2011 IEEE 4th International Conference on Cloud Computing

The KOALA Cloud Manager
Cloud Service Management the Easy Way

Christian Baun, Marcel Kunze, Viktor Mauch
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1
D-76344 Eggenstein-Leopoldshafen
Germany
christian.baun,marcel.kunze,viktor.mauch@kit.edu

***Abstract*—The key advantages of cloud computing are flexibility, scalability (elasticity) and usability and these features originate in the combination of virtualization technologies with web services [1]. However, in contrast to the elasticity and flexibility of cloud services the traditional management methods and tools seem inappropriate because they usually require local software installation with continuous updates and patches. Furthermore, the solutions are often proprietary and only conform to the cloud service offerings of specific service providers, making it difficult to work with cloud services of different providers. A vendor agnostic generic cloud based software service would bear many advantages in this area. This paper describes the design of a better management solution – the KOALA cloud management service – for cloud services and its implementation.**

***Keywords*-Cloud Computing; Service Management**

The Amazon Web Services (AWS) are a popular collection of public cloud service offerings. Due to their widespread use their API may be considered a de-facto standard for cloud services. Amongst others, the AWS include the Elastic Compute Cloud (EC2) to operate scalable virtual server instances, the Simple Storage Service (S3) for web objects and the persistent block level storage service Elastic Block Store (EBS). Implementations that adhere to the AWS API benefit from the growing ecosystem of compatible management tools and services. Various open source infrastructure solutions like Eucalyptus [2], Nimbus [3] and OpenNebula [4] exist and most of them implement at least a subset of the EC2, S3 and EBS APIs.

I. Management of Cloud Services

The existing management tools for cloud infrastructure and storage services can be classified as follows:

- Command-line tools like the Amazon API Tools, the Eucalyptus Euca2ools, Google GSutil and s3cmd.
- Locally installed management applications with a graphical user interface like EC2Dream and Cyberduck.
- Firefox browser extensions like ElasticFox and S3Fox.
- Software services such as the AWS Management Console, Google Storage Manager and Ylastic.

The command-line tools support both, public and private cloud infrastructure services, but they require local installation and may lack ease of use as there is no graphical user interface. Locally installed applications with a graphical user interface in general provide a better level of usability but require a local installation too. The browser extensions only work with the Firefox browser. Furthermore, the customers have to install and maintain the management tool on their local computer. The customers of software services have the freedom of choice to use any combination of operating system and browser. However, the AWS Management Console is in line with just the AWS cloud service offerings. It is not foreseen to configure it in a way to manage other cloud infrastructures. The same holds for the Google Storage Manager that cannot be configured to work with any other storage services. Ylastic offers support for most AWS cloud services and Eucalyptus infrastructures but as the access keys are stored with the provider, the customer always has to trust the provider of the management tool regarding privacy and security.

While many customers of cloud services own smartphones or tablet computers, the current management tools for cloud services more or less lack compatibility with such devices. The browser extensions are often incompatible with mobile devices because they do not run the Firefox browser and the use of command-line tools is mostly impracticable. Drawbacks of native applications for these devices is their limited storage capacity, the need to perform regular updates and the development effort to support more than a single platform. In contrast to browser extensions and command-line tools the software services work with any browser and therefore also with mobile devices.

II. Management tool requirements

The drawbacks of tools like browser extensions, command-line tools and applications with a graphical user interface that require a local installation indicate that an online tool is best suited to work with multiple cloud services. This should be delivered as a software service that can be used

978-0-7695-4460-1/11 $26.00 © 2011 IEEE
DOI 10.1109/CLOUD.2011.64

with all kinds of devices and browsers. The service could be implemented on a scalable platform service such as Google App Engine. It is also quite easy to optimize a software service for mobile devices that lack physical keyboards and have a limited screen size.

If the software service is operated on a public cloud platform, the customers have no need to worry about scalability and availability. In scenarios where security or privacy concerns exist, it should be possible to operate the management tool on a private cloud platform, too, avoiding to store the access credentials off-premise. In principle all secret keys should be encrypted due to security concerns as they are stored.

The fact that a software service is not functional without a internet connection does not resemble any problem as working with cloud services requires in most use cases to be connected with the internet anyway.

III. KOALA Cloud Manager

KOALA is a management tool for cloud services realized as a software service. It offers all functionality needed to interact with cloud services compatible to the AWS API. Public cloud infrastructure services such as AWS and services based on Eucalyptus, Nimbus, OpenNebula are currently supported. The software also helps to manage S3 and EBS compliant storage resources like Google Storage and Host Europe Cloud Storage and it interacts with the Elastic Load Balancing (ELB) service [5].

A. Architecture

The KOALA architecture has been built on top of the Google App Engine platform services. The design offers several interesting choices to deploy the software as a service using a mashup of public and private clouds. The tool may be installed and run either on the public Google cloud platform or on private platform services like AppScale [6] and typhoonAE. If KOALA operates on a private cloud platform it is possible to control the cloud infrastructure with a management application running inside the same cloud. This helps customers with specific requirements regarding security and privacy. As an option, KOALA may be deployed on a private platform service, running on top of a public cloud AWS instance. Furthermore, as typhoonAE can be build and run inside Macintosh and Linux environments, KOALA may execute as well without any public or private cloud platform and virtualization solution like Xen or VMware.

Users of smartphones are able to use KOALA, too, as the whole graphical user interface has been implemented in HTML without JavaScript or proprietary technologies like Adobe Flash. KOALA includes a customized version for mobile devices that provides a user interface optimized for the display resolution of smartphones and the usage of touch screens.

B. Platform-specific Issues

If KOALA executes on the basis of the Google App Engine the restrictions of this platform service need to be considered. Applications running inside the App Engine exclusively communicate via URL Fetch, XMPP, E-Mail and the port numbers 80, 443, 4443, 8080-8089, 8188, 8444 and 8990. This does not resemble any problem for public cloud services like the AWS or Google Storage as these services rely on port 80 and 443 only. However, it is a challenge for the operation of private cloud infrastructures like Eucalyptus, Nimbus and OpenNebula. The default port numbers supported by Eucalyptus, Nimbus and OpenNebula are 8773, 8442 and 4567. Therefore the administrator of a private cloud infrastructure has to route the infrastructure access port to one of the supported ports.

If KOALA operates on a private cloud platform service like AppScale or typhoonAE, the customers are free to use any port number allowed by the hosting system.

IV. Conclusion and Future Work

The emerging market of cloud services currently lacks generic management solutions with good usability. In this paper we present the design and implementation of KOALA, an open source management tool to fill this gap. KOALA is a web-based application that supports services that are compatible to EC2, S3, EBS and ELB. The software itself may be installed and operated within public and private clouds. KOALA is not just a new tool or service; it is a new approach to work in a seamless and transparent way with cloud infrastructures and storage services of various public cloud providers as well as private cloud solutions.

References

[1] C. Baun, M. Kunze, J. Nimis, and S. Tai, *Cloud Computing: Web-basierte dynamische IT-Services*. Springer, 2011.

[2] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems," in *UCSB Computer Science Technical Report Number 2008-10*, 2008.

[3] P. Marshall, K. Keahey, and T. Freeman, "Elastic Site Using Clouds to Elastically Extend Site Resources," April 2010.

[4] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, no. 5, 2009.

[5] C. Baun and M. Kunze, "The KOALA Cloud Management Service – A Modern Approach for Cloud Infrastructure Management," in *CloudCP: International Workshop on Cloud Computing Platforms*, April 2011.

[6] C. Bunch, N. Chohan, C. Krintz, J. Chohan, J. Kupferman, P. Lakhina, Y. Li, and Y. Nomura, "An Evaluation of Distributed Datastores Using the AppScale Cloud Platform," in *IEEE Cloud10: International Conference on Cloud Computing*, July 2010.