

ECE1779: Introduction to Cloud Computing

Winter 2013

Home
Course Information Sheet
Announcements
Projects
Calendar and Lecture Notes
Video Tutorials

Assignment 1

Amazon Web Services

Due Date: March 8 (extended March 15)

Objective

This assignment will expose you to the following AWS technologies: EC2, S3, CloudWatch, and Load Balancing.

Description

Your task is to develop a web application that can resize its worker pool based on user demand. Your application should provide two web-based interfaces: **userUI** and **managerUI**.

userUI should let users perform the following tasks:

1. Upload a new image. Your application should store the new image, as well as 3 pre-computed transformations on S3. Use the popular ImageMagick tool to transform the image. Both ImageMagick and im4java, a library that provides a pure-java interface to ImageMagick, are pre-installed in the provided AMI.
2. View a page with thumbnails of all their images.
3. View an specific image and all its transformed versions.

managerUI should support the following functionality

1. List workers, and their CPU utilization
2. An option for manually growing the worker pool.
3. An option for manually shrinking the worker pool.
4. An option for configuring the auto-scaling policy by setting the following parameters:
 - CPU threshold for growing the worker pool
 - CPU threshold for shrinking the worker pool
 - Ratio by which to expand the worker pool (e.g., a ratio of 2 doubles the number of workers).
 - Ratio by which to shrink the worker pool (e.g., a ratio of 4 shuts down 75% of the current workers).

Requirements

1. All images should be store in S3
2. Store information about user account and the images owned by a user (and their transformations) in the provided database. For simplicity, this database consists of two tables with the following structure:

users		
Name	Type	Description
id	int	User ID (autoincrement)
login	varchar(16)	User login
password	varchar(16)	User password

images		
Name	Type	Description
id	int	Image ID (autoincrement)
userId	int	User ID
key1	varchar(255)	S3 key of original image
key2	varchar(255)	S3 key of 1st transformation
key3	varchar(255)	S3 key of 2nd transformation
key4	varchar(255)	S3 key of 3rd transformation

This database is purposely simple to let you concentrate on the AWS-related parts of the assignment. For example, the schema limits the implementation to a naive user authentication mechanism that stores password in clear text. Such an approach is unsafe and would not be used in a real deployment.

3. Provide a welcome page that lets returning users log into their account, and enables new users to create a new account.
4. Your application should monitor the load on its pool of workers using the AWS CloudWatch API. When the load exceeds (or drops bellow) the configurable threshold, your application should use the AWS EC2 API to resize its worker pool. Do **NOT** use AWS Auto Scaling feature for this assignment. **Important:** in order to use AWS CloudWatch, you need to enable monitoring of your EC2 instances.
5. Use only EC2 m1.small instances for your worker pool
6. Use EC2 Load Balancing to distribute request among your workers. Your application should use the AWS Elastic Load Balancing API to add and remove workers from the load balancing pool as you reconfigure your worker pool in response to load.
7. To simplify testing of your application under load, the upload function of the userUI should conform to the following specification:

```
action name = /ece1779/servlet/FileUpload
enctype = multipart/form-data
```

```
method = post
field1 name = userID type = string
field2 name = theFile type = file
```

A sample form that conform to this specification is available [here](#).

A load generator that conforms to this interfaces will be made available to you in the coming week.

8. Password protect the functionality of the managerUI. Store the user and password as parameters in web.xml.

Deliverables

I will test your application using your AWS account.

To assist me in this task, submit documentation for your applications including the following details:

- General application architecture
- Instructions for starting the application, including the AMI(s) that I should run.
- Any credentials I will need to use.
- Anything else you think I will need to understand your code and how it works.

EC2 Account Information

You will receive an email with information about how to log into your EC2 account, as well as the login and password for the database.

Resources

To help you get started, an AMI is provided with the following software:

- Java 1.6
- Eclipse
- Firefox
- Tomcat 6
- ece1779 web application with all the examples we covered in class and some additional goodies. Specifically,
 - ece1779.ec2.FileUpload.java illustrates the use of im4java and S3
 - ece1779.ec2.StartInstance.java illustrates the EC2 SDK
 - ece1779.ec2.InstanceMetrics.java illustrates the CloudWatch SDK
- AWS Java SDK documentation (in /home/ubuntu/documentation/aws)
- ImageMagick
- im4java (ImageMagick java bindings)
- im4java Documentation (in /home/ubuntu/documentation/im4java)
- im4java Examples (in /home/ubuntu/documentation/im4java/samples)
- mysql-admin and mysql-query-browser
- Xvnc

- twm (a simple window manager)