University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Piotr Styczyński
Student no. 386038

Michał Balcerzak
Student no. 385130

Michał Ołtarzewski
Student no. 382783

Gor Safaryan
Student no. 381501

Omigost - AWS Cost Optimization Tool

Bachelor's thesis
in COMPUTER SCIENCE

Supervisor:
dr Janina Zofia Mincer-Daszkiewicz
Instytut Informatyki

May 2018

Supervisor's statement

Hereby I confirm that the presented thesis was prepared under my supervision and that it fulfils the requirements for the degree of Bachelor of Computer Science.

Date Supervisor's signature

Authors' statements

Hereby I declare that the presented thesis was prepared by me and none of its contents was obtained by means that are against the law.

The thesis has never before been a subject of any procedure of obtaining an academic degree.

Moreover, I declare that the present version of the thesis is identical to the attached electronic version.

Date Authors' signatures

Abstract

Authors describe the design and implementation process, in-depth system and code architecture as well as used communication protocols, storing methods and other internals of the AWS Cost Optimization System.

Keywords

cloud computing, Amazon Web Services, cost optimization, cost management, machine termination

Thesis domain (Socrates-Erasmus subject area codes)

11.3 Informatics, Computer Science

Subject classification

D. Software

Tytuł pracy w języku polskim

Omigost - Narzędzie Optymalizacji Kosztów AWS

Contents

Chapter 1

Introduction

1.1. Overview

1.1.1. What is cloud computing

Cloud computing became one of the most important paradigm shifts in the area of real world software engineering. It has reshaped the whole process of how applications are developed and deployed, and reduced the amount of upfront investment required to start an internet business. While commercial cloud computing services were first offered in 2006 by Amazon Inc, the original idea and preliminary implementation traces back to Multics OS developed by MIT, GE and Bell Labs. However the idea of time-sharing systems that was the ancestor of further cloud concept was widespread in 60ies [**Markus**].

1.1.2. The term itself

The term "Cloud computing" can refer to every layer of application stack: hardware, hosting platform, software and even to a single function. Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS) [**Armbrust**]. Namely, it is a shared pool of computer resources such as computing capacity, transient and persistent memory, which can be acquired or released on demand. The undisputed power of cloud computing constitutes in its elasticity and granularity: i.e. it allows users to ask for hundreds of computers for only 5 minute usage which are shipped during several minutes. Such services are usually offered over remote network connection and users are billed for the portion of the resources they have used. Depending on the cloud infrastructure type the payment models can be different [**Laatikainen**], but the common spendings are associated with data storage, data transfer, and computing timeshare.

1.1.3. Cloud resources

Cloud computing has introduced 3 new aspects which are radically different from a traditional computing paradigm in terms of hardware infrastructure. The availability of practically infinite on demand computing resources, which allows users to deploy applications without any kind of resource planning. The elimination of massive initial financial commitment, therefore allowing small companies to increase their hardware usage proportional to their

needs. The ability to granularly allocate computing resources on any kind of timeframe (from minutes up to days etc.), as well as the ability to release them as there is no more need.

Depending on the service offered by the provider we differentiate 3 main models of cloud computing:

1. Infrastructure as a service (IAAS).
2. Platform as a service (PAAS).
3. Software as a service(SAAS).

1.1.4. IAAS – Infrastructure as a service

In such form of service, the provider allocates an instance of virtual machine for the client and ensures that minimal building blocks for the IT infrastructure are present: network, storage, computing capacity, load balancer, VLAN etc. Usually providers of such services run pools of hypervisors such as Xen, VMware, QEMU etc which host and manage those virtual machines. From the user perspective it usually looks like a command line interface through which user has full control over the allocated virtual machine. Some of the well known services are Amazon EC2, Google Compute Engine, Microsoft Azure IAAS.

1.1.5. PAAS – Platform as a service

Platform as a service alleviates the need for the developers to manage the operating system and provides programming language specific execution environment as well as the underlying structure (hardware, network, storage etc). This adds another layer of convenience over IAAS making the deployment and the development of applications much more fluent process. While users benefit from the automatic software maintenance, OS security patches etc, they also lose full control over the virtual machine instances. AWS Elastic Beanstalk, Heroku and Google App Engine are some of the most popular PAAS services.

1.1.6. SAAS – Software as a service

This form of service takes control of every layer of application and the user has nothing to do with the underlying infrastructure. This is by all means one of the most popular form of cloud computing as it might not require any kind of technical background to be used. Examples of SAAS services are Google Cloud Vision, Google Docs, Microsoft Office 365.

Chapter 2

Problem statement

2.1. Motivation

As there are plenty of various billing models for cloud services [**GLaatikainen**] the effective management of them became a tough problem. The ease of resource allocation led to situation when tracking tiniest details of billings is an unaffordable challenge.

Many of bussiness are utilizing dedicated teams for cost management or use specialized tools i.e. Cloudability [**Cloudability**], Apptio [**Apptio**], Snow Software [**SnowSoftware**] and many more.

The tooling that exists is targetting wideworld-scale companies that are able to require expensive licenses and hire cost-optimization teams. The software as it is in case of Cloudability is complex for average user and do not provide easy way to incorporate custom business flows into the tool. Amazon as one on the leading cloud providers offers tools for exploration of expenses including public APIs [**AWSCostManagement**], but the tools are rather simple and do not satisfy all the needs of potential clients.

The common case that is unresolved is the distribution of research and development resources. We observed that there exists no tool that would support request for resources of individual worker with regards to custom management propagations as specified by client bussiness model. Cloudability [**CloudabilityAlerts**] offers simple alerts, but they lack Slack support and beforementioned propagation abilities.

As there exist an obvious gap in the market, our software solution - Omigost - tries to provide these unique features to the client:

1. Easy cloud management without expensive licenses or complex knowledge.
2. Ability to provide intuitive interface for individual workers to request resources.
3. Painless integration with existing business flows.

2.2. Overview of exisiting solutions

Businesses that choose to rely upon cloud services often reach a point where resources they're using up gradually become less and less manageable. As the problem is well known to the cloud market, both Amazon and other third-party companies made attempts to fulfill these needs by creating custom software fitting certain roles in optimising AWS expenses that include:

1. configuring budget limits and alerting users when it is exceeded,

2. instance alerting management,

3. cost analytics.

Some of the most prominent tools currently available on the market that improve resource management experience for AWS cloud are described in the following sections.

2.2.1. AWS Budgets

AWS Budgets is a part of Amazon Web Services that allows to set limits of a certain types that apply to a chosen period of time. When a limit is either exceeded, close to be exceeded or is forecasted to exceed the configured threshold before the end of that period, the administrator of that account is notified by email. Types of resources one can put this kind of a budget on include:

1. Money spent in total or on a certain type of machines.

2. Utilisation of selected services.

3. Utilisation or coverage of reserved instances.

[**AWSDocs**]

2.2.2. AWS Cost Explorer and AWS Cost Management

AWS Cost Explorer enables access to all budget data. User can define and generate custom reports in a form of a data chart spanning a selected time interval with chosen time granularity of the samples. AWS Cost Explorer is also a basis for AWS Cost Management, which is basically a set of predefined reports that form an easily accessible dashboard.

2.2.3. Cloudability

Atlassian's Cloudability delivers a budget system functionality analogous to AWS Budgets along with tools for predicting future spendings and <STH ELSE TODO>. In comparison to Amazon's native tools Cloudability also allows multiaccount management, saving effort of having to set up budgets separately in every owned account.

2.2.4. Stax.io

TODO

2.2.5. Apptio

Apptio provides a set of tools for both analysis of expenses and planning future ones. It is also possible to organize resources into groups to make reports even clearer.

2.2.6. SnowSoftware

TODO

Bibliography

[Apptio] *https://www.apptio.com/products.*

[Armbrust] 2009 M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz et al. *Above the Clouds: A Berkeley View of Cloud Computing.* Technical Report No. UCB/EECS-2009-28. *http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.*

[AWSCostManagement] *https://docs.aws.amazon.com/aws-cost-management/latest/APIReference/Welcome.html.*

[AWSDocs] Amazon, *Amazon AWS documentation.*

[Cloudability] *https://www.cloudability.com/product/transform/.*

[CloudabilityAlerts] *https://blog.cloudability.com/creating-budget-alerts-by-tag-with-cloudability/.*

[Laatikainen] 2013 G. Laatikainen, A. Ojala, O. Mazhelis *Cloud Services Pricing Models.*

[Markus] 2011 M. Böhm, S. Leimeisier, C. Riedl, H. Krcmar *Cloud Computing and Computing Evolution* Technische Universität München (TUM), Germany.

[SnowSoftware] *https://go.snowsoftware.com/.*