Setting up a new FedCloud site in collaboration with the industry

Fernando Aguilar
aguilarf@ifca.unican.es
Instituto de Física de Cantabria (IFCA)
Santander - Spain
Introduction

- Framework: Internationalization of EBD-CSIC. Updating of information and knowledge systems.
- Doñana National Park: area of marshes, shallow streams, sand dunes and delta.
- Unique biodiversity. Variety of ecosystems and shelters wildlife including thousands of European and African migratory birds and endangered species such as the Spanish imperial eagle and the Iberian lynx.
Introduction

- Contribution to Lifewatch ESFRI. EGI compatible (FedCloud site). Pilot.
- LifeWatch is the European e-Science infrastructure for biodiversity and ecosystem research. ESFRI
- Aims to provide advanced capabilities for research on the complex biodiversity system.
- EGI-LFW CC.
- FEDER + Ministry funds.
- Industry Collaboration - Regional development.
- IaaS - Distributed Control - Shared Environments - Data Preservation
Introduction

Final User

Data Acquisition

Data Portal

Distributed Control

Storage Solution

Collaborative Environments

IaaS

VO Manager

Manage/dev

IaaS

DMP Tool

Invenio

PyVISA

Cloudify

OpenStack

Mesos

ONedata
IaaS + Distributed Control
IaaS Platform

- **Existing Hardware.**
- **Existing procured VMWare licenses for every node.**
- **Requirement to prefer VMWare as hypervisor.**
  - Analysis of limitations.
  - Some issues found:
    - SSH key injection.
    - Snapshot creation is slow.
    - First instantiation of an image is slow.
    - Neutron can’t be used unless you have NSX (extra licenses), and not very mature anyway.

- **3 clusters:**
  - **EBD-1:**
    - 1x Chassis HP Blade C7000
    - 11x Blade HP Proliant BL460G9 (2x CPU E5 2670v3, 192GB RAM, 2HD 1TB)
    - 1x EMC VNX5600 Storage Array (300TB SAS + 350TB SATA)
  - **EBD-2:**
    - 1x Chassis HP Blade C7000
    - 10x Blade HP Proliant BL460G9 (2x CPU E5 2670v3, 192GB RAM, 2HD 1TB)
  - **RBD:**
    - 1x Chassis DELL M1000
    - 18x Blade DELL M620 (2x CPU E5 2680v2, 192GB RAM, 2HD 500GB)
    - 1x EMC VNX5600 Storage Array (300TB SAS + 350TB SATA)
IaaS Platform

- Distribution selected: OpenStack Mirantis 6 (Juno)
- Deployment based on Fuel.
- Creation and loading of base images catalogue
- Integration of cloud-init customization mechanism in base images.
- Load testing (~hundreds of simultaneous images)

---

**Analysis**
- Procured equipment and licenses
- FedCloud Integration
- Deployment Plan

**OpenStack Deployment**
- Node Selection
- Network Topology Design
- VMWare Deployment
- OpenStack Deployment

**FedCloud Integration**
- OCCII
- BDII
- VOMS
- APEL

**Validation Use Cases**
- Open Modeller
- Contenedores

---

![MIRANTIS OPENSTACK Diagram](image)
IaaS Platform

**Analysis**
- Procured equipment and licenses
- FedCloud Integration
- Deployment Plan

**OpenStack Deployment**
- Node Selection
- Network Topology Design
- VMWare Deployment
- OpenStack Deployment

**FedCloud Integration**
- OCCI
- BDII
- VOMS
- APEL

**Validation Use Cases**
- Open Modeller
- Contenedores
IaaS Platform

- Analysis
  - Procured equipment and licenses
  - FedCloud Integration
  - Deployment Plan

OpenStack Deployment
- Node Selection
- Network Topology Design
- VMWare Deployment
- OpenStack Deployment

FedCloud Integration
- OCCI
- BDII
- VOMS
- APEL

Validation Use Cases
- Open Modeller
- Contenedores

- OCCI:
  - Integration with Nova API.
  - Existing driver for OpenStack Icehouse. Adapted to Juno.

- BDII:
  - BDII Integration.
  - Site-BDII Deployment.

- VOMS:
  - Keystone-VOMS Integration.
  - Keystone + Apache + SSL + WSGI Integration.
  - Check correct integration when used by rest of OpenStack Modules
  - Integration of Virtual Organizations (VOs) Lifewatch y dtema.

- AEPL/SSM:
  - Module Integration.
  - Information upload.
Proyecto 1 – Plataforma IaaS

Analysis
- Procured equipment and licenses
- FedCloud Integration
- Deployment Plan

OpenStack Deployment
- Node Selection
- Network Topology Design
- VMWare Deployment
- OpenStack Deployment

FedCloud Integration
- OCCI
- BDII
- VOMS
- APEL

Validation
- Use Cases
  - Open Modeller
  - Contenedores

Sample Images (R, Python)

Private Image Registry

Container Execution Cluster (VMs based on CoreOS)

Cluster Orchestration (Kubernetes)

OpenStack

Linux groups
- Application
- Linux Control Group
- Packages and Libraries
- Docker Container

Docker
- Application
- Packages and Libraries
- Host Operating System
Distributed Control Platform

- Distributed Control Platform:
  - VOs Requirements definition in SLAs.
  - Service packaging and deployment.
  - Service Provisioning, deployment, monitoring and resource accounting.

- Solution Main aspects:
  - TOSCA as the mechanism for service definition.
  - Under the project scope, focus on discrete VM based deployments but consider the possibility of different deployment mechanisms with different managers.

Initial Analysis
- OpenStack Components
- TOSCA Implementations
- Other related projects.

Orchestration
- Heat deployment
- FedCloud Integration
- Cloudify Deployment

Self-Service Portal
- Design.
- Coding, based on Horizon.
- Testing.

Service Packaging
- Validation use case selection.
- Service Packaging.
- Testing
Distributed Control Platform

Initial Analysis
- OpenStack Components
- TOSCA Implementations
- Other related projects.

Self-Service Portal
- Service Catalogue
- Monitoring
- Accounting

Provided Services

OpenStack deployment Manager

FedCloud

Distributed Control Platform

TOSCA

FedCloud Integration

OpenStack Platform (Mirantis)

Virtualization Platform (VMWare)

Hardware
Distributed Control Platform

- Cloudify as the TOSCA implementation to use for discrete-VM deployments.

- Other possible (in the future) implementations:
  - OpenShift PaaS.
  - Cluster Managers (Mesos)
  - Container Orchestators (Kubernetes / Swarm).

---

**Initial Analysis**
- OpenStack Components
- TOSCA Implementations
- Other related projects.

**Orchestration**
- Heat deployment
- FedCloud Integration
- Cloudify Deployment

**Self-Service Portal**
- Design.
- Coding, based on Horizon.
- Testing.

**Service Packaging**
- Validation use case selection.
- Service Packaging.
- Testing.
Distributed Control Platform

Initial Analysis
- OpenStack Components
- TOSCA Implementations
- Other related projects.

Orchestration
- Heat deployment
- FedCloud Integration
- Cloudify Deployment

Self-Service Portal
- Design.
- Coding, based on Horizon.
- Testing.

Service Packaging
- Validation use case selection.
- Service Packaging.
- Testing

Available services

IaaS Platform

VO Manager

Self Service Portal

Service Deployment Manager (e.g. Cloudify)

Deployed Services

VO User

IaaS Manager
Distributed Control Platform

• Basado en Horizon.

Initial Analysis
- OpenStack Components
- TOSCA Implementations
- Other related projects.

Orchestration
- Heat deployment
- FedCloud Integration
- Cloudify Deployment

Self-Service Portal
- Design.
- Coding, based on Horizon.
- Testing.

Service Packaging
- Validation use case selection.
- Service Packaging.
- Testing
Distributed Control Platform

- OpenModeller
- OpenProject.
- GeoServer.
- Possible pilot of OpenShift (PaaS) integration.

Initial Analysis
- OpenStack Components
- TOSCA Implementations
- Other related projects.

Orchestration
- Heat deployment
- FedCloud Integration
- Cloudify Deployment

Self-Service Portal
- Design.
- Coding, based on Horizon.
- Testing.

Service Packaging
- Validation use case selection.
- Service Packaging.
- Testing
Shared Environments
<table>
<thead>
<tr>
<th>Grupo multinacional tecnológico</th>
<th>Sede principal en España (Madrid)</th>
<th>Más de 1.100 empleados</th>
<th>Aeronáutica, Espacio, Defensa, Seguridad, Sanidad, Transporte, Banca y finanzas, y Tecnologías de la Información y la Comunicación</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital privado</td>
<td>Oficinas en 10 países</td>
<td></td>
<td>Origen vinculado al sector espacial</td>
</tr>
<tr>
<td>Fundado en 1984</td>
<td></td>
<td></td>
<td>Ingeniería, desarrollo e integración de sistemas, software, hardware, servicios y productos especializados</td>
</tr>
</tbody>
</table>
PLATAFORMA COMO SERVICIO (PaaS)

What is it?

- Red Hat PaaS Solution (Open Source leader)
- PaaS = Platform as a Service
  - Cloud application platform for automated host, configuration, deployment, administration in a elastic cloud environment.
  - Developer oriented – Scalable and dynamic environments.
  - Advantages:
    - Easier application development and deployment
    - Simpler administration
    - Selection within a catalogue of available environments.
    - Automatic scalable
    - Compatible with different clouds: public, private...
Free service. Web available. RNA-seq analysis

https://trufa.ifca.es/web/

Components:
- TRUFA-web: Front-end web (Python)
- Job-scheduler (Slurm)
- Computing (CUTADAPT, FASTQC...)

HPC architecture:
- Chain of components
- Static configuration

Not cloud oriented → Troubles to deployment over OpenShift
OPENSHIFT ORIGIN (v3)

OpenShift Origin (v3) = Next Generation PaaS

- Application = FrontEnd + BackEnd + Network + Storage
- Microservices and container support (Docker, Kubernetes)
  - More computing density
  - More flexibility
- DevOps support
  - Continuous integration (i.e. GitHub)
    - Automated deployment of uploads
  - Multi environment deployment
    - Dev → Preproduction → Production
- Administration: user management, groups, quotas, monitoring
- Integration: API REST - Workflows
- **PaaS Solution:**
  - OpenShift Origin (v3)
  (despliegue sobre IaaS basado en OpenStack)

  [https://osmaster.ebd.csic.and](https://osmaster.ebd.csic.and)

- **Workflow Python:**
  - Jupyter Hub
  - Kernel Python (kernels R y Matlab compatible)

  [https://jupyter-gmv.paas.ebd.csic.and](https://jupyter-gmv.paas.ebd.csic.and)
Configuration for OpenShift deployment, emulating HPC architecture:

- Configuration 1 (2 nodes):
  - Container A: TRUFA-web + Job-Scheduler
  - Container B: cComputing layer

- Configuration 2 (11 nodes):
  - Container A: TRUFA-web + Job-Scheduler
  - Container B (x 10): Computing layer
Proof of concept of TRUFA deployment over Apache Mesos

Apache Mesos allows to emulate a grid architecture based on containers (Docker) over cloud

Proposed architecture:
- TRUFA-web → Container deployment
- Job-Scheduler → Mesos (Marathon, Aurora)
  - TRUFA-web modification to change slurm by mesos and components
- Computation layer → Deployment over containers
Data Acquisition
ADEVICE is a Spanish SME focused on Internet of Things.  
We design devices
We create solutions

This is achieved by offering an interface for data which abstracts the complexity of heterogeneous sensors and instruments and reducing the maintenance costs with unattended devices.

www.adevice.es
@adevice
Propietary LabVIEW software installed on Remote Computer
- Data acquisition configured via TeamViewer
- Data exposed by LabVIEW web service
System Architecture

After

- Open Source pyVISA software and Remote Instruments Application installed on Remote Computer
- Data acquisition configured via web service interface
- Data sent to final storage
- Remote Instruments Application usable via IPython Notebook friendly front-end interface.
Remote Instruments Application Architecture
Components

- **Web Service interface**: REST interface for setting up instruments and periodic tasks.
- **Configuration storage**: Saves instruments connection parameters and periodic tasks configurations.
- **pyVISA and Backends**: Libraries for connecting to instruments.
- **Readings Temporary Storage**: Store Readings before sending them.
- **Periodic Tasks**:
  - **Data sending**: Sends readings to a data receiver web service.
  - **Readings collecting**: Collect readings from the instruments
Integration
with portal and final storage

Usable via IPython Jupyter Notebook friendly front-end
• Built over REST API client
• (work in progress)

Readings sent to storage via simple REST API
• Data sent: country id, application id, instrument id, parameter id, data owner, reading data, reading time.
**Integration**

*with portal and final storage*

Usable via IPython Jupyter Notebook friendly front-end
- Built over REST API client
- *(work in progress)*

Readings sent to storage via simple REST API
- Data sent: country id, application id, instrument id, parameter id, data owner, reading data, reading time.
DMP + Storage + Data Portal
Viavansi

Founded in 2000 in Seville, Spain, is a SME specialized in:

- Open Data / Open Government,
- Mobile Applications
- Electronic Signature solutions.

Since the beginning, it has been characterized by the development of innovative solutions, its highly qualified team, as well as its willing to improve the citizen experience. It has a consolidated experience within Private and Public Institutions and it has developed multiple projects in Spain and Latin American countries.
Data Planning
DMPTool LifeWatch
DMPTool LifeWatch is a tool that allows Data Management Planning. We have added **new functionality**:

- DOI's allocation on a DMP
- Add taxonomies to DMPTool
- Ability to associate taxonomies to sections of DMP
- Adapting the generation of PDF / RTF for viewing new features
- Adaptation of Rest API to add new features
- Generating RDF/XML associated with DMP
- Integration with the Data storage and the LifeWatch Open Science Framework
Create and consult DMP

Create DMP and consult DMP: Public, from your institution or user
Cuerda del Pozo (Life ROEM+)

1. Types of data produced

In this project, we have data from mainly three sources: platform, external data and models output.

Platform

There is a platform placed in the middle of the reservoir with an infrastructure that can get and store data related to the water and its environment. The platform cabin supports in its structure a weather station, a net radiometer, and a GPS receiver, as well as an external directional antenna for communication with the PC server in the office in the shore. Also there is installed an altimeter and a probe measuring the depth of the water reservoir and the depth. It also supports two renewable energy systems: an aerogenerator and a set of solar panels. Furthermore, there is a profiler infrastructure that takes data from the bottom to the top of the water column. All this information is stored and sent to the PC server. The system is currently taking data and stores it since 2010.

The following list gives information about the parameters taken:

• The weather station installs is a Vaisala WXT-520. It measures the barometric pressure, humidity, precipitation, temperature, and wind speed and direction.
Using taxonomies and ontologies

You can load a taxonomy or ontology whose concepts will be used later in a DMP.
Using taxonomies and ontologies

Then you can add a new section to a DMP template associated with the taxonomy.
Using taxonomies and ontologies

For each DMP in addition to the usual PDF and RTF that DMPTool can generate, you can now generate a file in RDF/XML format for semantic web.
Using taxonomies and ontologies

In the RDF / XML generated will appear the DMP metadata, URIs of ontology concepts used and the access URL to the associated PDF
Data storage and retrieval

Onedata
Onedata

Onedata is an open source storage solution for integrating access to data from various providers in a High Availability scenario.

Onedata hides system complexity providing a global filesystem-like transparent view of data accessible from everywhere, laptop, server, cluster, cloud or grid.

Onedata allows authentication using well known authentication providers Google, FB, Twitter and X.509 and token based authentication.
Onedata: Use in the project

Onedata is being used as main target for storing user data info, letting user access their files through onedata.org site and LifeWatch portal.

Onedata stores data from sensor devices, user experiment sources and results, keeping them protected by user role and device access rules.
Issues found

Issues found are divided in two main areas:

- Deployment scenario is limited to very specific Server environments, mainly Scientific Linux servers, which may require specific knowledge for most IT administration teams.

- As a product in recent production status, does not have much documentation or community support, so most issues must be solved by their support team (which is always readily available).
Execution environments

Jupyter Notebook (IPhyton)
The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.

Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.
LifeWatch Open Science Framework

Powered by Invenio
LifeWatch Open Science Framework

The LifeWatch Open Science Framework is a tool for researchers and general public that brings together data management features with a platform for analyzing those data.

- Share and discover Data Management Plans (DMPs), Datasets and Software.
- Combine Datasets and Software to create Analysis that can be executed on cloud resources.
- Preserve and make citable all records via DOIs.
Data Life-cycle

- Collaborative management of the entire workflow of data from plan to final publication.
- Fine grained control of records: decide when to publish, access rights, assignment of DOIs...
Preservation possibilities

• A local persistent identifier (PID) is associated to each record created in the framework.

• Digital Object Identifier (DOI) minted to make records uniquely citable widely; records owners decide when to mint the DOI.

• Records (and metadata) entered into the framework are stored safely in reliable disk and backed up to tape for preservation.

• Not limited to datasets: DMPs, Software and Analisys as first-class citizens for preservation.
### Sample dataset record

**Processed Physicochemical Water Parameters 2013**

Ecohydros S.L.

<table>
<thead>
<tr>
<th>Publication Date</th>
<th>2015-11-05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent Identifiers</td>
<td>PID lifewatch.openscience.3 DOI 10.5281/1wdaap.3</td>
</tr>
<tr>
<td>Access Rights</td>
<td>open</td>
</tr>
<tr>
<td>Description</td>
<td>Processed physicochemical water data taken at Cuerda del Pozo Reservoir during 2013. Files of the dataset include depth and temperature for the thermocline analysis</td>
</tr>
<tr>
<td>Keywords</td>
<td>Cuerda del Pozo Duero Soria confederacion hidrografica depth fresh water reservoir temperature</td>
</tr>
</tbody>
</table>
Execution environments for analysis

- Execution environment available using the cloud infrastructure.
- User selected OS, flavor and environment.
Exporting the catalog in RDF/DCAT

Data catalog of the Portal can be exported in RDF/XML format based on the vocabulary DCAT, developed by the World Wide Web Consortium (W3C) and allowing the standardization in the definition of catalogues of documents and information resources.

A catalogue of documents and information resources is represented by instances of the dcat:Catalog class and includes a collection of sets of information dcat:Dataset resources.

This RDF/XML file may allow the federation in top-level catalogs and could be used for machines in semantic web environments.
Conclusions

- Collaboration with industry impacts directly to the regional development: experience, knowledge.
- Existing tools integration: added value. Continuously adapting.
- Two approaches: VO managers (and developers) + Final users.
- Cloud environment - Data oriented
- Pilot for Lifewatch infrastructure.
Thanks for your attention