ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 3, March 2013

TOSCA ENABLING CLOUD PORTABILITY

(SahithiTummalapalli,RaviKanth.P,Yuvaraj.K, SreenivasVelagapudi)

ABSTRACT:
The cloud is a computing architecture characterized by a large number of interconnected identical computing devices that can scale on demand and that communicate via an IP network. Many technologies commonly associated with computing cloud architectures are not intrinsically associated with them and could just as easily be provided by a different architecture. The reality is that using any technology, except the most primitive, causes some degree of dependency on that technology or its service provider. This notion of using the cloud computing features created by one provider by another causes vendor lock-in.To avoid this vendorlock-in and for making the cloud services portable,their management must be portable to the targeted environment and application components themselves.Here,in this paper we show how the portability of these operational aspects can be enabled using Topology and Orchestration Specification for Cloud Applications(TOSCA).

KEY WORDS:

Vendorlockin,Portability,TOSCA,plans,work-flows, service template.

1.INTRODUCTION:

Cloud computing is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand.It is a culmination of numerous attempts at large scale computing with seamless access to virtually limitless resources.Irrespective of the several advantages of cloud computing,one of the key drawbacks of investing in an IT solution is lock-in.vendor lock-in defined as the situation in which a customer using a product or service cannot easily transition to a competitor's product or service. The inability to move cloud services and their components between providers results in an inadequate IT environment and causes vendor lock-in.Management and the operational aspects are the significant and increasing cost factors of cloud services portability which have not been addressed so far.One of the important reason for this, is the lack of an industry standard for defining applications and their management. Without an appropriate standardized format ensuring compliance, trust,and security the biggest area of critique preventing is difficult.Apart from these, significant growth in hybrid cloud is driving the need for interoperability and openess across on-premise and public cloud environments.
Here, we describe how the portable and standardized management of cloud services is enabled through the Topology and Orchestration Specification for Cloud Applications (TOSCA), a recently initiated standardization effort from OASIS.TOSCA formalizes a significant amount of interactions between somebody who develops IT delivered services and the one who actually operates them. The goal of TOSCA is to improve the portability and manageability of applications by composing a service once and playing it on any cloud.

2.ROLE OF TOSCA:

TOSCA is a specification which adds value to the relationships between user,providers and developers of IT provided services.The roles are oriented on a model where a cloud service developer provides services which they distribute via further channels,primarily cloud service providers and which are eventually offered to service consumers.TOSCA provides a compelling value statement for each role and its corresponding actor and the reason why it makes sense to use the TOSCA specification for those who develop the cloud services and for who deploy and operate them is defined below:
Cloud service consumer benefits indirectly from the standardization which TOSCA brings to the cloud service developer and cloud service provider.These benefits consumer by lowering the setup and the operational costs from TOSCA automation.A cloud service developer uses TOSCA as the standard to get their developed services at cloud service providers in place.By this,cloud service developers are able to

choose from a wider range of cloud service providers and can work with multiple cloud service providers in different legal environments with reasonable effort.A cloud service provider uses TOSCA to rapidly offer and deploy cloud services developed by cloud service developers for cloud service consumers.Cloud service providers act as resellers for sevices developed by cloud service developers,they can optimize deployment and operational procedures and expenses and can extend service offerings and revenue chances.Organizations or persons in these roles typically apply to separate market actors but one actor may also serve in multiple roles.

3.TOSCA SERVICE TEMPLATE :

TOSCA provides a format for describing a service's application topology in the form of a service template and a description how to manage this application during the applications lifecycle i.e how to deploy terminate or shutdown a service. Such a package is called a TOSCA Service Template. With this, TOSCA specifies a generic language to model services and their management. However, a TOSCA Service Template does not provide a representation of a business process that is using the application components. That is, the relations between the corresponding activities of a business process are not captured but the application topology underneath.

A.TOPOLOGY OF TOSCA:

The topology is a graph of nodes and edges,which represents the structure of the service. Both Node and Relationship Templates reference a type, the Node Type and Relationship Type respectively. The type defines the kind of properties and the lifecycle,the templates must follow. The Node Type, for instance, refers an XML Schema Definition (XSD) and defines the states of the lifecycle of this specific type , which both must be followed by the Node Template. Node types must be defined that are referenced from a node template of a topology template. Node Types is the definition of Interfaces which define the operations possible on Node Templates of this type. Interfaces define which management operations are supported by this node.Some particular node types will occur frequently like an "DBMS" node type and "operating system" node type etc.The nodes are connected by relationships which are binary i.e connecting two nodes which are also directed and may have many cycles.Both nodes and relationships are typed into and hold into a set of type specified properties,bringing meaning and variability to these generic TOSCA elements.

B.Plans in TOSCA:

TOSCA enables application developers and operators to model management best practices and reoccurring tasks explicitly into so-called plans.TOSCA supports built in "build and management plans" that help to set up,operate,maintain and tear down the service i.e a complete life cycle management .These management plans,which operate directly on the topology description are created by the experts in the specified fields.This allows each party cloud service provider and subscriber to focus on their key functions because it decouples cloud infrastructure from cloud content. .If these plans were portable between different environments and providers,the achieved reusability and automation of service management would significantly reduce the total cost of ownership. Plans are used to manage the lifecycle of an application.The idea driving this is that the organization which has the experience and technical knowledge how to operate and manage this cloud service models incorporate their knowledge and best practices as plans.This enables the management of cloud services to be reusable and portable,because users are able to execute predefined plans without requiring extensive knowledge of the service itself.By using standard workflow languages arbitrary web services can be invoked as needed by the respective plan. Instead of introducing a new way to define workflows,plans use existing workflow languages such as Business Process Model andNotation(BPMN) or the Business Process Execution(BPEL). BPEL is an OASIS standard executable language for specifying actions within business processes with web services.Processes in BPEL export and import information by using web services interfaces exclusively. BPMN is a graphical representation for specifying business processes in a business process model.It was previously known as Business Process Modeling Notation.Recently,BPMN4 is introduced which is a extension of BPMN2.0.It allows accessing the elements of a service topology directly,which in turn eases the development of management plans.Moreover,Plans can use any workflow language supported by a management environment.

4.ADAPTING TOSCA:

The challenge of making cloud providers interchangeable is targeted by TOSCA which

provides a common language to describe application topologies. Developed application components make assumptions on each other and their runtime environment. These assumptions are not covered by a TOSCA description. However, they also have to be considered when moving components between environments. For instance, the availability of virtual machines found in public clouds is often lower than the availability guaranteed by private cloud environments. During the migration between such different environments, these different properties have to be regarded by changes in the application architecture to cope with component failure.Additionally,describing details like how clouds offer resources,who is accessing the cloud and how the services of the offerings found in these clouds behave allows the classification of the clouds and their offerings with respect to the supported patterns and enables the correlation of different cloud architectures that application developers need to adapt when using a concrete cloud offering.

To adapt a given application described by a TOSCA Service Template, we need additional information on how the topology of the given applications actually looks like, i.e., which components are used in which way. The application topology of an application described in TOSCA contains one or more Node Templates that describe the components a serviceis built of. To find proper substitutes for adapting either the whole application or specific parts there of these Node Templates need to be annotated by the proposed Cloud services that indicate which operations are used for the implementation of this node. The Cloud applications catalog in the offering and instantination phase provides information that help developers to identify required Cloud services that should be used when performing a certain operations, e.g.,useful services that should be implemented when moving an application component from a private to a public Cloud.

Based on our assumptions the adaptation of an application always requires adapting the original TOSCA Template that describes the complete application of interest. These changes may affect both the Topology Template as well as the Plans that are included. Within the Topology Template single Node Templates or groups of Node Templates may be replaced by a new Node Template that implements the new functionality. Changing parts of a TOSCA Service Template affects both the relations between different Node Templates and the dependencies to existing Plans. To exchange specific Node Templates TOSCA provides import functionality for other TOSCA Templates that allows the usage of Node Templates. In some cases it might also be sufficient to only adapt existing or add new Plans to a given TOSCA Template. For example, a new Plan may describe how a specific component is suspended in times of no usage. The Topology Template is not affected in this case.After creating service template definition and related artifacts as a cloud service archive using TOSCA ,the cloud management consumes service definition which is available as a new service in the service catalog.

5.WORKING WITH TOSCA:

A topology template and its corresponding plans are referred to collectively as a service template. The executables required to actually instantiate, run and manage the cloud application are packaged with the service template into a CSAR (i.e. a Cloud Service ARchive). Typically, a CSAR is a self-contained and portable representation of a cloud application that can be deployed and managed in an environment that supports TOSCA. TOSCA supports processing of service templates in two different ways. The first, referred to as imperative processing of a service template, requires the complete topology of a cloud application to be explicitly defined, as well as all of its management behavior by means of plans i.e it specifies precisely how a cloud application is structured and managed . The second way, referred to as declarative processing of a service template, is based on the assumption that it is possible to infer the management behavior of the corresponding cloud application; this typically requires the precise definition of the semantics of node types and relationship types and their correct interpretation within a TOSCA environment and this precisely specifies what structural elements of acloud applications are needed and what management behavior is to be realized.

There are different roles invoved in modeling a cloud application using TOSCA and these includes the type architect,the artifact developer and the application architect. Each of these roles is a specialization of the generic role cloud application service developer. Depending on the application being developed, the same person may fulfill more than one of these specialized roles. Concerted actions of these three roles are required to create a TOSCA service template and a corresponding TOSCA Cloud Service Archive (CSAR). The type architect is an expert for the types of components required by applications as well as the various types of connections between these components. This especially includes knowledge of the local management behavior of such

components and connections, which is defined as the operations of these components and connections. Types of components are defined as TOSCA Node Types, and types of connections are defined as TOSCA Relationship Types. The artifact developer is an expert in code artifacts. They are in charge of providing and describing the installables and executables required to instantiate and manage a cloud application. For this purpose, the artifact developer defines the corresponding TOSCA Node Type Implementations and Relationship Type Implementations. The application architect is an expert in both, the overall structure of a cloud application, its composite types and artifacts, as well as its global management behavior covering its complete lifecycle. The structure of a cloud application is specified by means of a topology template. Thus, the application architect identifies and defines the node templates making up a cloud application as well as the relationship templates wiring the collection of node templates into the topology of the cloud application. There are two other roles that may be concerned with TOSCA modeled cloud applications: the cloud service consumer and the cloud service provider. The cloud service consumer makes use of a modeled cloud application service, e.g. by following the self-service paradigm, browsing a cloud service catalogue and deciding to subscribe to a particular cloud service. The cloud service provider offers an environment in which cloud services can be run, especially provisioned, managed, and decommissioned.

6.TOSCA EXTENDING THE EXISTING CLOUD STANDARDS:

Extending the existing standards involves mapping TOSCA to DMTF OVF.It is the deployment artifact of a node in a TOSCA service template that can be represented by using an image definition such as an OVF package.TOSCA assembles and orchestrates virtual images into larger structures and relates it to existing infrastructure.Tosca extends other cloud standard DMTF Common Information Model (CIM). TOSCA has a higher level of abstraction to describe both applications and infrastructure components, enabling cloud orchestration .In these DMTF CIM,TOSCA supports a richer set of relationships to express cloud application topologies.TOSCA also extends DMTF Cloud Infrastructure Management Interface (CIMI) in which TOSCA orchestrates appropriate behavior to enable CIMI infrastructure management to do the right thing at the right time.The other cloud standard which TOSCA extends is OASIS Service Component Architecture (SCA) .TOSCA extends SCA by providing topologies and orchestration services that bring together SCA applications with cloud infrastructure.

7.CONCLUSION:

The design and prototypical implementation of the extendable TOSCA framework that allows for a convenient specification of the requirements covering the domain-specific handling of the TOSCA elements' properties. An important requirement at this juncture was an architecture that enables the assessability of the intermediary results, i.e. a human domain expert can evaluate the determined correspondences and manipulate them if desired.The popularity of these services would be greatly enhanced if a cloud service stack based on the current crop of cloud standards made moving individualized services from one provider to another.

8.REFERENCES:

1. Tobias Binz,Gerd Breiter,Frank Leymann, Thomas Spatzier " Portable Cloud Services Using TOSCA",IEEE Press,2012.
2. T. Binz, F. Leymann, and D. Schumm"CMotion: A Framework for Migration of Applications into and between Clouds,"Proc. Int'l Conf. Service-Oriented Computing and Applications, IEEE Press, 2012.
3. M. Behrendt et al., "IBM Cloud Computing Reference Architecture," Open Group submission, Feb. 2011;www.opengroup.org/cloudcomputing/uploads/40/23840/CCRA.IBMSubmission.02282011.doc.
4. G. Breiter and M. Behrendt, "Lifecycle and Characteristics of Services in the World of Cloud Computing," IBM J.Research and Development, vol. 53, no. 4,2009.
5. Web Services Business Process Execution Language (BPEL) Version 2.0., OASIS specification, 2007.
6. Topology and Orchestration Specification for Cloud Applications (TOSCA), OASIS specification, Oct. 2011; www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca.
7. Business Process Model and Notation(BPMN) Version 2.0, Object Management Group specification, Jan. 2011.
8. F. Leymann et al., "Moving Applications to the Cloud: An Approach Based on Application Model Enrichment," Int'l J. Cooperative Information Systems, vol. 20, no. 3, 2011.

9. Jorge Cardoso "Description and portability of cloud services with USDL and TOSCA"
10. Topology and Orchestration Specification for Cloud Applications (TOSCA) Primer Version 1.0. 31 January 2013. OASIS Committee Note Draft 01. http://docs.oasis-open.org/tosca/tosca-primer/v1.0/cnd01/tosca-primer-v1.0-cnd01.html.

AUTHORS:

- SahithiTummalapalli is currently pursuing final year of bachelor degree program in the department of computer science and engineering in KLUniversity,Vaddeswaram, Guntur,AndhraPradesh,India,9494653080.
- Ravikanth.P is currently pursuing final year of bachelor degree program in the department of computer science and engineering in KLUniversity,Vaddeswaram, Guntur, Andhra Pradesh,India,.
- Yuvaraj.K is currently pursuing final degree ofbachelor degree program in the department of computer science and engineering in KLUniversity,Vaddeswaram, Guntur, Andhra Pradesh,India.
-
- Sreenivasvelagapudi is currently working as assistant professor in computer acience and engineering department, kluniversity,Vaddeswaram,Guntur,AndhraPradesh,India,