Chapter 11

Building Clouds: An Integrative Approach for an Automated Deployment of Elastic Cloud Services

Leonard Heilig
University of Hamburg, Germany

Stefan Voß
University of Hamburg, Germany

Lars Wulfken
Amazon Web Services, Germany

ABSTRACT

The highly automated and scalable nature of cloud computing encourages practitioners and scholars to reconsider software delivery processes. To efficiently leverage the benefits of elastic clouds, applications are typically provisioned and deployed several times a day in different environments and regions. Related procedures must support agile development and deployment of software components as well as the associated management of different versions and configurations. To efficiently utilize auto-scaling mechanisms, the latency produced by provisioning and deployment activities needs to be reduced while ensuring consistency, repeatability and reliability. In this chapter, an integrative process-oriented approach to improve the overall quality and performance of deployment processes is presented. Based on a comprehensive analysis of requirements, the proposed process considers the link between deployment and configuration management as well as tools and organizational aspects. Further, related research challenges and a case study focusing on the Adobe Shared Cloud are presented.

1. INTRODUCTION

Cloud computing fundamentally transforms the way software applications and hardware systems are designed, delivered and consumed (Marston et al., 2011). Several economic and technical potentials are associated with adopting cloud computing, but most important are a high degree of automation, the option to dynamically scale applications according to unexpected or anticipated demand, and usage-based pricing models without involving monetary investments. In this regard, Chang et al.
(2014) describe characteristics of cloud computing and provide an extensive overview on cloud adoption frameworks. Especially software companies can benefit from the high flexibility and cost-effectiveness of cloud computing. Typically, software delivery involves several complex activities being carried out in several environments, before software components become finally available to customers. Within these deployment processes, the quality of software components needs to be extensively validated in order to comply with functional and non-functional quality of service (QoS) requirements as well as to avoid outages once software components are delivered to customers. In particular agile development teams require efficient deployment processes allowing an autonomous and continuous integration of components within short development cycles. During operations, the performance and quality of software components must be ensured under volatile demands and conditions. In contrast to traditional computing infrastructures, the high elasticity of cloud infrastructure allows cloud application providers to adapt their services according to certain conditions, for example, by automatically adding or removing underlying computing resources. Economically, the latter implies that unnecessary expenses for unused or underused infrastructure can be reduced (Rodero-Merino et al., 2010). Auto-scaling mechanisms, however, require that involved software components are deployed within a very short time span. As provisioning and deployment activities normally take, if automated, several minutes to complete, novel mechanisms are required to significantly reduce deployment time. Consequently, a high degree of deployment automation is essential for ensuring the performance and quality of software components throughout their entire lifecycle from development to phase out. More specifically, automated deployment is essential to realize continuous integration and a cost-efficient and QoS-compliant scaling behavior. To leverage the high degree of automation in cloud environments, however, cloud application providers must establish standardized deployment processes by considering best practices, tools and organizational aspects.

In general, the overall complexity and heterogeneity of cloud infrastructure and deployed application services is growing (Breitenbücher et al., 2013). A prerequisite for efficiently managing complex software systems and enabling automated deployment is the adoption of configuration management practices. In this regard, it is important to analyze specific requirements of cloud-based software components in order to efficiently deliver and manage multiple versions of those components in different cloud environments (e.g., testing, staging, production) and multiple regions (e.g., EU, US, etc.) while ensuring consistency, a high service level and business continuity. For example, provisioning activities to build underlying infrastructure resources need to be adapted to the deployment of software components. These provisioning activities can be standardized and automated by using dedicated interfaces of the cloud infrastructure provider. A high degree of standardization facilitates that processes are repeatable, testable, auditable and accurate (Karunakaran, 2013). Furthermore, auto-scaling mechanisms must be integrated to trigger deployment processes for scaling software systems to changing conditions. To fully utilize cloud capabilities both in development stages and during operations, it becomes increasingly important to establish a standardized end-to-end process model connecting all these different activities by simultaneously considering cloud-specific requirements, tool support and organizational aspects.

Although configuration management is an inherent part of most development and deployment activities in most software organizations, a lack of integrative cloud-based end-to-end approaches can be identified both in practice and academia. An integrative approach is essential for improving the overall quality and performance of software delivery and operations of elastic software components in cloud-based environments. To fully understand the