

Amenesik

TOSCA Model

Specifications Version 1.0a

Iain James Marshall
29/03/2016

Table des matières

Introduction	1
TOSCA Definitions	2
Import	2
Location.....	2
Service Template.....	2
Name	3
Tags	3
Node Template	3
Node Type	3
Node Implementation.....	3
Node Relationship Type	4
Node Relationship Implementation.....	4
Requirement Type	4
Capability Type.....	4
Artifact Type.....	4
Policy Type	4
Mapping	5
Examples	6
Single Node	6
Web Server with Database	6
References	7
OCCI.....	7
TOSCA.....	7
CIMI.....	7
CORDS	7
AMENESIK	7

Introduction

The Remote Command Interface of the Amenesik Cloud Engine has been extended to offer the use of TOSCA standard for the description of Cloud Applications and their subsequent deployment and management. This API performs transformation of a TOSCA definitions document resulting in the corresponding CORDS Manifest document being introduced into the Amenesik Cloud Engine. This document describes the TOSCA type and template definitions and the way in which they are used during this process.

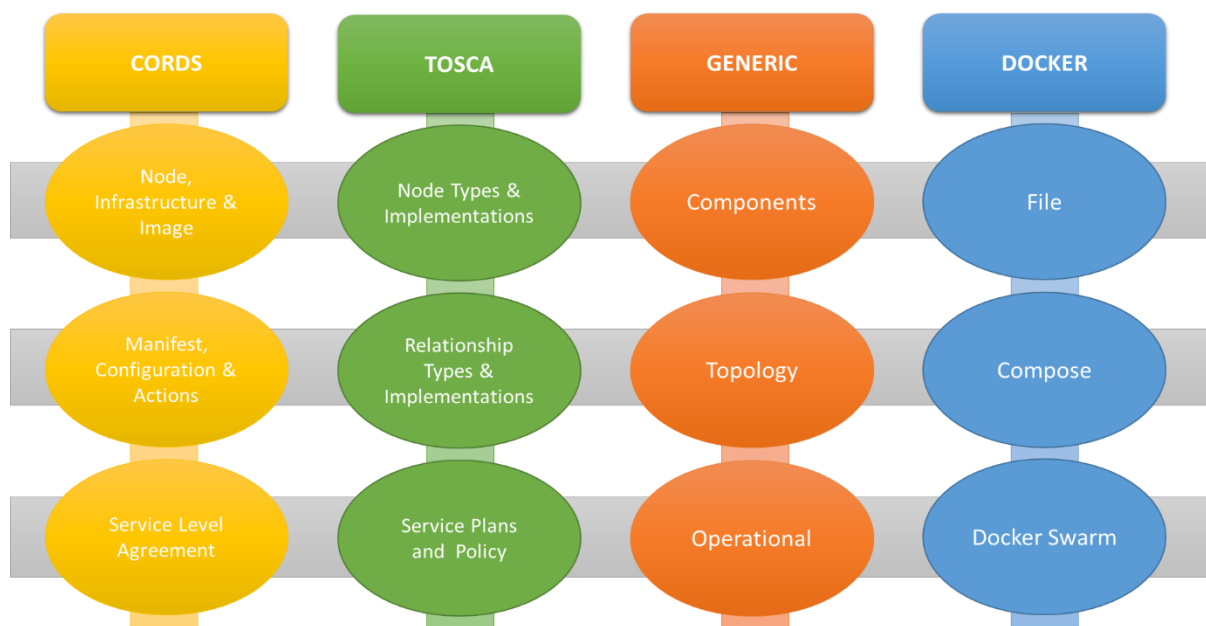
The TOSCA transformation tool is integrated with the Amenesik Cloud Engine through the Database, the OCCI interface and through the Amenesik Remote Command Service.

For further information, relating to the Amenesik Cloud Operator, please refer to the corresponding document entitled: Amenesik Cloud Operator.

TOSCA Definitions

A TOSCA definitions document is an XML document respecting the specifications described by version 1.0 of the TOSCA standard. A precise detailed description of the TOSCA standard can be found by following the link to this specification included in the reference section at the end of this document.

A TOSCA definition document, intended for conversion to a CORDS Manifest and subsequent use by the Amenesik Cloud Engine, should contain exactly one Service Template definition. In addition, the document should contain or import all secondary type and template definitions required for correct interpretation or understanding of the Service Template. The use of the TOSCA definition types will be described in the following sections of this document. In each description, only the attributes and elements which are used will be described. All other attributes and elements will currently be silently ignored. The following diagram depicts the relationship between TOSCA and CORDS with respect to a generic orchestration model and with respect to the emerging container model.



Import

This element allows a TOSCA definitions document to be included by the current TOSCA definitions document. This may be used to include or import a collection of standard TOSCA definitions required for the interpretation of a master **Service Template** definition.

Location

The string value of the *Location* attribute of the **Import** element will be used as the URI or filename of the TOSCA definitions document to be imported. The document will be submitted for processing and may provide definitions for any of the elementary types.

Service Template

The master TOSCA document should contain one single Service Template definition. This service template, and the Node Template definitions that it contains, will be processed, by the TOSCA

transformation tool, to produce the corresponding CORDS Manifest document. The actual semantics of the various Node Templates will be provided by the accompanying Node Types and their associated Node Type Implementations.

Name

The value of the *Name* attribute of the **Service Template** element will be used “as is” as the value of the *Name* of the resulting CORDS **Manifest**.

Tags

The values of the *Name* and *Value* attributes, of the **Tag** elements of the **Tags** section of the **Service Template** definition, will be concatenated to create the value of the *Description* attribute for the resulting CORDS **Manifest**. Each colon separated name/value pair of Tag attributes will be comma separated in the resulting string value.

Node Template

The **Node Template** elements, of the **Topology Template** section of the **Service Template** definition, will be processed to determine both the virtual machine and/or container nodes and the software installation and configuration operations required for the correct operation of the described cloud application. The recursively resolved value of the *Type* attribute of the *Node Template* element will be used to determine the actual nature of the node as follows:

- When the value of this *Type* attribute is found to be a direct or indirect relation of the normative type “tosca.nodes.Compute”, then the **Node Template** definition will represent a virtual machine or container and will result in a corresponding infrastructure **Node** being added to the output CORDS **Manifest**.

In this case the Capability elements of the Capabilities section of the Node Template will be used to provide the hardware and software characteristics of the virtual machine or container as follows:

- Otherwise, the **Node Template** definition will represent a software installation or configuration procedure and will result in the corresponding **Configuration** and **Release Action** expressions being added to the output CORDS **Manifest**.

In this case the **Requirement** elements of the **Requirements** section of the **Node Template** will be used to determine the actual virtual machine or container to which the software installation of the **Node Template** applies.

Node Type

The information provided by these definitions is currently used to determine the different fundamental nature of **Note Template** types.

Node Implementation

The information provided by these definitions is currently used to provide the installation and configuration instructions of the software nodes via the **Operations**, Create, Start, Stop and Delete of the Standard **Interface**.

Node Relationship Type

This element definition is currently unused but will eventually be used for the determination of inter node relationships and dependencies

Node Relationship Implementation

This element definition is currently unused but will eventually be used for the determination of inter node relationships and dependencies

Requirement Type

This element definition is currently unused but will eventually be used for the determination of inter node relationships and dependencies.

Capability Type

This element definition is currently unused but will eventually be used for the determination of inter node relationships and dependencies

Artifact Type

This element definition is currently unused but will eventually be used for the determination of the types of software delivery packages and procedures.

Policy Type

This element definition is currently unused but will eventually be used for the determination of the service instance operational policy concerning monitoring and other runtime operations.

Mapping

This section of the document shows the mapping between the TOSCA XML elements and attributes and the CORDS XML elements and attributes. For clarity the mapping to CORDS nodes will be shown apart from the CORDS configuration section.

<pre> <manifest name="load-balanced-web-application-with- database"> <node name="one" type="simple" entry="none" access="public" scope="normal" provider="any"> <infrastructure name="load-balanced-web-application-with- database:one"> <compute name="load-balanced-web- application-with-database:one" cores="1" memory="1G" architecture="x86"/> <storage name="load-balanced-web- application-with-database:one" size="10G"/> </storage> <network name="load-balanced-web- application-with-database:one" label="account"> <port name="cosacs" from="8286" to="8286" protocol="tcp" direction="inout" range="0.0.0.0/0"/> <port name="http" from="80" to="80" protocol="tcp" direction="inout" range="0.0.0.0/0"/> <port name="ssh" from="22" to="22" protocol="tcp" direction="inout" range="0.0.0.0/0"/> </network> </infrastructure> <image name="load-balanced-web-application- with-database:one" agent="none"> <system name="ubuntu 14.10"/> </image> </node> </pre>	<pre> <ServiceTemplate name="load-balanced-web-application-with- database"> <Tags/> <TopologyTemplate> <NodeTemplate name="one" type="tosca.nodes.Compute"> <Capabilities> <Capability name="host"> <Properties> <num_cpus>1</num_cpus> <disk_size>10G</disk_size> <mem_size>1G</mem_size> </Properties> </Capability> <Capability name="os"> <Properties> <architecture>x86</architecture> <type>linux</type> <distribution>ubuntu</distribution> <version>14.10</version> </Properties> </Capability> </Capabilities> </NodeTemplate> </pre>
---	--

Examples

Single Node

This example shows a single compute node with no particular specialisation.

```
<Definitions>
  <Import location="ace-tosca-defaults.xml"/>
  <ServiceTemplate name="single-compute-node">
    <Tags>
      <Tag name="Title" value="Single Compute Node"/>
      <Tag name="SubTitle" value="Demonstration of TOSCA Conversion"/>
      <Tag name="Version" value="1.0a.01"/>
      <Tag name="Author" value="Iain James Marshall"/>
      <Tag name="Date" value="24th March 2016"/>
    </Tags>
    <TopologyTemplate>
      <NodeTemplate name="VM" type="tosca.nodes.Compute">
        <Capabilities>
          <Capability name="host">
            <Properties>
              <num_cpus>1</num_cpus>
              <disk_size>10G</disk_size>
              <mem_size>1G</mem_size>
            </Properties>
          </Capability>
          <Capability name="os">
            <Properties>
              <architecture>x86</architecture>
              <type>linux</type>
              <distribution>ubuntu</distribution>
              <version>14.10</version>
            </Properties>
          </Capability>
        </Capabilities>
      </NodeTemplate>
    </TopologyTemplate>
  </ServiceTemplate>
</Definitions>
```

Web Server with Database

This example shows a compute node on which a web server and database have been installed. For clarity the TAGS section of the Service Template and the Properties of the Node Templates have been omitted.

```
<Definitions>
  <Import location="ace-tosca-defaults.xml"/>
  <ServiceTemplate name="web-server-with-database">
    <Tags/>
    <TopologyTemplate>
      <NodeTemplate name="VM" type="tosca.nodes.Compute">
        <Capabilities>
          <Capability name="host">
            <Properties/>
          </Capability>
          <Capability name="os">
            <Properties/>
          </Capability>
        </Capabilities>
      </NodeTemplate>
      <NodeTemplate name="db" type="tosca.nodes.DBMS.PostgreSQL">
        <Requirements>
          <Requirement name="VM" type="host"/>
        </Requirements>
      </NodeTemplate>
      <NodeTemplate name="ws" type="tosca.nodes.WebServer.apache">
        <Requirements>
          <Requirement name="VM" type="host"/>
        </Requirements>
      </NodeTemplate>
    </TopologyTemplate>
  </ServiceTemplate>
</Definitions>
```

References

This section of the document provides a collection of links to cloud standards documentation and Amenesik support documents.

OCCI

The following documents are available from the OGF web site:

- OCCI CORE Version 1.1:
<https://www.ogf.org/documents/GFD.183.pdf>
- OCCI INFRASTRUCTURE Version 1.1 :
<https://www.ogf.org/documents/GFD.184.pdf>
- OCCI http Version 1.1 :
<https://www.ogf.org/documents/GFD.185.pdf>

TOSCA

The following documents are available from the OASIS web site

- TOSCA Version 1.1:
<http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf>
- TOSCA Namespace:
<http://docs.oasis-open.org/tosca/ns/2011/12>

CIMI

The following documents are available from the DMTF web site

- CIMI Version 1.1 :
http://www.dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.1.pdf

CORDS

The following documents are available from the CompatibleOne community web site:

- CORDS Version 1.1 :
<http://www.compatibleone.com/community/wp-content/uploads/2014/05/CordsReferenceManualV2.15.pdf>

AMENESIK

The following documents are available from the AMENESIK web site:

- Amenesik Enterprise Cloud (AEC) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikCloud.pdf>
- Amenesik Cloud Engine (ACE) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikCloudEngine.pdf>
- Amenesik Manifest Editor (AME) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikManifestEditor.pdf>
- Amenesik Agreement Editor (ASE) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikServiceEditor.pdf>
- Amenesik Service Dashboard (ASD) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikServiceDashboard.pdf>
- Amenesik Cloud Operator (ACO) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikCloudOperator.pdf>
- Amenesik Platform Editor (APE) Version 1.1:

<http://www.amenesik.com/cloud/AmenesikPlatformEditor.pdf>

- Amenesik Platform Service (APS) Version 1.1:
<http://www.amenesik.com/cloud/AmenesikPlatformService.pdf>