Declarative Modeling for Cloud Deployments

Giuseppe Attardi, Alex Barchiesi, Alberto Colla, Fulvio Galeazzi, Gianni Marzulli, Paolo Velati
Consortium GARR
Via dei Tizii 6
Roma, Italy

Cloud computing offers several benefits: resources can be allocated on demand, scaling according to varying usage patterns and eventually reducing the costs for individual groups to provision and maintain their own infrastructure. Providing cloud computing services besides connectivity is a natural evolution for NRENs as users demand such services and also because the networking infrastructure itself is evolving towards using virtualization techniques.

Building a cloud platform is though a daunting task, that requires coordinating the deployment of many services, interrelated and dependent on each other. Provisioning, servicing and maintaining the platform should be automated.

For the deployment of the GARR Federated Cloud Computing Platform, we chose an *intent-based* approach, which relies on *declarative modelin*g for specifying the requirements of the service provisioning, describing the parts that compose the system, any specific constraint requirements and the supplier/consumer relations between them. An automated orchestration tool analyzes the model, compares it with the current state of the system being deployed, determines the resources that need to be provisioned, generates a sequence of executable steps needed to bring the deployment in line with the model, and coordinates the execution of those steps until the system reaches a configuration that satisfies all constraints.

Intent-based deployment is starting to be used also in networking through *Intent-Based Networking System*s (IBNS), which allow for automation at scale, building and operating networks through mechanisms that support full life cycle management of network infrastructures. IBNS provide "networking middleware to replace intelligence that was previously only available from networking engineers/architects", says a Gartner report. “The intelligent systems are driven by algorithms which translate business intent into network configurations”.

Cloud system architects express the intent of their design in a declarative fashion by specifying the service functionality abstracting from the details related to service deployment. The approach offers these benefits:

- Portability: models are described in a declarative fashion, abstracting from the specifics of a cloud provider and hence they can be ported across different cloud platforms
- Consistency: both physical and virtual infrastructure can be modeled, as well as the relationships between infrastructure, network, and application components.
- Automation: mapping a model into infrastructure and cloud-specific deployment operations is the task of the Orchestrator rather than of a system administrator.

We chose a declarative service modeling approach for orchestrating three types of deployments:

1. Deployment of the cloud platform itself: defining the architecture of the cloud through a

declarative model, allows us to deploy multiple copies of the platform over different regions, to scale the platform easily, to add components to the platform to enrich its functionality.

2. Deployment as a Service. We provide the same orchestrator tool to cloud users, so that they can deploy and scale cloud applications in a fully automated fashion.
3. Deployment of specialized versions of the platform. For example we have added a Container Platform to our basic IaaS platform.

The *Container Platform* is described declaratively by a model like the following, that specifies the required components (Kubernetes master and worker nodes, distributed resilient storage, distributed key-value store, cluster monitoring dashboard, federated authentication provider, etc.):

Figure 2. Declarative Model for a Kubernetes cluster.

The model also describes constraints about the units, e.g. requirements on the capabilities of specific nodes (like the availability of GPUs on worker nodes), and relations among the units, for example the master/worker relation between nodes, the connections to the storage module. The orchestrator for the model generates a plan of deployment, selecting the physical server to be used, deploying the OS and installing the software for the components assigned to those nodes. Scaling of the cluster and the application is also performed automatically, just by changing the number of nodes requested expressed in the model.

The deployment of federated regions is facilitated by sharing the model of the cloud architecture with anyone wishing to join and to contribute resources to the federation. Setting up a region and joining the federation is a task that can be accomplished in just a few hours. Federated resources are integrated into a single cloud platform available to any user of the federation.

Finally, by providing the same orchestrator to cloud users, we offer *Deployment as a Service* (DaaS), which allows self-service deployment of cloud applications. The cloud thus addresses the long tail of science, allowing researchers of any discipline, without expertise in system or cloud administration, to deploy applications readily available for their peruse. Ready-made application can be selected from a public catalogue and can be deployed with a few clicks through a web GUI.

This article reports the major features of the design and deployment of the GARR Federated Cloud,

involving authentication, workload management, storage provisioning, monitoring and operations. In particular it will describe how we adapted the OpenStack basic facilities to the needs of a federation of multiple independent organizations, how we control resource allocation according to committed plans and correspondingly how we handle accounting and billing of resource usage.
The code for the accounting system is available as OpenSource, to be shared with other NRENs.

The GARR Cloud Computing Platform is currently in use by over 700 users, who deployed over 1200 VM. The infrastructure consists of 5 regions, with a total of about 9000 vCPUs and 10 PB of storage. Several universities are in an advanced state of planning for joining the federation.

This work has been done in part within task JRA-4 of Géant project GN4-2.

References

1. OpenStack. http://docs.OpenStack.org
2. The Federated Cloud Platform Reference Architecture. 2017. https://cloud.garr.it/_downloads/ReferenceArchitecture.pdf
3. A. Parameswaran, A. Chaddha. Cloud Interoperability and Standardization. SETLabs Briefings - Infosys, vol. 7, no. 7, pp. 19-26, 2012.
4. The Business Times. The next leap in networking. October 2017. http://www.businesstimes.com.sg/hub/empowering-enterprise/the-next-leap-in-networking-infrastructure