Abstracting application deployment on Cloud infrastructures

D C Aftimiei\(^1\), E Fattibene\(^1\), R Gargana\(^2\), M Panella\(^1\) and D Salomoni\(^1\)
\(^1\) INFN-CNAF (Centro Nazionale Tecnologie Informatiche)
\(^2\) INFN-LNF (Laboratori Nazionali di Frascati)

E-mail: enrico.fattibene@cnaf.infn.it, matteo.panella@cnaf.infn.it

Abstract. Deploying a complex application on a Cloud-based infrastructure can be a challenging task. In this contribution we present an approach for Cloud-based deployment of applications and its present or future implementation in the framework of several projects, such as “!CHAOS: a cloud of controls” \(^1\), a project funded by MIUR (Italian Ministry of Research and Education) to create a Cloud-based deployment of a control system and data acquisition framework, "INDIGO-DataCloud" \(^2\), an EC H2020 project targeting among other things high-level deployment of applications on hybrid Clouds, and "Open City Platform"\(^3\), an Italian project aiming to provide open Cloud solutions for Italian Public Administrations. We considered to use an orchestration service to hide the complex deployment of the application components, and to build an abstraction layer on top of the orchestration one. Through Heat \(^4\) orchestration service, we prototyped a dynamic, on-demand, scalable platform of software components, based on OpenStack infrastructures. On top of the orchestration service we developed a prototype of a web interface exploiting the Heat APIs. The user can start an instance of the application without having knowledge about the underlying Cloud infrastructure and services. Moreover, the platform instance can be customized by choosing parameters related to the application such as the size of a File System or the number of instances of a NoSQL DB cluster. As soon as the desired platform is running, the web interface offers the possibility to scale some infrastructure components. In this contribution we describe the solution design and implementation, based on the application requirements, the details of the development of both the Heat templates and of the web interface, together with possible exploitation strategies of this work in Cloud data centers.

1. Introduction

In a Cloud environment, users that need to run an application can exploit a series of features that should simplify the deployment. The facility to start a virtual service, the possibility to run multiple instances in order to improve availability and to auto-scale the application on preconditioned states are characteristics that motivate developers to design their software to take advantage of Cloud features. On the other hand, deploying an application on Cloud could be a difficult operation, because it could consist of multiple components and use cases could require deployment features that are difficult to orchestrate, such as high availability, auto-scaling and disaster recovery features. Moreover, the complexity is accentuated if users have to deploy the application components on top of an Infrastructure-as-a-Service and are not familiar with this kind of environment. Beyond the initial
deployment, issues are represented by the reproducibility of start-up procedure on the same Cloud IaaS and, even more so, on different infrastructures based on the same Cloud stack.

In order to mitigate the complexity of deploying an application on a Cloud IaaS and to increase the reproducibility level of the procedure, we developed an abstraction layer that reduces the human intervention and the error chance during the application lifecycle. In this paper we describe the approach used to build this layer and the implementation in the framework of some projects.

2. Abstraction layer
This layer is composed by two abstraction levels: a Cloud side and a User side abstraction. By exploiting both these levels the user is capable to deploy an application not only without knowledge of the application details but also without directly interacting with the Cloud IaaS.

![Abstraction layer schema](image)

In figure 1 the abstraction approach is depicted. In a Cloud framework, the application and its dependencies are running on an IaaS. The Cloud side abstraction consists of a set of templates designed to work with the Cloud Orchestration service (e.g. OpenStack Heat), i.e. a programmatic approach to create and deploy full application configurations. A template declares all the virtual resources that are needed to run the application, the order of services instantiation and the configuration steps. The complexity of setting up application clusters for the basic IT services is hidden to the end user and handled entirely by the configuration management system. This reduces the risks to get the installation process wrong, as each step can be tested and improved in the template. Furthermore the deployment time is significantly shorter than manual setup performed by a trained system administrator. Once a template describing an application framework is ready, it can be used every time users need to deploy that application. This provides a significant advantage when
deploying recurring infrastructures and testbeds.

Running a set of templates requires the interaction with the Cloud administration layer. Designed to avoid this interaction, the second abstraction layer consists in a web interface exploiting the Orchestration service APIs. Through this interface the user can instantiate a complete application stack, possibly choosing configuration parameters and scaling the application by running new service instances.

3. Use cases

3.1. !CHAOS

Our approach has been implemented in the framework of the !CHAOS (Control system based on Highly Abstracted and Open Structure), a project started as a candidate Distributed Control System (DCS) and Data Acquisition (DAQ) system for the SuperB INFN Flagship Project and subsequently evolved into a fully autonomous project “!CHAOS: a cloud of controls” supported by MIUR and developed by INFN. The framework itself requires a number of basic IT services in order to run:

- a non-relational database for runtime configuration data
- a fast key-value store for keeping track of live process state
- a shared filesystem with POSIX semantics for long-term data archival
- a VPN concentrator for remote access (optional)

The entire software stack for basic IT services uses exclusively open-source software.

On these backend services rely the !CHAOS main services (!CHAOS Data Service - CDS and MetaData Service - MDS). We studied a solution for each of these services (both the common and the !CHAOS specific ones), to integrate them in a generic IaaS based on OpenStack. The solution is based on Heat, the OpenStack component that implements the function of infrastructure orchestration. It is an automated orchestration system capable of starting up a complete !CHAOS deployment on an OpenStack cloud environment and optionally allowing the system to scale as needed with little to no human intervention.

The main Heat template consists of a series of sub-templates for automated deployment of major services of !CHAOS in a Cloud environment:

- The template for MongoDB was derived from an open-source implementation by Rackspace and is capable of performing cluster bring-up in a completely unattended way.
- The Couchbase cluster template was partially derived from the MongoDB template, but the software configuration was entirely developed in-house. The setup is again fully unattended and can handle horizontal scaling of the Couchbase cluster via HTTP POST requests to an endpoint provided by the OpenStack Orchestration Service, without the need for human intervention on the Couchbase cluster itself.
- Network access services are implemented as a virtualized OpenVPN server and a Heat template that configures the OpenStack component to manage virtual network services (Neutron) to route traffic to/from the VPN. Additional security can be guaranteed by Neutron services (like Firewall-as-a-Service) if supported by the target Cloud.
- The shared POSIX filesystem is a virtualized Ceph cluster. It provides a 3-way replicated POSIX-like filesystem to !CHAOS services for archival. The same cluster may also act as an object storage if so configured. The setup is fully unattended and the filesystem size is only limited by the available quota on the target Cloud.
- The !CHAOS application components are deployed using a specialized template that populates the configuration files automatically with the major backend services addresses. Moreover, the shared POSIX filesystem is mounted automatically on all Data Service nodes.

On top of the custom templates lies a prototype of web interface developed in PHP language, exploiting the Heat APIs. Through this interface the user can instantiate a complete !CHAOS infrastructure, choosing the filesystem size and the number of MongoDB shards[5] and Couchbase
instances. As soon as the infrastructure is running, the web interface offers the possibility to scale the infrastructure, adding an instance to the Couchbase cluster. Figure 2 shows the complete !CHAOS deployment on an OpenStack Cloud environment realized through the use of this automated orchestration system.

![Image of !CHAOS cloud deployment architecture](image)

**Figure 2 - !CHAOS cloud deployment architecture**

### 3.2. Future implementations

#### 3.2.1. INDIGO-DataCloud

INDIGO-DataCloud (INDIGO for short) is a project started in April 2015, funded under the EC Horizon 2020 framework program. It includes 26 European partners located in 11 countries and addresses the challenge of developing open source software, deployable in the form of a data/computing platform, aimed to scientific communities and designed to be deployed on public or private Clouds and integrated with existing resources or e-infrastructures.

One of the main objectives of the project is to develop and implement a solution that is able to deploy in a transparent and powerful way both services and applications in a distributed and heterogeneous environment (EGI - European Grid Infrastructure - Grid and Federated Cloud, IaaS Cloud, HPC clusters, etc). In order to address this objective, we identified different areas where our solution can provide an important contribution:

- Standardization of the Orchestration service based on TOSCA templates
- Standard Web interfaces as Future Gateway programmable interface
- Dynamic deployment of a virtual site for interactive analysis
- Virtual batch systems on opportunistic cloud
We will continue our work by creating, or just improving the existing, templates in order to address as many as possible of the above points.

3.2.2. OpenCityPlatform

Open City Platform (OCP) is an industrial research project funded by the Italian Ministry of University and Research (MIUR), started in 2014. It intends to research, develop and test new technological solutions open, interoperable and usable on-demand in the field of Cloud Computing, along with new sustainable organizational models for the public administration, to innovate, with scientific results, with new standards and technological solutions, the provision of services by the Local Public Administration (PAL).

For this project, in the following months the activity will concentrate in demonstrating the usefulness of our solution for:
- Automated deployment and scaling of high-demand applications for Public Administrations
- Design and reengineering of Cloud applications
- Cloud Formation and Disaster Recovery as a Service
- Integration of PaaS components, in particular PaaS for eGov
- Open Data and the Open Service Engine and integration into business models

4. Conclusions

The work performed and presented in this article contributed to the result of a "!CHAOS as a service", deployable on open-source cloud infrastructures. This solution reduces the time of framework deployment and permits to instantiate all the !CHAOS frontend and backend services with minimal human intervention. Possible future improvements could be the deployment of services on different regions (i.e. IaaS running on different data centres) to provide geographical redundancy for services and a solution to auto-scale deployed infrastructures on the basis of monitoring metrics calculated on the running !CHAOS application instance.

Moreover, the approach of abstracting the complexity of application deployment through an orchestration service and an overhead Web interface can be applied to other use cases for data center users and administrators, i.e. deployment of recurring infrastructures or testbeds. We already identified the projects INDIGO - DataCloud and OCP that can benefit from our experience of the presented solution, for which the following months and activities will be concentrated on the creation of some PoC regarding the different areas of the two projects.

References

[1] Antonucci F et al, 'CHAOS: a cloud of controls - MIUR project proposal', INFN-14-15/LNF
[2] INDIGO – DataCloud project web page - https://www.indigo-datacloud.eu
[3] OpenCityPlatform project web page - http://www.opencityplatform.eu
[4] OpenStack Orchestration - Heat project - https://wiki.openstack.org/wiki/Heat
[5] MongoDB Sharding - https://docs.mongodb.com/manual/sharding