Content

• FIWARE Lab Cloud Hosting: Overview and Architecture
• FIWARE Lab: Basic functionalities
  – FIWARE Compute Services
  – FIWARE Network Services
  – FIWARE Storage Services
• FIWARE Lab: PaaS, working with Blueprints
• Reference Information
FIWARE Lab Cloud Hosting: Overview and Architecture
FIWARE Lab Cloud Hosting
FIWARE Lab Cloud Hosting

- Cloud Portal
- Compute
- Object Storage
- PaaS
- Account
- SDC
- Monitoring
Summary

• **Account**: Manages identity and organizations; provides authentication and authorization for other services (OpenStack Keystone)

• **Compute**: Manages the lifecycle of compute instances. Responsibilities include spawning, scheduling and decommissioning of VMs (OpenStack Nova)

• **Network**: Enable Network-Connectivity-as-a-Service for other services, e.g. Compute, (OpenStack Neutron)

• **Storage**:  
  – Persistent block storage for running compute instances (OpenStack Cinder)  
  – Stores and retrieves arbitrary unstructured data object and provide storage for other services, e.g. Image, (OpenStack Swift)

• **Image**: Stores and retrieves VM disk images used by compute (OpenStack Glance)

• **Monitoring**: Monitoring information about VMs

• **SDC**: Deploying Software in VMs

• **PaaS Manager**:  
  – Working with regions  
  – Creating Tiers and deploying Blueprints
FIWARE Lab Cloud – Multiregion
Account

• Creating an account
  – https://account.lab.fi-ware.org

• Understanding organizations
  – Mapped to OS tenants

• Signing in in Cloud Portal
  – https://cloud.lab.fi-ware.org
  – SSO
Add new user

Enter your email and password to access to the FIWARE Lab.

If you do not have it or forgot it, sign up or request for a new one.
FIWARE Lab: Basic functionalities
FIWARE Lab: Basic functionalities

• Create keypair (private key)

• Create security group (incoming ports to VM, e.g. 22 for ssh)

• Deploy an instance
  – choice from a library of predefined images, e.g. Centros, Ubuntu, etc.
  – choice flavor of resource configuration (vCPU, memory, user disk, ephemeral disk).
  – choice security group.
  – choice keypair to ssh into VM.
  – specify configuration scripts (optional).

• Associate public IP with VM.

• Create private networks and associate to VMs.

• Create storage volumes and attach to VM.
You must create a keypair to access to the servers.

Success: Keypair mex deleted.
### Security Groups

#### Security

**Floating IPs**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>default</td>
</tr>
</tbody>
</table>

*Create a Security Group*
Create and edit Security Group rules

<table>
<thead>
<tr>
<th>IP Protocol</th>
<th>From Port</th>
<th>To Port</th>
<th>Source</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>22</td>
<td>22</td>
<td>0.0.0.0/0 (CIDR)</td>
<td>Delete Rule</td>
</tr>
<tr>
<td>TCP</td>
<td>3306</td>
<td>3306</td>
<td>0.0.0.0/0 (CIDR)</td>
<td>Delete Rule</td>
</tr>
</tbody>
</table>

Add Rule

<table>
<thead>
<tr>
<th>IP Protocol</th>
<th>From Port *</th>
<th>To Port *</th>
<th>Source Group</th>
<th>CIDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>required</td>
<td>required</td>
<td>CIDR</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>

* Mandatory fields.
Launch Instances

Launch new instance

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Visibility</th>
<th>Container Format</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoINC</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>CentOS-6.3-sdc</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>CentOS-6.3-x86_64</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>CentOS-6.5-x64</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>CentOS6.3init</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>CentOS65init</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>LPCI-internal</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>MiWi-POI server</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>MiWi-POI server</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
<tr>
<td>Snapshot_orion_citysense</td>
<td>active</td>
<td>public</td>
<td>OVF</td>
<td>Launch</td>
</tr>
</tbody>
</table>

Success: Released Floating IP 130.206.85.21

2014 © FIWARE. The use of FIWARE Lab services is subject to the acceptance of the Terms and Conditions, Personal Data Protection Policy and Cookies Policy
Launch Instances

### Details

**Instance Name** *

Flavor

ml.tiny

**Instance Count** *

1

### Description

Specify the details for launching an instance. The chart below shows the resources used by this project in relation to the project's quotas.

### Flavor Details

<table>
<thead>
<tr>
<th>Resource</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ml.tiny</td>
</tr>
<tr>
<td>VCPUs</td>
<td>1</td>
</tr>
<tr>
<td>Root Disk</td>
<td>0 GB</td>
</tr>
<tr>
<td>Ephemeral Disk</td>
<td>0 GB</td>
</tr>
<tr>
<td>Total Disk</td>
<td>0 GB</td>
</tr>
<tr>
<td>RAM</td>
<td>812 MB</td>
</tr>
</tbody>
</table>

### Project Quotas

<table>
<thead>
<tr>
<th>Resource Count</th>
<th>Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Count</td>
<td>0</td>
</tr>
<tr>
<td>VCPUs (3)</td>
<td>3</td>
</tr>
<tr>
<td>Disk (20 GB)</td>
<td>980 GB</td>
</tr>
<tr>
<td>Memory (4608 MB)</td>
<td>20392 MB</td>
</tr>
</tbody>
</table>

* Mandatory fields.
Launch Instances

Instance Name: HHI
Image: CentOS65init
Flavor: m1.tiny
Instance Count: 1
Keypair: hamburg
Security Group: hamburg

To access the instance:
You need to include a security group with port 22 opened to access via SSH.
You need to assign a floating IP to access from an external network.

* Mandatory fields.
Allocate IP to a project

Security

Allocate new IP

Success: Released Floating IP 130.206.83.21

2014 © FIWARE. The use of FIWARE Lab services is subject to the acceptance of the Terms and Conditions, Personal Data Protection Policy and Cookies Policy
Allocate Floating IP

Security

Associate Floating IP

Associate Floating IP:

130.206.83.21

to instance:

HH1

and to IP Address:

Select IP to associate with

Description

Associate a floating ip with an instance.

Cancel

Associate IP

Success: Successfully allocated floating IP
## Instances Overview

### Instances

#### Overview

<table>
<thead>
<tr>
<th>Info</th>
<th>Specs</th>
<th>IP Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong></td>
<td><strong>RAM:</strong></td>
<td><strong>private:</strong></td>
</tr>
<tr>
<td><strong>125cd18e-fa14-4f5a-8d4e-14a524b5dafe</strong></td>
<td><strong>S12MB</strong></td>
<td>10.0.4.209</td>
</tr>
<tr>
<td><strong>Status:</strong></td>
<td><strong>VCPUs:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ACTIVE</strong></td>
<td><strong>1 CPU</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Disk:</strong></td>
<td><strong>Disk:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>0GB</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Security Groups

- **default**

#### Meta

- **Key name:** hamburg
- **Image Name:** CentOS65init
- **region:** Spain

#### Installed Software

- **Edit**

**Success:** Successfully allocated floating IP
Access to the instance

```
fla@flamac:~$ ssh -i hamburg.pem root@130.206.85.122
Last login: Fri Aug 22 13:06:41 2014 from 10.0.0.1
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[root@hh1 ~]#
```
Multi-tenancy: Do I need to isolate tenants?
- Even if you trust them, you might want isolation.
- Tenant creates his own network(s) and router(s) allowing complex network topologies for multi-tier applications.

- Create private network with subnets.
- Create router and interfaces.
- Set gateway.
- Deploy instance on network.
Create your own network
Add subnet associate to the previous network

### Networks

<table>
<thead>
<tr>
<th>Name</th>
<th>Subnets associated</th>
<th>Shared</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>demo-net</td>
<td></td>
<td>No</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>federation-int-net-01</td>
<td>federation-int-sub-01 10.100.10.0/24</td>
<td>Yes</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>node-int-net-01</td>
<td>node-int-sub-01 10.101.10.0/24</td>
<td>Yes</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>private2</td>
<td>subnet_private2 10.0.2.0/24</td>
<td>No</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

Success: Network demo-net created.
Add subnet associate to the previous network

Add Network Address (CIDR) | Define DNS server
---|---

Network Address: 192.168.194.0/24 | DNS Name Server: 8.8.8.8

Enable DHCP

* Mandatory fields.
Create a router

Assign router name

* Mandatory fields.
Set gateway

Assign a gateway to the router
Set gateway

Select the network
Assign subnet

Double Click on the router name
Assign subnet

Router Detail

Add interface (subnet)

Success: Gateway interface is added.
Assign subnet

**Add Interface**

**Subnet**
(demo-net): 192.168.194.0/24 (demo-subnet)

**Router Name**
demo-router

**Router ID**
6ede42a2-0b3e-4321-8354-5e2d64bec85f

**Description**
You can connect a specified external network to the router. The external network is regarded as a default route of the router and the router acts as a gateway for external connectivity.

* Mandatory fields.

---

Select the network to connect
Deploy a new instance: Networking

Select the network to connect
Create a volume

Create Volume

Volume Name *
volume1

Description
Volumes are block devices that can be attached to instances.

A volume

Size (GB) *
1 8

* Mandatory fields.

Create Volume
Attach a volume to an instance

### Manage Volume Attachments

<table>
<thead>
<tr>
<th>Attachments</th>
<th>Instance</th>
<th>Device</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detach Volumes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Displaying 0 items

### Attach To Instance

**Attach to Instance**

- **Device Name**

- **/dev/vdb**

* Mandatory fields.
### Volumes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size (GB)</th>
<th>Status</th>
<th>Attachments</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume1</td>
<td>A volume</td>
<td>1</td>
<td>in-use</td>
<td>1</td>
</tr>
</tbody>
</table>

Displaying 1 Item
FIWARE Object Store, i.e. OpenStack Swift

- Create container
- Upload objects, i.e. files
- Download objects
- Storlets (to be deployed)
  - Small computer programs that can be deployed and get executed inside a Swift cluster in an isolated manner.
  - Bring the compute to the data thus saving on the bandwidth required to bring the data to the compute.
  - Based on a Swift cluster empowered with the storlet engine Swift middleware and Docker, which allows the execution of user written code inside Swift in an isolated manner.
Object Storage API


- **Authentication to get initial token**
  
  ```
  username='email@company.com' password='mypassword' curl -d '{"auth": {"passwordCredentials":
  {"username":"$username", "password":"$password"}}}' -H 'Content-type: application/json' http://cloud.lab.fi-ware.org:4730/v2.0/tokens
  ```

- **Use initial token to get tenant**
  
  ```
  curl -H 'x-auth-token: $token' http://cloud.lab.fi-ware.org:4730/v2.0/tenants
  ```

- **Authenticate tenant to get token for Object Storage**
  
  ```
  curl -d '{"auth": {"passwordCredentials": {"username":"$username", "password":"$password"},
  "tenantId":"$tenantId"}}' -H 'Content-type: application/json' http://cloud.lab.fi-ware.org:4730/v2.0/tokens
  ```

- **Object Storage URL**
  
  ```
  http://130.206.82.9:8080/v1/AUTH_tenantId
  ```
FIWARE Lab: PaaS, working with Blueprints
FIWARE PaaS

• Deploying components for your application.

• Create blueprint templates.

• Create Tiers on a blueprint template.

• Launch blueprint templates -> create blueprint instances.

• See details of the blueprint instance.

• Check the SW installed on the blueprint instance.
Real scenario

• Users want to define lots of parameters.
  – Password, ports, default installation.

• Users want to install several things in the same server.
  – Tomcat + git, tomcat + java + git, …

• Users need to deploy complex environment.
  – One server for Tomcat, another for MySQL, …

• Some parameters are unknown before instantiate the system.
Deploy example

Blueprint template: fiware1

- **Blueprint Template**: platform specification to be deployed.
- **Tier**: Each kind of software and server to be deployed.
- Each Tier can be deployed in one or several servers (e.g. tomcat, 2-5 servers).
- **Blueprint Instance**: Deployed in the testbed.

Tier 1: Tomcat

Server: 1

Tier 2: Context Broker

Server: 1

Servers: 2-5

Tier 2: Context Broker

Tier 3: MongoDB

Context Broker

mongoDB
Create a new blueprint template

Press the option “Blueprint Templates” from menu

You can open the Catalog of Blueprint Template or create one from the scratch.
Create a new blueprint template

Name *

hh-template

Description

From here you can create a new blueprint.

description

* Mandatory fields.
Add tiers

**Blueprint Templates**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Tiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>hh-template</td>
<td>description</td>
<td>0</td>
</tr>
</tbody>
</table>

To add new Tier, press “hh-template” name.

Success: Blueprint hh-template created.
Add tiers

After press “Add Tier” you see this window to define the servers of this tier.

You should specify the maximum, minimum and current number of servers.

You must select a Keypair to access to those servers.
Add software in tier

Install software pressing the mouse right click.
Add software in tier

<table>
<thead>
<tr>
<th>Software in Tier</th>
<th>Software in Catalogue</th>
</tr>
</thead>
<tbody>
<tr>
<td>git 1.7</td>
<td>apache2 4.0</td>
</tr>
<tr>
<td>Remove</td>
<td>cep 3.2.0</td>
</tr>
<tr>
<td>Edit Attributes</td>
<td>django 1.5.5</td>
</tr>
<tr>
<td></td>
<td>git 1.7</td>
</tr>
<tr>
<td></td>
<td>marketplace 3.2.1</td>
</tr>
<tr>
<td></td>
<td>mediawiki 1.17.0</td>
</tr>
<tr>
<td></td>
<td>mongodbconfig 2.2.3</td>
</tr>
</tbody>
</table>

* Mandatory fields.

Edit special attributes like default port.
Add software in tier
Connect network

Introduce an alias to connect together several tiers.

Drag&Drop the network to connect the tier..
Topology

Network Topology in Blueprint test

demotier1

Internet

demotier2

Internet

Internet

demotier2

Internet

Internet alias1
Press “Action” and select “Launch Template” to launch the Instance.
Launch a Blueprint Template -> Blueprint Instance

You should specify the “Name” and “Description” for your blueprint.
Firstly, the deployment of infrastructure.
Secondly, the installation of the software.
Finally, if all was ok.

Pressing the name you can see the tiers of this blueprint.
Press it to get information of your server.
## Blueprint Instances / HHBlueInstance / hh-tier1

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>IP Address</th>
<th>Size</th>
<th>PaaS Status</th>
<th>Task</th>
<th>Power State</th>
</tr>
</thead>
<tbody>
<tr>
<td>HHBlueInstance-hh-tier1-1-000104</td>
<td>10.0.5.82</td>
<td></td>
<td>INSTALLED</td>
<td></td>
<td>RUNNING</td>
</tr>
<tr>
<td></td>
<td>130.206.85.124</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Success:** Security group hamburg created.
Reference Information
Documentation

• FIWARE Cloud Portal:

• FIWARE Cloud Infrastructure

• FIWARE eLearning Platform
  – http://edu.fi-ware.org/

• More detailed presentation
  – Slides: http://tinyurl.com/fiwarelab-cloud
Now What?!!
If you have any question or problem contact to

**fiware-lab-help@lists.fi-ware.org**

You can go to stackoverflow and ask question with
the tag **fiware** and/or **filab**.