Chapter 6
Claudia Service Management Platform

Daniel Moran
Telefonica I+D, Spain

ABSTRACT

Claudia is an IaaS solution created by Telefonica R&D aimed at managing services as whole entities and adding advanced features such as elasticity, service level, hardware monitoring, and virtual machine autoconfiguration. Claudia also provides an abstraction layer on top of both private and public Virtualization Providers. In this chapter, the Open Source version of Claudia will be installed, and it will be used to show how a load-balanced Web service can be deployed and managed in the system.

ARCHITECTURE

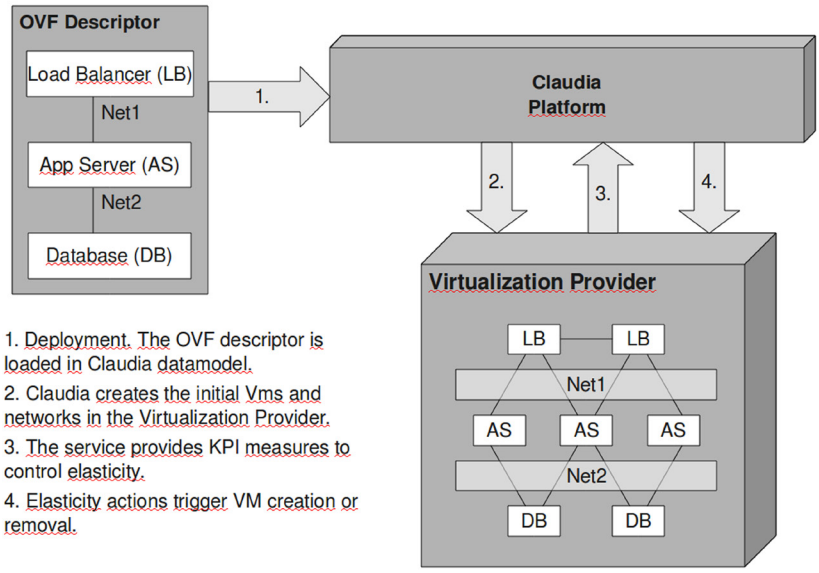
Overview

Claudia aims to provide a way to manage virtualized services as a whole. Its potential users are service providers with applications relying on complex stacks of software, whose installation processes and administration may be highly labor-intensive. Infrastructure as a Service platforms based on virtual machines already offer a way to provision needed hardware faster than with physical machines, but burden the service providers with the administration of each separate Virtual Machine. Claudia offers an IaaS solution that lets these providers define their service architecture and images (using a standard DMTF OVF descriptor with some extensions) and then deploy and manage them as a whole entity, inside a virtualization provider (See Figure 1). The service definition includes the hardware requirements needed for each service element as well as for shared service networks, in order to assign the hardware requirements needed for the full stack. This approach quickens deployments and updates of the service and gives it the ability to modify the assigned resources at runtime in an unattended manner, based on service level monitoring information (through the creation and removal of VMs).

Some terms must be defined before proceeding with the architecture description. A service is a set of interconnected virtual machines working together to provide a set of functionalities. A service provider who owns at least one service in the Platform is called a customer. Services are described as collections of VM classes called Virtual Execution Environments (VEE). Each

DOI: 10.4018/978-1-4666-0098-0.ch006

Figure 1. An overview of the service deployment process in Claudia



VEE is defined with the following elements: a collection of hardware requirements, a disk image, a set of VEE properties (including the deployment order, and the initial, minimum and maximum number of instantiated VMs) and a list of network connections to other components. Once deployed, each VEE will be instantiated as one or more Virtual Machines (VM); elasticity rules may be defined to modify the number of instances at runtime, based on service level monitoring. Customers may define Key Performance Indicators (KPI), to offer meaningful measures to manage the elasticity A real value can be used as a KPI; values are sent from probes inside the VEEs to the REST Monitoring entry point provided by the platform. The whole service structure is defined in the OVF descriptor of the service.

Once VMs are instantiated from the VEE templates (which lack runtime information) some reconfiguration mechanism must be provided in order to configure the instantiated machines at runtime. This mechanism is based on the OVF environment file. At runtime, when a new VM is going to be created, Claudia collects all the environment information needed to setup the VM (e.g., network address, addresses of other deployed machines of the same service or the monitoring address) and serializes it in a OVF environment file (a standard XML file defined by the DMTF containing all this information as a list of properties). This file is loaded on an additional disk device attached to the VM, from which the VM can load the required data. The retrieval and reconfiguration mechanism is up to the service. A working example with bash scripts can be found in the use case section.

Internal Structure

Claudia architecture is composed of three main components: Clotho (the Service Life Cycle Manager), the TCloud Server, and a collection of drivers corresponding to the different cloud providers and for Clotho itself.

The central Service Manager, called Clotho, manages the service deployment and life-cycle.