

*Cloud Networking and Service Provisioning. ENCS 691K. Fall 2015*

*Dr. Roch H. Glitho*

*By*

*Ali Sangari (Student ID: 26816304)*

In this report, I have described the long-term project for Cloud Networks and Service Provisioning course. To do this, an overview of the developed software has been provided with screenshots of the final work, source code in zip format and also a deployable war file. Then I have discussed the cloud services and service providers that I have provisioned by mentioning the challenges that I faced, and a brief comparison between these services.

TABLE OF CONTENTS

[1 Project Overview 3](#_Toc438020284)

[1.1 User Registeration 3](#_Toc438020285)

[1.2 User login 3](#_Toc438020286)

[1.3 User Account Termination 3](#_Toc438020287)

[1.4 Addition of New Products 3](#_Toc438020288)

[1.5 Placing a Product in Auction 3](#_Toc438020289)

[1.6 Bidding on Products that are placed in auction 4](#_Toc438020290)

[2 Provisioned Cloud Services and Service Provides 4](#_Toc438020291)

[2.1 Amazun Web Services 4](#_Toc438020292)

[2.2 Microsoft Azure 4](#_Toc438020293)

[3 Challenges in Setting up Amazon EC2 5](#_Toc438020294)

[3.1 Setting up Security 5](#_Toc438020295)

[3.2 Remote Connection to VM 5](#_Toc438020296)

[3.3 Installation of Required Application 5](#_Toc438020297)

[3.4 Firewall Configuration 5](#_Toc438020298)

[4 Challenges in Setting up Microsoft Azure 5](#_Toc438020299)

[5 Auction Service 6](#_Toc438020300)

[6 Object Relationships 6](#_Toc438020301)

[7 Source Code 7](#_Toc438020302)

[8 Deployable Project file 8](#_Toc438020303)

[Appendix 1. 9](#_Toc438020304)

[Application Screenshots 9](#_Toc438020305)

# Project Overview

Aim of this project was to develop an auction system to be deployed on a cloud platform.

Required functionalities of the auction system are as follows,

## User Registeration

Any new user must b able to register and use the service. A user record includes *Name*, *username*, and *password*.

## User login

To be able to use the system a visitor must have a valid record in the system and also be logged in to the system.

## User Account Termination

A user must be able to terminate his/her account given that he/she is not currently engaged in an auction. Engaging in an auction is defined as either having a product in the action that other user are bidding for or currently having the highest bid on at least one product.

## Addition of New Products

Once a user is logged in, he/she must be able to add new products to his/her profile.Fields that are included in a Product object include, *Name*, and *Price*.

## Placing a Product in Auction

At any time after a product has been added to the user profile it can be placed in auction, given that it isn’t already in auction.

A product remains in auction for a priod of x minutes (currently configured for on minute),to receive bids. At the end of one minute the product will be sold to the highest bidder, hence will be removed from its’s original owners’ account to the new owner.

In the event that a product does not receive any bids during the allocated time slot, it will be renewed for another block of time, until it is sold.

## Bidding on Products that are placed in auction

Once a product is in auction, it can receive bids from any user who is logged in to the system. Every bid automatically raises the price by 5 units. The initial bid will match the product’s base price to form the highest current bid.

# Provisioned Cloud Services and Service Provides

In this project two cloud solution providers that I provisioned and deploy the application on were Amazon Web Services (AWS) and Microsoft Azure.

## Amazun Web Services

Amazon provides a FreeTier service which includes all of its services free for one year.

These services includes Elastic Computing 2 (EC2) which is a Virtual Machine (VM) with different Operating Systems to choose from. Other services include Simple Storage System (S3), Amazon DynamoDB, Amazon IoT, RDS, etc.

During this project, I tested a few of the above services, but the main service that was used to deploy and run the code in the end was Amazon EC2.

## Microsoft Azure

Microsoft Azure provides free service for users of DreamSpark. DreamSpark is service or account that Microsoft has built for students many Mocrosoft products and services during their studies for free. The management console of Azure is completely different from that of AWS. It looks more appealing and less overwhelming. All services are pre-pakaged, menu driven, and there is lesser amount of configuration required to run a service. On the other handazure provides less freebies for development outside of Microsoft’s development eco-system.

# Challenges in Setting up Amazon EC2

The following are some of the challenges during the set up of the system to be able to run my project.

## Setting up Security

Figuring out how to configure security to be able to access my application remotely (through a browser) took the most time. I realize many people had a similar issue the first time they tried to use EC2. The process is simple enough, but jut not very obvius to the first time user.

## Remote Connection to VM

The OS image that I installed on the VM was Windows Server so my remote connection to VM was thrugh Remote Desktop. Amazon generates a key file that helps discover the password for the administrator account.

## Installation of Required Application

After connecting to the VM instance, I needed to configure the system to run my application. Since my code was written in Java, I installed and configured Java 1.8 and Tomcat 9. No surprises here!

## Firewall Configuration

A bit of Firewall configuration was needed to allow incomming and outgoing requests and responses on port 8080 (default port set on tomcat) that I wanted to use to access the application.

# Challenges in Setting up Microsoft Azure

The free service has many limitations compared to Amazon Web Services. These limitations include application containers/servers, databases (SQL and NoSQL). Understandabily, Azure, based on DreamSpark, imposes fewer limitations on Microsoft Ecosystem users. Tomcat, a NoSQL database, and many other services were not considered part of DreamSpark, therefore they would become paid service. For every paid service Microsoft provides one month free trial but it is one twelvth of what Amazon offers.

# Auction Service

To develop the project a RESTful approach was chosen.

* It was developed in Java using Jersey REST framework.
* The front-end was developed HTML and JavaScript / jQuery for development and CSS for styling the UI.
* JSON was used to communicate data between front-end and back-end of the application.

One benefit of this approach is that front-end and back-end became completely decoupled. This means UI can freely evolve without anydependency on the server code. It can even be hosted on separate server than the one running the back-end code.

Each functionality mentioned in Project Overview was developed as a service and exposed via HTTP POST, PUT, GET, and DELETE.

# Object Relationships

AuctionSystem is made up of the following objects,

* *User* {name, username, password}
* *Product* {id, name, price, username, inAuction, basePrice, highestBid, highestBidder}
* *Auction* {productId}

And a special object *AuctionTimer* that is started as soon as product is placed in auction and then notifies the auction as soon as the set auction time block ends.

The following diagram shows the relationship between above objects.

|  |
| --- |
| Belongs to  Times products  In auction  Contains |
| ***Diagram 1.*** *Object relationship* |

# Source Code

The following .zip file can be extracted and imported into eclipse. (requires java 1.7, and Maven plug-in for eclipse)



# Deployable Project file

The following .war file can be used to deploy the application in tomcat. (requires java 1.7 and tomcat 7,8, or 9)



# Appendix 1.

## Application Screenshots

|  |
| --- |
|  |
| ***Scrshot 1.*** *In the initial state, application allows users to Register, Login and view items that are in auction. Other parts of the application requires users to be logged in to the system.* |

|  |
| --- |
|  |
| ***Scrshot 2.*** *After user registers, he/she is automatically logged in to the system and redirected to My Items page. Some pages are only visible one the user has logged in.* |

|  |
| --- |
|  |
| ***Scrshot 3.*** *In My Items page, user can create new products. Also he/she can place any of their existing items in auction. As soon as an item enters the auction, a timer is started for one minute. After the one minute, if there are any bids on the product, that item is removed from auction and its ownership is transferred to the user with the highest bid. Otherwise, it stays in auction for another minute before the same check-and-action is repeated.* |

|  |
| --- |
|  |
| ***Scrshot 4.*** *When a user clicks on “Click to bid”, the current bid is increased by five units (dollars). Timer and bid increment can be modified in Constants.java.* |