Digital archive as a service: automatic deployment of an Invenio-based repository using TOSCA orchestration and Apache Mesos

Speaker: Marica Antonacci - INFN

Alberto Brigandi (Concept Reply), Miguel Caballer (UPV), Eva Cetinić (IRB), Davor Davidovic (IRB), Giacinto Donvito (INFN), Germán Moltó (UPV), Davide Salomoni (INFN)
Motivation

• **What**: provide a **service** that simplifies the process of **creating and managing repositories** of various digital assets using **cloud** resources

• **Why**: help individual researchers or small-to-moderate-sized research groups to **address challenges** like:
  • Resource management and availability
  • Installation/configuration process
  • Service operation and maintenance
  • Scalability

• **How**: **INDIGO-DataCloud** project provides open-source tools and solutions for building services on heterogeneous and hybrid cloud environments
Digital archive as a Service

1. Submit Template

2. Select Provider & Deploy

3. Monitor/Adjust Config

4. Access Repo

INDIGO Orchestration

1. User

INDIGO Paas Micro-Services

User Community

Cloud Providers:
- OpenNebula
- AWS
- OpenStack
The **PaaS Orchestrator** allows the **transparent access** to heterogeneous **cloud environments** (Openstack, OpenNebula, AWS, Azure, etc.) and the selection of the **best** resource **providers** based on criteria like user's SLAs, services availability and data location.
The deployment model

From docker-compose…

..to distributed deployment on Mesos

Marathon-LB
Marathon Framework
Mesos Master
Mesos Slave
Mesos Slave
Mesos Slave

Persistent storage
Automated deployment

1. **Automatic deployment** of a complete HA elastic Mesos cluster
   
   • The virtual machines are contextualized using **ansible** playbooks:
     
     • Each server receives the proper configuration according to its role in the virtual cluster

2. **Automatic deployment** of the Invenio services as **docker containers** running on top of the Mesos cluster
   
   • The clustered services are managed through a **Marathon group** in order to preserve the dependencies among the containerized services
Virtual cluster features

• **Marathon** manages **long-running services**
  • it takes care of keeping containers up and running
  • provides health checking for detecting when the services are not alive
• The deployed services are accessible through the **cluster edge load-balancer**
  • HAProxy configuration is dynamically updated to route ingress traffic
  • SSL support to secure the deployed services
• **Persistent storage** is provided to the containers running stateful services
  • Using block devices provisioned in the cloud
• Resources can be **automatically scaled** depending on the real workload
The TOSCA Template

• **Standard** description of the **topology** of the applications in cloud

• The **digital repository** instance can be easily **customized** using the input parameters: you can change

  • the Docker images for the different services

  • The resources (mem, cores, storage size) needed to run each service

```yaml
cache_cpus:
  type: float
  description: Number of CPUs for cache (redis) container
  required: no
  default: 1.0

cache_mem:
  type: integer
  description: RAM in MB for cache container
  required: no
  default: 1024

cache_image:
  type: string
  description: docker image for cache container
  required: no
  default: 'redis'

db_cpus:
  type: float
  description: Number of CPUs for DB (postgres) container
  required: no
  default: 1.0

db_mem:
  type: integer
  description: RAM in MB for DB container
  required: no
  default: 1024

db_image:
  type: string
```
Conclusions

In the framework of the INDIGO-DataCloud project we have implemented a solution for deploying on-demand Invenio-based data repositories exploiting cloud resources.

A demonstrator was developed for supporting the Arts and Humanities Research use-case and is now being extended for the DARIAH Thematic Service in the EOSC-HUB project.

The implemented solution is based on INDIGO tools and allows a user to:

• provision and scale automatically cloud resources through the INDIGO PaaS services;

• deploy automatically the repository components as docker containers on top of a Mesos/Marathon cluster dynamically installed and configured;

• monitor and manage the services through the user-friendly Marathon GUI.