ISSN 2347-4289

Effective Portable Cloud Services Using TOSCA

K. Veeravel

Department of MCA, VSB Engineering College, Karur, Tamilnadu
kveeravelmca@gmail.com

ABSTRACT: The ability to move cloud services and their components between providers ensures an adequate and cost-efficient IT environment and avoids vendor lock-in. Research has already addressed movability and migration on a functional level.1,2 However, no one has yet examined cloud service portability with regard to management and operational tasks, which are a significant and increasing cost factor. One reason is the lack of an industry standard for defining composite applications and their management. Without an appropriate standardized format, ensuring compliance, trust, and security the biggest area of critique preventing the cloud's wider adoption is difficult. Dealing with these challenges in industry and research has the potential to bring cloud computing to the next level. TOSCA will enable the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behavior of these services (e.g., deploy, patch, shutdown)--independent of the supplier creating the service, and any particular cloud provider or hosting technology. TOSCA will also make it possible for higher-level operational behavior to be associated with cloud infrastructure management.

INTRODUCTION

For cloud services to be portable, their management must also be portable to the targeted environment, as must the application components themselves. Here, the authors show how plans in the Topology and Orchestration Specification for Cloud Applications (TOSCA) can enable portability of these operational aspects underneath all the hype, the essence of cloud computing is the industrialization of IT. Similar to mass production lines in other industries (such as the auto industry), cloud computing standardizes offered services and thus increases automation significantly. Consequently, enterprises are increasingly utilizing cloud technology; however, major challenges such as portability, standardization of service definitions, and security remain inadequately addressed. In the existing system portability, standardization of service definitions and security remain inadequately addressed. The ability to move cloud services and their components between providers ensures an adequate and cost-efficient IT environment and avoids vendor lock-in. The movability and migration on function level with regard to management and operational tasks, which are a significant and increasing cost factor. In the offering phase, the cloud service provider creates a cloud service offering based on a service template, adding all provider- and offering-specific information. This includes aspects such as pricing and specific technical information such as IP address range and application configurations. Finally, the offering is published in a service catalog. In the subscription and instantiation phase, the cloud service consumer browsing the service catalog can select and subscribe to the respective offering. The consumer customizes the service through points of variability (for example, selecting "small," "medium," or "large" for the service's size), signs a contract, and accepts the offering's terms and conditions. This subscription process triggers the instantiation of the cloud service instance. The cloud management platform aggregates all the required resources from the common resource pools, for example infrastructure components, and automatically deploys, installs, and configures the service's necessary pieces.

Cloud Computing Reference Architecture(CCRA):

Cloud Computing Reference Architecture (CCRA) defines three main roles typically encountered in any cloud computing environment: the cloud service creator, cloud service and cloud service consumer. the service creator develops cloud services; the service provider runs those services, and provides them to service consumers, which can also be IT systems. Consumers can be billed for all (or a subset of) their interactions with cloud services and the provisioned service instances.

Cloud Service Life Cycle:

In the definition phase, all management knowledge required for the specific cloud service in particular, how to instantiate is captured in a service template. This knowledge includes information about the cloud service's internal structure along with operational and management aspects to build, manage, and terminate cloud services. In the offering phase, the cloud service provider creates a cloud service offering based on a service template, adding all provider- and offering-specific information. This includes aspects such as pricing and specific technical information such as IP address range and application configurations. In instantiation phase, the cloud service consumer browsing the service catalog can select and subscribe to the respective offering. The consumer customizes the service through points of variability signs a contract, and accepts the offering's terms and conditions. In the life cycle's *production* phase, the cloud management platform uses management plans to manage the service instance for compliance with the service-level agreements (SLAs) negotiated at subscription time.

Topology and Orchestration Specification:

TOSCA describes composite applications and their management in a modular and portable fashion. It thus defines service templates that contain a cloud service's topology and its operational aspects to deploy, terminate, and manage . The creator of a cloud service captures its structure in a service topology graph with nodes and relationships. TOSCA specifies the metamodel for types and templates that is, the language fordefining them. Concrete types aren't part of the specification and must be standardized by the respectivedomain expert groups. Nodes in particular define a range of information to deploy, terminate, and manage the cloud service: *instance states* represent the node's internal state during production as part of the topology

ISSN 2347-4289

Planning Of Topology:

Plans use the cloud service topology in three ways: First, plans orchestrate the management interfaces and operations defined in TOSCA nodes. Operations are described in the Web Services Description Language (WSDL), Representational State Transfer (REST), or scripts that implement particular management operations on the respective node. Second the service template is instantiate . Second, plans can inspect the topology model to access a service's nodes and relationships. This enables flexible plans whose behavior is based on the topology and changes therein the respective nodes. Finally, plans read and write a service's instance information that is, the node's instance state, such as properties containing credentials, IP addresses, and so on.

Portability of plan:

TOSCA plans' portability is inherited from the workf low language and engines used . In TOSCA, these services are explicitlydescribed in the nodes as operations. For operations referencing external services, portability isn't a problem. TOSCA models contain myriad references to files, such as those with additional TOSCA definitions, plans, and other artifacts. **T**OSCA allows service creators to gather into plans those activities necessary to deploy, manage, and terminate the described cloud service.

CONCLUSION:

To address the third major challenge of cloud computing we identified in the introduction, we're researching how to ensure compliance and security throughout the entire service life cycle. Security experts are enabled to annotate policies to TOSCA elements. Policy-specific plug-ins will then enforce these policies in the management environment during the cloud service's whole life cycle. TOSCA provides an exchange format for policies but no guarantee that the management environment and providers also enforce these policies. This must be ensured by a trustworthy entity through certification of management environments, providers, and services. TOSCA models contain myriad references to files, such as those with additional TOSCA definitions, plans, and other artifacts. Consequently, future work must be invested toward a TOSCA archive, a self-contained package of these artifacts, similar to Java EAR files. This cloud service could then be offered in the marketplace and, due to its portability, operated in different management environments.

REFERENCES

[1]. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50–55, 2009.

[2]. S. Kamara and K. Lauter, "Cryptographic cloud storage," in RLCPS, January 2010, LNCS. Springer, Heidelberg.

[3]. A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.

[4]. I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.

[5]. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of S&P, 2000.

[6]. E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, 2003, http:// eprint.iacr.org/2003/216.

[7]. Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. of ACNS, 2005.

[8]. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS, 2006.

[9]. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. of EUROCRYPT, 2004.

[10]. M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in Proc. of CRYPTO, 2007.