

Amenesik

Amenesik Cloud

Introduction

Iain James Marshall
13/05/2016

Table des matières

Introduction	1
Overview of Cloud Computing	2
In the Beginning	2
Data Centres	2
Cloud Management Software	2
Cloud Applications	2
Cloud Application Configurations	3
CompatibleOne	4
The Project	4
Generic Cloud Model	4
Open Cloud Computing Interface	4
CompatibleOne Resource Description System	4
Amenesik Enterprise Cloud	5
Amenesik Service Dashboard	5
Amenesik Cloud Engine	5
Amenesik Manifest Editor	6
Amenesik Service Editor	7
Amenesik Cloud Operator	7
Amenesik Platform Editor	7
Amenesik Platform Service	7
References	8
OCCI	8
TOSCA	8
CIMI	8
CORDS	8
AMENESIK	8

Introduction

This document provides an introduction to the Amenesik Enterprise Cloud and an overview of the underlying principles and the accompanying toolset.

An overview of cloud computing in general is presented first followed by the current trends in cloud computing.

This is followed by a presentation of the CompatibleOne project and finally an overview of the various tools comprising the Amenesik Enterprise Cloud.

Overview of Cloud Computing

This section of this document provides an introduction to the concept of cloud computing, a vast subject based on the idea of providing computer services as an on demand utility in much the same way that electricity or water is made available.

In the Beginning

The concept was born when Amazon realised that the surplus computer hardware and other resources in their data centres, required to meet peak sales periods such as Christmas, could be made available to third party users when it was no longer needed for their own use. A new company, Amazon Web Services (AWS), was set up to manage and promote their new product line and the resulting concept of Amazon Elastic Cloud Computing (EC2).

Data Centres

Modern day computer hardware with multicore central processing units, (CPUs), and often multiple CPUs, combined with network connectable disk storage and computer virtualisation software, known as hypervisors, make it possible to redefine a single physical machine as a collection of individual virtual computers. These high performance computer hardware units, known as blades, and accompanying disk arrays and network switches are stacked up together in storage racks which are lined up into corridors inside a modern data centre. Each rack containing somewhere in the region of 8 computers with up to 64 cores each. With only 10 corridors of 10 such racks each, this would represent 800 computers with around 6400 individual cores. In reality the numbers are much higher. Data centre management software, such as Amazon AWS EC2, Windows Azure or OpenStack NOVA is responsible for managing and harnessing this array of computing, storage and network hardware such that virtual computer configurations can be created and made available on demand over the internet for use by remote clients.

Cloud Management Software

There are several major commercial actors in the cloud computing field. In addition to the major proprietary solutions (AMAZON AWS, Microsoft AZURE, Google Compute and IBM SOFTLAYER) a large number of hosting companies (RACKSPACE, OVH, REDHAT, DELL etc.) are using Open Source alternatives such as OpenStack NOVA or Open Nebula as the basis for their public cloud offerings. As would be expected the proprietary solutions of the major actors are certainly not interoperable since the objective of their business is to capture the market and customers. It could be expected that the situation would be better in the case of the Open Source solutions, however rapid and exponential and often uncoordinated growth of the various Open Source code bases has not helped to ensure this, leading to a degree of disparity between the different commercial versions of Open Stack alone.

Many large multinational companies operate their own private data centres. These private clouds, as they are known, can be based on either proprietary or open source software solutions.

Cloud Applications

Cloud Applications differ from standard hardware based applications in that, from the outset, they are designed with scalability and mobility in mind, or least they should be.

The term Scalability refers to the ability of the Application to increase or decrease the quantity of computing resources that it uses at any one moment in time. This is possible within the cloud deployment framework since on demand resources, compute, network or storage, can be added, or released, under the control or direction of the application as required by the changing circumstances encountered during the application life cycle.

The term Mobility refers to the ability of the Application, and its components, to be deployed into the most appropriate geographical region and in many cases this may involve the use of multiple alternative cloud management technologies and providers since not all commercial operators cover all regions to the same level of detail.

The combination of these two techniques, scalability and mobility, allow applications to expand and contract as required to satisfy the actual user load encountered at any one moment in time. In addition, resources can be allocated, or placed, nearer to the customers in order to alleviate network bottlenecks and improve the overall user experience.

Cloud Application Configurations

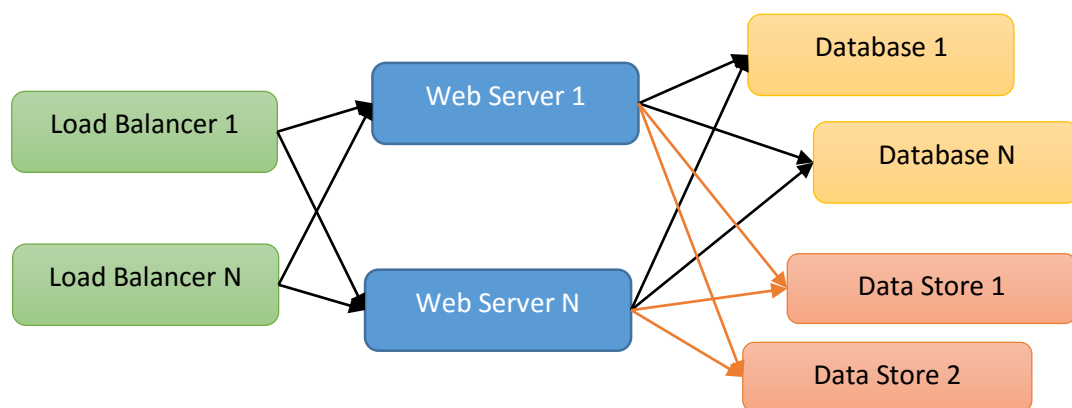
Cloud applications are invariably Web Applications, based on modern day web technology of one kind or another. This usually takes the form of an HTTP Web server receiving client requests and delivering content of both a static and dynamic nature.

The HTTP Web Server will usually be connected with a database and perhaps a data object store and when required to exhibit a scalable nature will be operating behind a load balancer.

In complex configurations, where high load is anticipated, multiple load balancers can be distributing customer requests over multiple web servers connected across an array of databases and data stores. Typically web servers would be added or removed in response to increase and decrease in overall load. Multiple database servers are required to ensure back end performance, data integrity and high availability. Multiple load balancers are recommended in order to avoid the existence of an intrinsic single point of failure.

This architecture can be applied in a consistent manner for the description of cloud applications, irrespective of the nature and purpose of the actual application managed by the web application server.

The following diagram depicts this standard cloud application scenario:



CompatibleOne

The Project

The CompatibleOne project was initially intended to provide an Open Source Cloud Management software solution such as OpenStack, since at the time OpenStack did not exist, but when the project finally got under way OpenStack was already well established and the project was immediately repositioned. The new focus of the project was to be an Open Source software providing a solution to the problem of public and private cloud disparity with an aim to alleviating the resulting vendor lock-in. In this way the CompatibleOne Open Source Cloud Brokerage Platform came into existence.

Generic Cloud Model

The CompatibleOne approach was based on a generic descriptive model for the definition, deployment and life cycle management of simple and complex computing systems in the cloud. This model, the Compatible One Resource Description System (CORDS) was developed concurrently with other work in the field of cloud computing, namely Open Cloud Computing Interface (OCCI), Topology Orchestration for Scalable Cloud Applications (TOSCA) and Cloud Infrastructure Management Interface (CIMI). The state of advancement of these three intended standards was evaluated at the time and only OCCI was seen to be sufficiently mature for any kind of adoption as the basis for the CORDS model. Neither TOSCA nor CIMI were anywhere near complete at the time and precise information was not available due to their proprietary consortium approach with little concern for open standards or inter project collaboration.

Open Cloud Computing Interface

The Open Cloud Computing Interface (OCCI) is managed by the Open Gris Forum (OGF) and as its name implies, OCCI, was originally intended to provide an Open Standards Interface for the management of Cloud Computing. To this effect OCCI provides a coherent suite of standardised descriptions not only for the primary elements required for cloud computing, namely COMPUTE, NETWORK and STORAGE, but also for their delivery over an HTTP binding. The model has since been extended to provide definitions in other areas such as monitoring, service level agreements and platform as a service. The area of primary interest to CompatibleOne was, however, the underlying OCCI CORE. This is a META model descriptive system offering a standardised approach for the description of classes of data objects and their resulting instances.

CompatibleOne Resource Description System

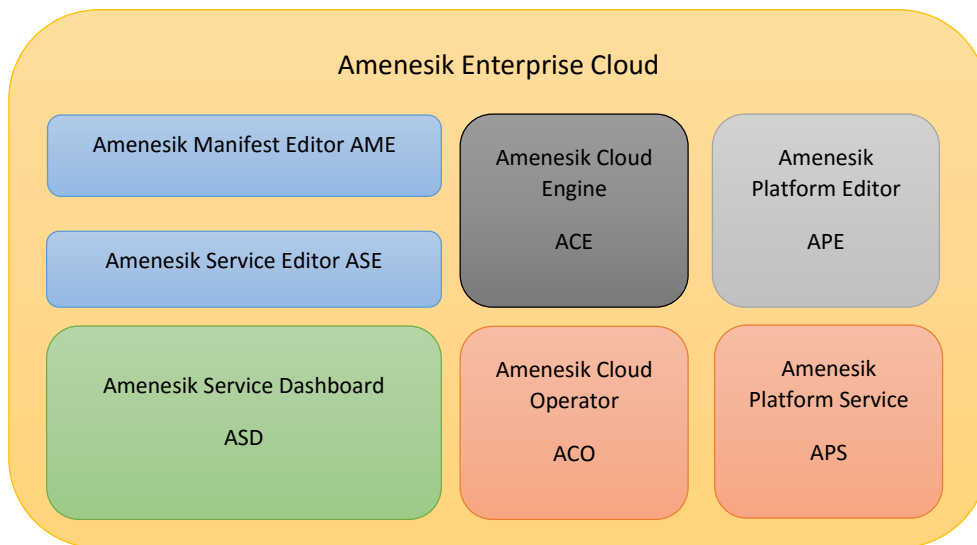
OCCI CORE provides the Open Standard Definition from which the CORDS model has been derived. In addition, the OCCI HTTP binding provides the Open Standard REST definition on which the resulting Advanced Capabilities for the CompatibleOne Resource Delivery System (ACCORDS) Platform is based.

This model provides a precise yet generic means not only for the description of cloud resources in terms of technical manifests, but also the offer of service in the form of commercial agreements and the management of deployed resources in the form of service instances and their adjacent monitoring and control systems. Each element in the CORDS model is represented by an OCCI category definition respecting the standard outlined in the OCCI CORE model. Further information to this effect can be found in the document entitled “The CORDS Model Technical Reference” resulting from the CompatibleOne project.

Amenesik Enterprise Cloud

The Amenesik Enterprise Cloud (AEC) offer is based on an industrialised version of the Cloud Brokerage software resulting from the CompatibleOne Open Source project. This version of CompatibleOne, known as the Amenesik Cloud Engine (ACE), has been re-engineered in order to be completely generated and subsequently delivered and managed by an integrated suite of platform definition, production and deployment tools.

The following diagram depicts the relationship between the various components of the Amenesik Enterprise Cloud:



Amenesik Service Dashboard

The Amenesik Service Dashboard provides a web application based user interface tool facilitating the day to day management of cloud applications deployed through the Amenesik Cloud Engine. This tool is intended for use by personnel of a non-technical background for the management of cloud application deployment operations.

This 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 Service Dashboard, please refer to the corresponding document entitled: Amenesik Service Dashboard.

Amenesik Cloud Engine

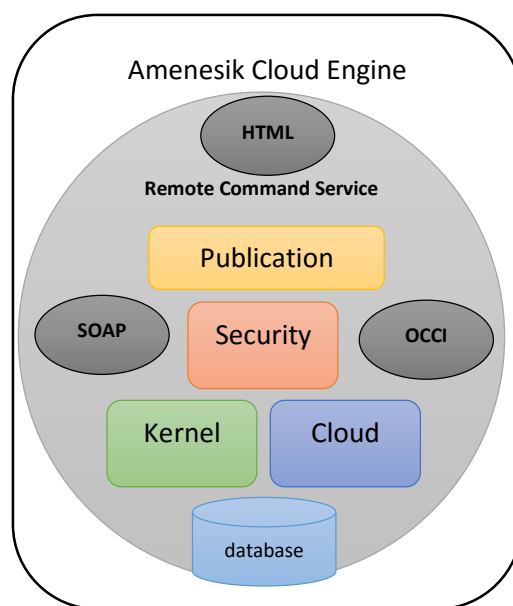
This component of the Amenesik Enterprise Cloud, based on an industrialisation of the CompatibleOne Accords Platform, is responsible for the management of all aspects of cloud application descriptions and their run time deployment. It presents the traditional OCCI interface allowing direct and authenticated access to all components of the CORDS model.

The engine has been entirely refactored for operational efficiency, using the Amenesik Platform Editor, resulting in the following four individual functional components:

1. The Publication Services. This component provides the centralised service registry through which all OCCI categories are made known and available for use by all other components of the system.
2. The Security Services. This component provides the categories required for the implementation and support of the Amenesik Security Policy.
3. The Kernel Services. This component provides the CORDS model management categories.
4. The Cloud Deployment Services. This component provides the various cloud provisioning categories for IAAS, PAAS and SAAS levels of service deployment.

The engine is accompanied by the Amenesik Remote Command service which can be accessed either through a SOAP interface and its adjacent WSDL definitions document or through the standard HTML Web Browser support interface.

The following diagram depicts the components of the Amenesik Cloud Engine:



Access to this component will be performed through the various Amenesik User Interface tools by the appropriately trained personnel. Direct access to the tool will be performed only by members of Amenesik or representatives having followed an intensive training.

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

Amenesik Manifest Editor

The Amenesik Manifest Editor provides a web application based user interface tool facilitating the tasks of creating and managing the cloud application technical description Manifest documents that are required for operation of the Amenesik Cloud Engine. This tool is intended for use by a cloud application architect with the required level of technical ability in the domain of Cloud Application Deployment Architectures.

This 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 Manifest Editor, please refer to the corresponding document entitled: Amenesik Manifest Editor.

Amenesik Service Editor

The Amenesik Service Editor provides a web application based user interface tool facilitating the tasks of creating and managing the cloud application commercial description Agreement documents that are required for operation of the Amenesik Cloud Engine. This tool is intended for use by a cloud application architect with the required level of technical understanding in the domain of Cloud Application Configuration.

This tool is integrated with the Amenesik Cloud Engine through the Database, the OCCl interface and through the Amenesik Remote Command Service.

For further information, relating to the Amenesik Service Editor, please refer to the corresponding document entitled: Amenesik Service Editor.

Amenesik Cloud Operator

The Amenesik Cloud Operator provides a web application based user interface tool facilitating the day to day configuration, management and maintenance of an Amenesik Cloud Engine by the platform operator. This tool is to be used by an authorised Amenesik Cloud Engine operator with the required level of both technical and commercial training.

This tool is integrated with the Amenesik Cloud Engine through the Database, the OCCl 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.

Amenesik Platform Editor

The Amenesik Platform Editor provides a web application based user interface tool facilitating the definition, generation and production of deployable instances of the Amenesik Cloud Engine.

This tool is integrated with the Amenesik Modelling Support tools of the Amenesik Master Cloud Engine through the Database, the OCCl interface and through the Amenesik Remote Command Service. This tool is to be used by the personnel of Amenesik for the definition of Amenesik Cloud Engine instances for use by customers through the Amenesik Platform Service.

For further information, relating to the Amenesik Platform Editor, please refer to the corresponding document entitled: Amenesik Platform Editor.

Amenesik Platform Service

The Amenesik Platform Service provides a web application based user interface tool facilitating the deployment and management of Amenesik Cloud Engine instances defined using the Amenesik Platform Editor tool.

This tool is integrated with the Amenesik Modelling Support tools of the Amenesik Master Cloud Engine through the Database, the OCCl interface and through the Amenesik Remote Command Service. This tool is to be used by customers of Amenesik for the self-service deployment of their Amenesik Cloud Engine and its associated toolset.

For further information, relating to the Amenesik Platform Service, please refer to the corresponding document entitled: Amenesik Platform Service.

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>