Chapter 8
Virtual Java Service Container

Víctor Gil
Oracle Corporation, Czech Republic

1. ABSTRACT

This chapter provides an overview of the architecture, installation, and usage of the Virtual Java Service Container (VJSC). It covers Version 1.0 of the VJSC. It also presents the motivation for such a solution and its position among other Java cloud platforms.

VJSC provides a framework to deploy and manage Java Services in the cloud. It can be used as a standalone solution, but it leverages its full potential when integrated with a cloud broker system, such as OpenNebula (2011). This integration effort, which was implemented in the scope of the RESERVOIR (Reservoir FP7, 2011) EU 7th Framework Programme, is described as well.

First, the authors present several cloud solutions, then they describe the architecture of the VJSC framework, including practical coding and usage examples, followed by the lessons learned as a wrap-up.

2. INTRODUCTION

Cloud computing has emerged as one of the fastest growing sectors inside the Information Technology industry. It has changed the way we think about IT infrastructure management.

Providers of software Services are now able to outsource the operation of the hardware platforms required by those Services. Cloud computing is a natural evolution of the widespread adoption of virtualization. Today virtualization technologies are very mature. Previously defined virtual machines can easily be deployed on cloud providers' data centers. Services running inside the virtual machines can then be easily managed. But some important features are still underdeveloped, such as the deployment of native applications without the creation of virtual machines.

The VJSC framework provides unified access to Java application containers (such as OSGi [2011] Containers or Servlet Containers [Oracle, 2011a]) and the Services running in them. It is deployed on top of existing technologies, such us Java SE 6 (Oracle, 2011b) and OSGi. The framework controls the life-cycle and gives access to the Services. It handles different kind of management operations such as migration.

DOI: 10.4018/978-1-4666-0098-0.ch008

2.1 Why Java?

As one of the most used enterprise platforms, Java is widely utilized to develop Services. The Java community numbers more than 6.5 million developers, it is the largest and most active on the planet. Being able to use a framework to natively deploy those Services inside clouds would create a major enhancement over present cloud computing technologies. By removing one software layer (i.e., the guest OS), performance increases, deployments are faster and complexity is reduced. A Java Service deployment can be timed in seconds, while provisioning and starting a Virtual Machine typically takes at least several minutes. Demand for such a solution is confirmed by investments made by companies such as Google.

2.2 State of the Art

This section tries to sum up the state of the art of existing technologies for hosting Java Services in clouds. The solutions are classified using adopted aaS (as a Service) taxonomy and compared to VJSC architecture. The first solution provides a preconfigured software stack for building and running Java Services enhanced with provisioning and management tools. The second one provides restricted Java runtime environment abstracting underlying specific infrastructure. Finally, the third one ports a proprietary Platform as a Service solution to a generic infrastructure.

2.2.1 IaaS: Cloud Foundry

Cloud Foundry (CloudFoundry, 2011) provides an integrated toolset to deploy enterprise Java applications using Amazon Web Services (Amazon, 2011a). The toolset is built on the open source Cloud Tools project (Cloud Tools Project, 2011). The Cloud Tools project provides a framework to provision and manage a Java Service stack built from Apache Tomcat (Apache, 2011a), Apache HTTP Server (Apache, 2011b), MySQL (Oracle, 2011c) and other components. The stack is provisioned with preconfigured Amazon Machine Images (AMIs) (Amazon, 2011b).

2.2.2 PaaS: Google App Engine

Google AppEngine (Google, 2011) is a PaaS, which enable users to run Web Services. Applications are built on Java 6 runtime environment with standard Java Servlets interface. Actions that could affect performance and scalability of other applications are restricted. Java 6 standard VM (Virtual Machine) as the base of the platform enables straightforward integration of other JVM languages and frameworks. This platform also provides APIs for Services like storage, email, caching, authentication, image manipulation, etc.

2.2.3 PaaS to IaaS: AppScale

AppScale (UC Santa Barbara RACELab, 2011) open platform enables the user to run a Google App Engine application outside of Google's infrastructure. It is built on the Google AppEngine SDK (Software Development Kit), replacing the non-scalable API (Application Programming Interface) implementation with pluggable, distributed and scalable components. Platform provides AMIs for Xen (2011) and KVM (2011) hypervisors. These images can be used to run applications on Amazon EC2 (Amazon, 2011c) or Eucalyptus (Eucalyptus Systems, 2011) frameworks.

2.2.4 Evaluation

The major issue of PaaS platforms like Google AppsEngine is the restriction to the Web Service execution model. This limits the usability of the platform for other types of applications.

VJSC overcomes the limitations of the PaaS solutions, plus providing an XML-RPC (UserLand Software, 2011) interface which allows Service Providers to have total management control and