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

AWS Cost Optimization Tool

Bachelor's thesis
in COMPUTER SCIENCE

Supervisor:
dr Janina Zofia Mincer-Daszkiewicz
Instytut Informatyki

February 2019

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

AWS, Amazon Web Services, cloud computing, cost optimization, cost management

Thesis domain (Socrates-Erasmus subject area codes)

11.3 Informatics, Computer Science

Subject classification

D. Software

Tytuł pracy w języku polskim

Narzędzie do Optymalizacji Kosztów AWS

Contents

1. Introduction
    1.1 Overview
    1.2 Aim of the thesis
    1.3 Structure of the thesis
    1.4 Contribution of each author
2. Problem statement
    2.1 Motivation
    2.2 Overview of exisiting solutions
    2.3 Our solution
3. Tool for AWS cost optimization
    3.1 Technology stack
    3.2 Overall Architecture
    3.3 Data models
    3.4 Views and design
    3.5 API
    3.6 Communication integration
    3.7 Configuration
4. Summary

A Deployment and integration guide

Chapter 1

Introduction

1.1. Overview

Cloud computing has became recently 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 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**].

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.

Nowadays the industry increasingly relies on cloud technologies. More and more companies start or move their products to cloud environment. Unfortunately it comes with additional costs imposed by cloud providers. In order to make business profitable companies try to reduce it as much as possible. Despite different solutions like employing specific "cloud cost optimization" team, more and more firms decide to benefit from dedicated software, which is supposed to help manage and optimize their cloud usage. Amount of such tools do not make it easy to chose the right one.

There are some notable solutions of AWS cloud optimization problem – *AWS Cost Explorer and AWS Cost Management*, *Cloudability*, *Apptio* and others widely used nowadays. In spite of that fact there is still place for new tools targeting omitted types of clients or wrapping and bundling the greatest features from exisiting ones.

1.2. Aim of the thesis

The primary objective of the thesis is to review exisiting tools used in cloud cost optimization and create our own new solution. We are focusing on Amazon Web Services platform maintained by Amazon as it is one of the most commonly used. In our tool called *Omigost* we will try to target small/middle sized companies creating simple and easy to use software with flexible configuration options.

1.3. Structure of the thesis

The thesis is structured as follows. In Chapter 2 we describe the problem of cloud cost optimization with additional description of selected existing solutions. Then, in Chapter 3 we present in detail our solution. Mentioned is among others whole system architecture, API and configuration. Finally, in the Chapter 4 we sum up the whole thesis. At the end in Appendix A we describe how to introduce our solution in a company.

1.4. Contribution of each author

It is important to mention that each author worked to some extent on every part of the thesis. However authors contributed mainly to the following parts:

- Michał Ołtarzewski
  - Software development process management
  - Design of backend architecture
  - Backend part implementation
- Michał Balcerzak
  - Research of existing solutions
  - Design of backend architecture
  - Backend part implementation
- Piotr Styczyński
  - Design and prototype of visual part of tool
  - Frontend part implementation
- Gor Safaryan
  - Research of AWS APIs and SDKs
  - Design of backend architecture
  - Backend part implementation

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.

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.

There exists an obvious gap in the market, our solution - Omigost will try to cover. Having one versatile tool removes need for using few detached pieces of software. Also intuitive interface and different types of notifications increases spendings clarity and helps in decisions connected to cutting costs. As a result it will allow companies to focus more on providing value to their clients in the same time saving more money. Problem we would like to solve is lack of the tool that has simultaneously following features:

- Free and Open Source
- Easy cloud management without complex knowledge
- Intuitive interface for individual workers to request resources
- Notifications for cost surpassing and redundant resources
- Highly configurable and flexible
- Integration with communication via Email and Slack

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 optimizing 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. Every description is supposed to show there is still room for a new tool improving cost optimization possibilities.

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/.*