 30 June 2017 (Federal Law Gazette I p. 2097, BGBl. I S. 2097 in German).

Finally, it is worth to notice that if an organisation plans to discontinue the operations of a node belonging to FIWARE Lab, a set of activities must be set in order that the FIWARE Lab management can offer a soft transition to the FIWARE users active in the discontinued node. These activities are described in the FIWARE Lab Operation Guide in the section "Steps to discontinue a FIWARE Lab node". It is mandatory that to announce the start of the disconnection process **1 Month** before discontinuing the FIWARE Lab node itself.

To be consistent with the above, we need a written commitment about the request of the ..... node. This way, we would then be able to activate the FIWARE Lab support team to provide assistance in the required activities to federate your node."

2 Statement to be signed by the node owner

"To whom it may concern,

as duly representative of ....., I ....., agree that ..... operates a FIWARE Lab node denominated .....

I do understand that we have to fulfil several steps to become a FIWARE Lab node and the different steps to discontinue a FIWARE Lab node besides with the defined SLA of the FIWARE Lab and the tools used to monitor and verify it.

I do understand that this allows to activate the FIWARE Lab support team to provide assistance to my organisation in the required activities to federate the ..... node.

I do understand that Personal Data are owned by FIWARE and, together with it, I will perform all the possible corresponding actions to comply with the German Data Protection Law.

If, for whatever reason, the organisation has to terminate the node operations this will communicated to you and the node users with an advance notice of 30 days."

Date: .....

Signature: .....

13 Annex E: Private Policy updated

FIWARE Privacy Policy

This is the main privacy policy for fiware.org and for certain services provided through these and other FIWARE websites and applications ("our sites") by FIWARE Foundation, e.V. ("FIWARE").

At FIWARE we collect different types of information about our users for three main reasons:

1. To provide personalised services unique to individual users.
2. To help us to monitor and improve the services we offer.
3. If we have permission from the user, to market services, always in scope of the FIWARE Foundation mission, to them.

There may be other privacy policies that apply to certain services we provide. Please read these when you register or subscribe for these services on these sites.

Our principles

1 FIWARE adopts the necessary technical and organisational measures to avoid the loss, misuse, alteration, unauthorised access or theft of the personal data provided, taking into account the state of technology, the nature of the data and the risks to which they are exposed. This means:

we make sure that we have appropriate security measures to protect your information; and

we make sure that when we ask another organisation to provide a service for us, they have appropriate security measures.

2. We will respect your privacy. You should receive marketing emails only from us and, if you agree, from other organisations we have carefully chosen following the current European and German legislation. We will make sure it is clear when you can make these choices. However, we may email you occasionally with information or questions about your registration, your subscription account or postings, for example, with reminders, warnings, business opportunities or copyright requests.

3. We will collect and use individual user details only if we have your permission or we have sensible business reasons for doing so, such as collecting enough information to manage subscriptions.

4. We will be clear in our dealings with you as to what information about you we will collect and how we will use it.

5. We will use personal information only for the purposes for which it was originally collected, how they are defined at the beginning of this document, and we will make sure we delete it securely.

6. Our site is accessible via the internet. This means that people around the world who access our website can see anything you post on the website or twitter comments with the FIWARE hash.

7. If we or our service providers transfer any information out of the European Union (EU), it will only be done with the relevant protection (stated under German law) being in place.

The information that it is collected from you consist on:

when you register or become a member of our portals

when you use the website

through cookies

Certain services that we provide may involve us collecting extra information (Membership, for example or subscription detail to the FIWARE Newsletter), such as where you are, therefore the service can be provided as designed.

Registration

The minimum information we need to register you is your name, email address and a password. We will ask you more questions for different services, including newsletter subscription. Unless we say otherwise, you have to answer all the registration questions.

All responsibility for the completion of forms with false, inaccurate, incomplete or outdated information shall vest on the Users.

To assist us in our marketing, in addition to the data that you provide to us if you register, we may also obtain data from trusted third parties to help us understand what you might be interested in. This ‘profiling’ information is produced from a variety of sources, including publicly available data or from sources such as surveys and polls where you have given your permission for your data to be shared. You can choose not to have such data shared with the FIWARE just informing us about it.

After you have registered, and with your permission, we may send you emails we think may interest you. At any time, you can decide not to receive these emails and will be able to 'unsubscribe'.

Who we share data with

We will not share your personal information with others for marketing purposes unless you have given us your permission. If we have your permission, we will share your information only with other organisations we have chosen carefully.

We can access and release personal information to keep to relevant laws and government requests, to operate our systems properly and to protect both us and our users.

Any other organisations who access your information in the course of providing services on our behalf will be governed by strict contractual restrictions to make sure that they protect your information and keep to data-protection and privacy laws which apply. We may also independently audit these service providers to make sure that they meet our standards. We may use service providers to help us run these sites (or services available on the sites), some of whom may be based outside the EU.

Google Analytics

We use Google Analytics on our sites for anonymous reporting of site usage and for advertising on the site. If you would like to opt-out of Google Analytics monitoring your behaviour on our sites please use this link (https://tools.google.com/dlpage/gaoptout/).

Legal information and how to contact us

FIWARE advises users of the FIWARE Open Source Community services ("the Users") that FIWARE complies with the current German legislation related to Personal Data Protection, Users' Privacy and the Secrecy And Security of Personal Data, as established in the Federal Data Protection Act in the version promulgated on 14 January 2003 (Federal Law Gazette I p. 66, BGBl. I S. 66 in German), as most recently amended by Article 7 of the Act of 30 June 2017 (Federal Law Gazette I p. 2097, BGBl. I S. 2097 in German). This Act serves to implement directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (OJ EC no. L 281, p. 31 ff.). The data controller is the organisation responsible for protecting information and, in our case, is FIWARE Foundation, e.V., Franklinstrasse 13A, 10587 Berlin.

If you would like access to or a copy of the personal information we hold about you, to request a correction, or have any questions about how we may use it or to make a complaint, please contact the Data Protection Manager at the address shown above or just email to fiware-personaldataprotection@lists.fiware.org.

Complaints will be dealt with by the Data Protection Manager, and will be responded to within 30 days at the latest.

If you are not satisfied with the way your complaint was handled, you may be able to refer your complaint to your local data protection regulator.

Changes to the privacy policy

Should we elect to change our privacy policy we will post the changes here. Where the changes are significant, we may also choose to email all our registered users with the new details. Where required by law, will we obtain your consent to make these changes.

Changes to this policy by date

June 2016: Creation of the FIWARE Privacy Policy.

14 Annex F: Cookies Policy updated

Cookies Policy FIWARE

In accordance with the applicable law regarding the use of cookies in relation to the provision of electronic communication services, We, the FIWARE Foundation e.V., hereby inform you about the cookies used on the websites owned by Us (hereinafter, the "Websites") and why they are used. By browsing the Websites, you are giving your consent for them to be used. The cookies used on our Websites include both 'our own' and those of third parties and they enable us to store and access information in relation to the language, the type of browser used and other general characteristics predefined by the user, and also to track and analyse the activity carried out in order to introduce improvements and to provide our services in a more efficient, personalised manner. Websites does not use advertising or behavioural advertising cookies. The use of cookies offers numerous advantages in the provision of information technology services, because, among others: (i) they make it easier for the user to browse the Website and access the different services on offer; (ii) they mean that users do not have to set up the general, predefined characteristics each time they enter the Website; (iii) they enable us to improve the functioning and the services provided through the Website, after the corresponding analysis of the information obtained through the cookies installed. However, users can set up their browsers to accept or reject cookies, or select those allowed and those excluded by following one of the following procedures, depending on the browser used:

- Google Chrome (in the Tools Menu)
  Settings > Show advanced options > Privacy (Content Settings) > Cookies
  More information: https://support.google.com/chrome/answer/95647?hl=en

- Microsoft Internet Explorer (in the Tools Menu)
  Internet > Privacy > Advanced
  More information: http://windows.microsoft.com/en-us/internet-explorer/delete-manage-cookies#ie=ie-9

- Firefox
  Options > Privacy > Cookies
  More information: http://support.mozilla.org/en-US/kb/enable-and-disable-cookies-website-preferences

- Safari, iPad y iPhone
  Preferences > Privacy
  More information: http://www.apple.com/privacy/use-of-cookies/

- Opera
  Settings > Preferences > Advanced > Cookies
  More information: http://help.opera.com/Linux/10.60/en/cookies.html

Analytical cookies:
These make it possible to track and analyse the behaviour of the users of the websites with which they are associated.

<Updated cookies content>

15 Annex G: Templates email for SLAs breaches

Template email for Breaching SLAs (FI-Next partners)

Subject: Node *[Name of the Node]* breached SLAs in the period from *[1st October 2017]* to *[10th October 2017]*

Dear *[Node Representative Name]*,
According to our checks your node breached the following SLAs in the period from *[1st October 2017]* to *[10th October 2017]*:

- SLA1:
- SLA2:
- …

As you are aware, the breach of SLAs may have different impacts on your participation to FIWARE Lab (c.f. Letter of intent) and consequently on your costs claim as FI-Next beneficiary.

Please verify carefully with the data at your disposal the reasons for the occurred breaches. If you are not able to proof that the SLAs violations are not due to your node (e.g. logs showing mal functioning of Keystone), or not able to restore the operations of your node according to the agreed SLAs within 30 days, we will enforce the due actions to remove your node from FIWARE Lab according to the Letter of Intent of your participation to it, signed by *[Node Representative Name]* on *[1st October 2017]*, and the consequent rejection of costs from *[1st October 2017]* claimed for Task 4.3.

Template email for Breaching SLAs (not part of FI-NEXT)

Subject: Node [Name of the Node] breached SLAs in the period from [1st October 2017] to [10th October 2017]

Dear *[Node Representative Name]*,
According to our checks your node breached the following SLAs in the period from *[1st October 2017]* to *[10th October 2017]*:

- SLA1:
- SLA2:
- …

As you are aware, the breach of SLAs may have different impacts on your participation to FIWARE Lab (c.f. Letter of intent).

Please verify carefully with the data at your disposal the reasons for the occurred breaches. If you are not able to proof that the SLAs violations are not due to your node (e.g. logs showing mal functioning of Keystone), or not able to restore the operations of your node according to the agreed SLAs within 30 days, we will enforce the due actions to remove your node from FIWARE Lab according to the Letter of Intent of your participation to it, signed by *[Node Representative Name]* on *[1st October 2017]*.

Template email for decision on node stopping following SLAs breaches

Subject: Decision on Node *[Name of the Node]* participation to FIWARE Lab following SLAs breaches in the period from *[1st October 2017]* to *[10th October 2017]*

Dear *[Node Representative Name]*,
As notified in our email on *[11th October 2017]*, our checks identified the following SLAs breaches in the period from *[1st October 2017]* to *[10th October 2017]*:

- SLA1:
- SLA2:
- …

You reply on *[13th October 2017]* was *[satisfactory/unsatisfactory]*. Accordingly, the FIWARE Lab Management team decided that your node *[can/cannot]* continue to be part of FIWARE Lab.

Decisions have been made taking into account Letter of Intent signed by *[Node Representative Name]* on *[1st October 2017]* that regulates participation to FIWARE Lab. Should you wish to appeal against them, please contact *Stefano De Panfilis* with proper documentation to sustain your claims. [This decision, as consequence, will determinate the rejection of costs from *[1st October 2017]* claimed for Task 4.3 in FI-NEXT.]