Lab 4: Google App Engine

Objectives:

1. Understanding how to create application on cloud
2. Using Google App Engine

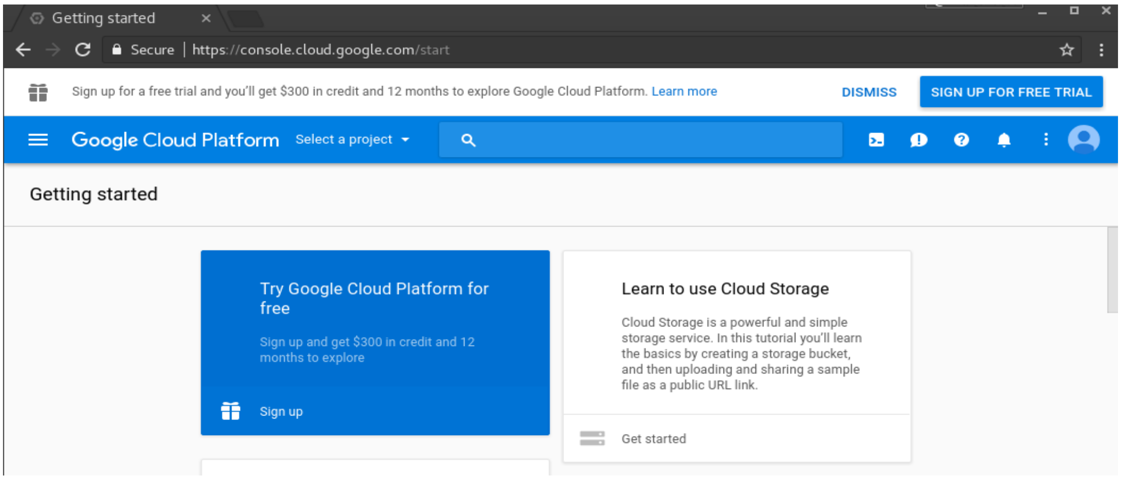
Overview:

1. Sign up a Google account
2. Setup Google Cloud SDK
3. Setup Eclipse
4. Create GAE Application with Eclipse
5. Run your application locally
6. Deploy your application on Google Cloud
7. Lab Assignment

1. Sign up a Google account

create a Google account (if you don't already have one), by visiting https://accounts.google.com/SignUp

Login into your account and go to https://console.cloud.google.com to see your Google Cloud Platform's management page. Verify if you can access to this page.



Sign up for a free trial and you'll get $300 in credit and 12 months to explore Google Cloud Platform. Learn more

Try Google Cloud Platform for free

Sign up and get $300 in credit and 12 months to explore

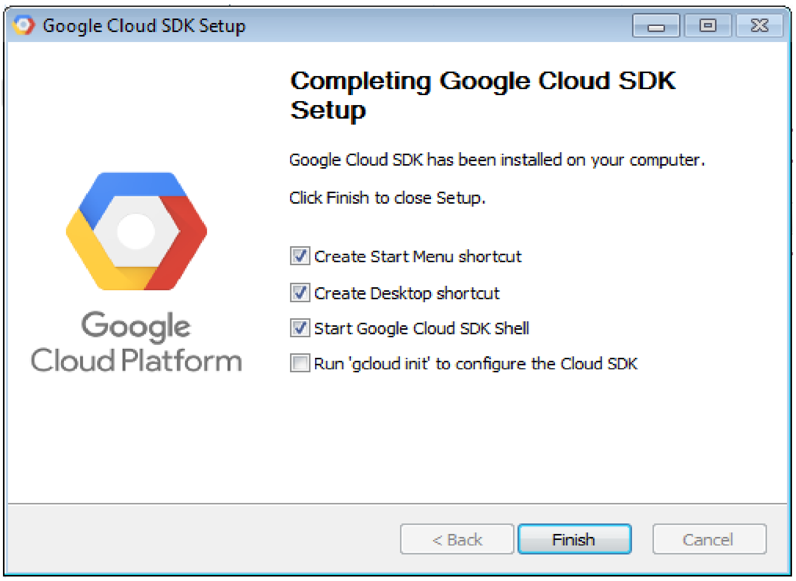
Learn to use Cloud Storage

Cloud Storage is a powerful and simple storage service. In this tutorial you'll learn the basics by creating a storage bucket, and then uploading and sharing a sample file as a public URL link.

2. Setup Google Cloud SDK

To develop application with Google App Engine, Google has prepared a set of tools and libraries for you to download. Go to https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe and install this Software Development Kit (SDK) as a single user.

At the end of installation, **uncheck** the box "*Run 'gcloud init' to configure the Cloud SDK*". (Because we will let Eclipse plugin to handle authentication for us.)



Completing Google Cloud SDK Setup

Google Cloud SDK has been installed on your computer.

Click Finish to close Setup.

☑ Create Start Menu shortcut

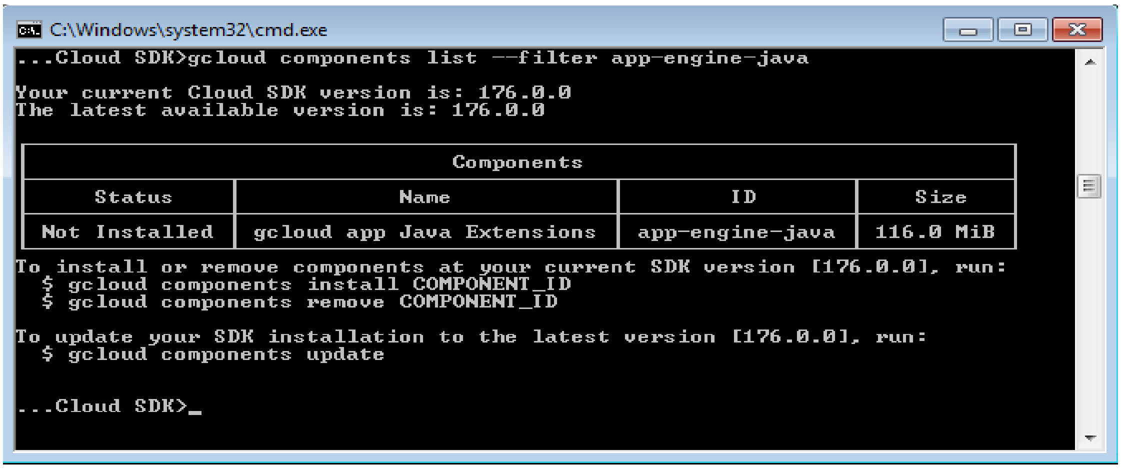
☑ Create Desktop shortcut

☑ Start Google Cloud SDK Shell

□ Run 'gcloud init' to configure the Cloud SDK

After installation, Google Cloud SDK Shell window will be displayed. Here we can list all components installed with Cloud SDK with command: "`gcloud components list`"

See if a component named "**app-engine-java**" is installed by using command: "`gcloud components list --filter app-engine-java`"



```
...Cloud SDK>gcloud components list --filter app-engine-java

Your current Cloud SDK version is: 176.0.0
The latest available version is: 176.0.0

┌──────────────────────────────────────────────────────────────────────────┐
│                               Components                                 │
├───────────────┬────────────────────────────┬─────────────────┬───────────┤
│    Status     │            Name            │       ID        │   Size    │
├───────────────┼────────────────────────────┼─────────────────┼───────────┤
│ Not Installed │ gcloud app Java Extensions │ app-engine-java │ 116.0 MiB │
└───────────────┴────────────────────────────┴─────────────────┴───────────┘
To install or remove components at your current SDK version [176.0.0], run:
  $ gcloud components install COMPONENT_ID
  $ gcloud components remove COMPONENT_ID

To update your SDK installation to the latest version [176.0.0], run:
  $ gcloud components update


...Cloud SDK>_
```

If the status is "**Not Installed**", we need to install it using command: "`gcloud components install app-engine-java`"

Verify again after the installation finished. The status must be changed to "**Installed**"

3. Setup Eclipse

In this lab, we will use Java version 8, and Eclipse version Oxygen (however, any version of Eclipse should be fine). If they are not installed, download and install them from these links:

- Java SDK 8 : http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Eclipse (IDE for Java Deveper version) : http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/oxygen1a

3.1 Setup Proxy for Eclipse (If you are using Mahidol Network)

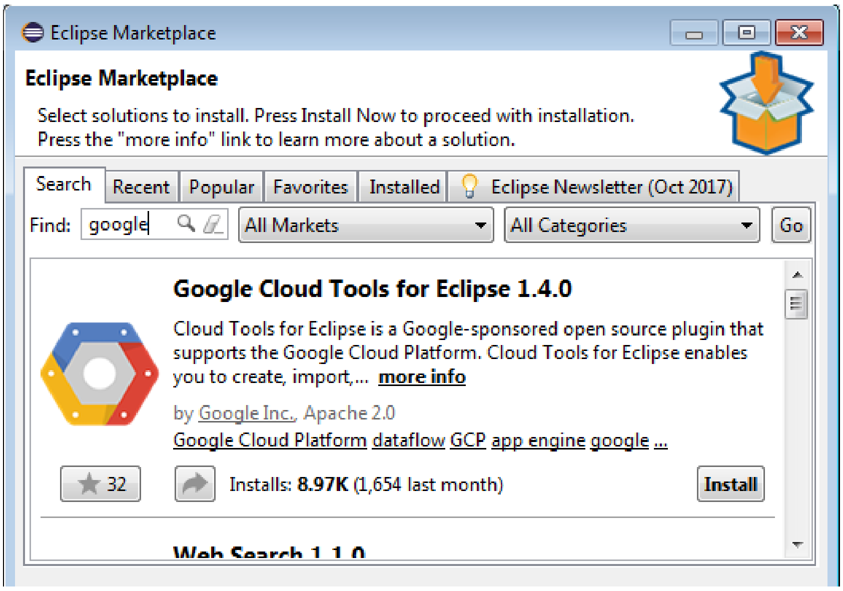
1) In Eclipse program, go to *Window → Preferences → General → Network Connections*
2) At "**Active Provider**" drop-down box, select "**Manual**"
3) In the table, select "**HTTP**" and click "**Edit...**" button
4) Configure it with:

```
Host: proxy-sa.mahidol
Port: 8080
```

5) Tick at "**Requires Authentication:**" check-box
6) Type in your Mahidol's Internet username and password
7) Click "**OK**", then click "**Apply and Close**"

3.2 Install Google Cloud Plugin for Eclipse's

1) In Eclipse program, go to *Help → Market Place*
2) search for "**google**" and install "**Google Cloud Tools for Eclipse**"



Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: google

Google Cloud Tools for Eclipse 1.4.0

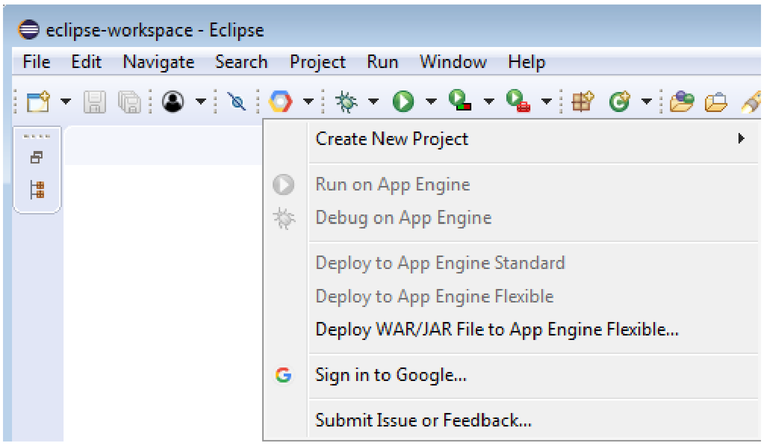
Cloud Tools for Eclipse is a Google-sponsored open source plugin that supports the Google Cloud Platform. Cloud Tools for Eclipse enables you to create, import,... more info

by Google Inc., Apache 2.0

Google Cloud Platform dataflow GCP app engine google ...

Installs: **8.97K** (1,654 last month)

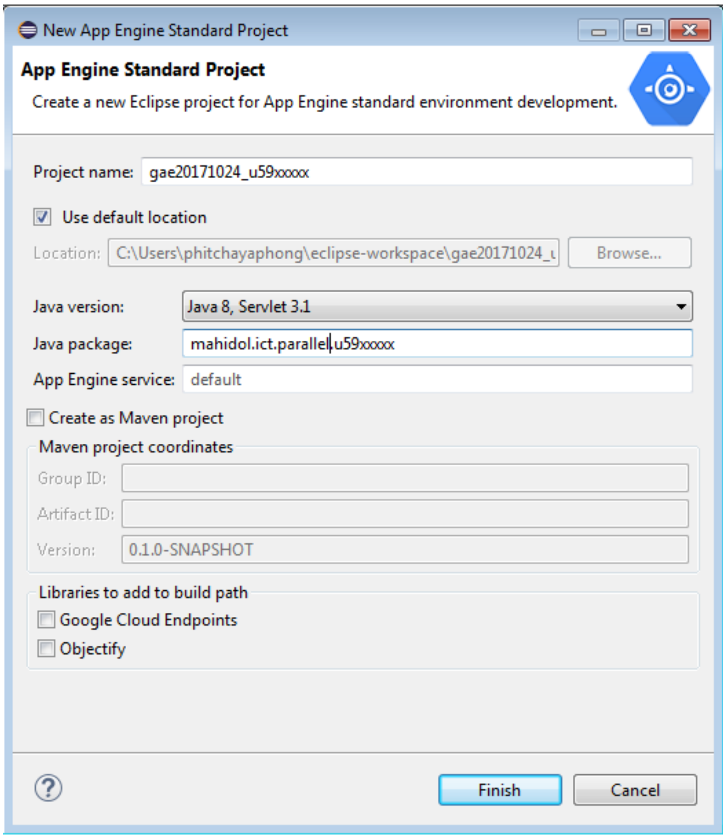
3) After installation, restart Eclipse program
4) Click on "Google Cloud Platform" icon and select "**Sign in to Google...**"


eclipse-workspace - Eclipse
File Edit Navigate Search Project Run Window Help
Create New Project
Run on App Engine
Debug on App Engine
Deploy to App Engine Standard
Deploy to App Engine Flexible
Deploy WAR/JAR File to App Engine Flexible...
Sign in to Google...
Submit Issue or Feedback...

5) Login to your Google Account and allow the permissions.
6) After you finished, you will see message “*Received verification code. You may now close this window...*”. You can close the web browser.

4. Create an Google App Engine Application

1) Click on the “Google Cloud Platform” button ( ) again.
2) Choose *Create New Project → Google App Engine Standard Java Project…*
3) Type-in project name and Java Package. You can choose anything, but try to make it unique. Here is an example.


New App Engine Standard Project
App Engine Standard Project
Create a new Eclipse project for App Engine standard environment development.
Browse...
Maven project coordinates
Libraries to add to build path
Finish
Cancel

Project name: gae20171024_u59xxxxx

☑ Use default location

Location: C:\Users\phitchayaphong\eclipse-workspace\gae20171024_u

Java version: Java 8, Servlet 3.1

Java package: mahidol.ict.parallel.u59xxxxx

App Engine service: default

☐ Create as Maven project

Group ID:

Artifact ID:

Version: 0.1.0-SNAPSHOT

☐ Google Cloud Endpoints

☐ Objectify

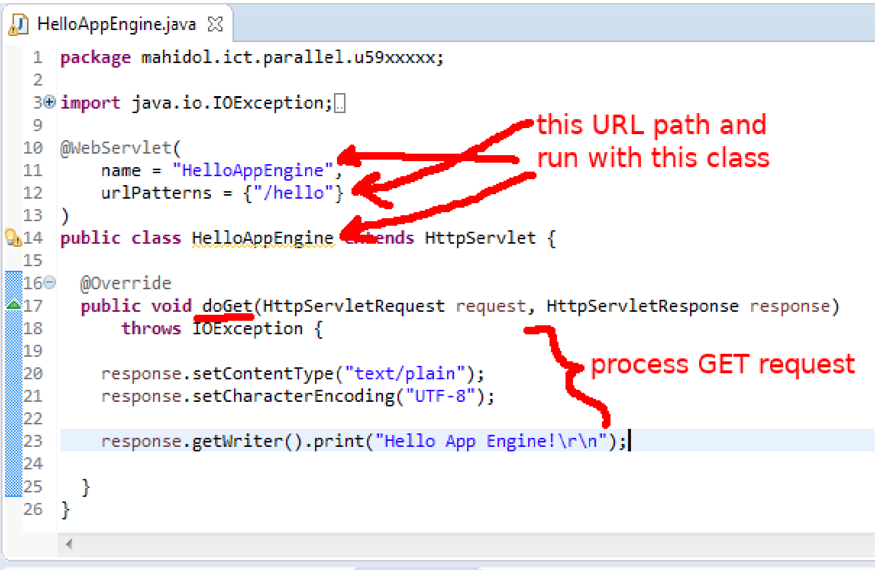
4) Click finish.

5) Eclipse will create a new project with some simple code for Google App Engine. The code will simply print “Hello App Engine!” out.

Code Explanation

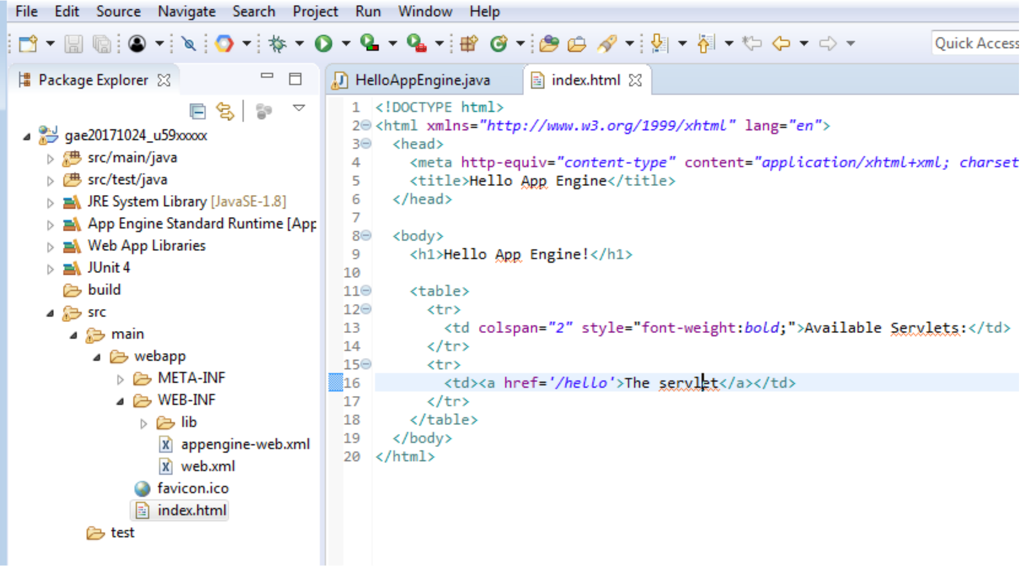
Google App Engine (Java) is an Java Servlet application. It maps each class to each URL path.

For example, this class, “**`HelloAppEngine`**”, is mapped to path “**/hello**”. That is when running this application, you can go to URL “http://localhost:8080/hello” and it will print out the words: “Hello App Engine!”.



```java
package mahidol.ict.parallel.u59xxxxx;

import java.io.IOException;

@WebServlet(
    name = "HelloAppEngine",
    urlPatterns = {"/hello"}
)
public class HelloAppEngine extends HttpServlet {

  @Override
  public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws IOException {

    response.setContentType("text/plain");
    response.setCharacterEncoding("UTF-8");

    response.getWriter().print("Hello App Engine!\r\n");

  }
}
```

However, the default front page is “**index.html**”, located in “**src/main/webapp**” folder. If you would like to modify the front page, you can change it there. It is written in HTML language.



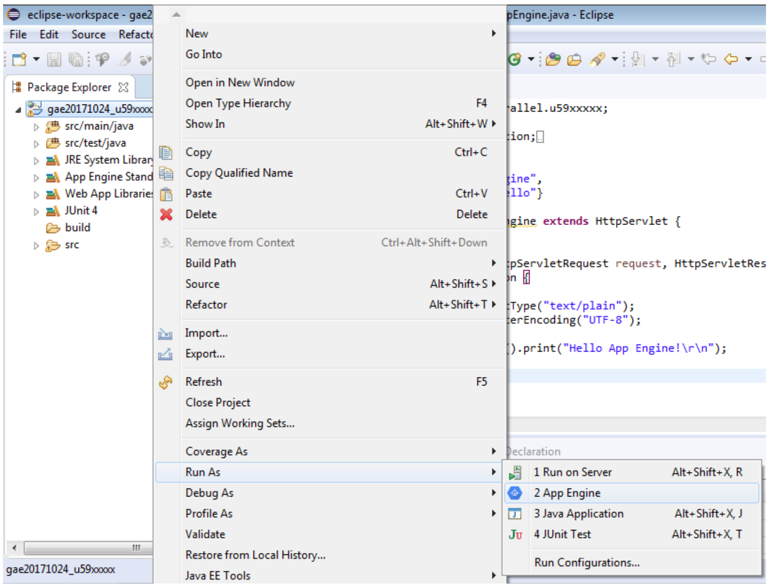
```
gae20171024_u59xxxxx
  src/main/java
  src/test/java
  JRE System Library [JavaSE-1.8]
  App Engine Standard Runtime [App
  Web App Libraries
  JUnit 4
  build
  src
    main
      webapp
        META-INF
        WEB-INF
          lib
          appengine-web.xml
          web.xml
        favicon.ico
        index.html
    test
```

```html
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
  <head>
    <meta http-equiv="content-type" content="application/xhtml+xml; charset
    <title>Hello App Engine</title>
  </head>

  <body>
    <h1>Hello App Engine!</h1>

    <table>
      <tr>
        <td colspan="2" style="font-weight:bold;">Available Servlets:</td>
      </tr>
      <tr>
        <td><a href='/hello'>The servlet</a></td>
      </tr>
    </table>
  </body>
</html>
```

5. Run your application locally

5.1 Run the Server

1) Right click on your project, select *Run As* → *App Engine*

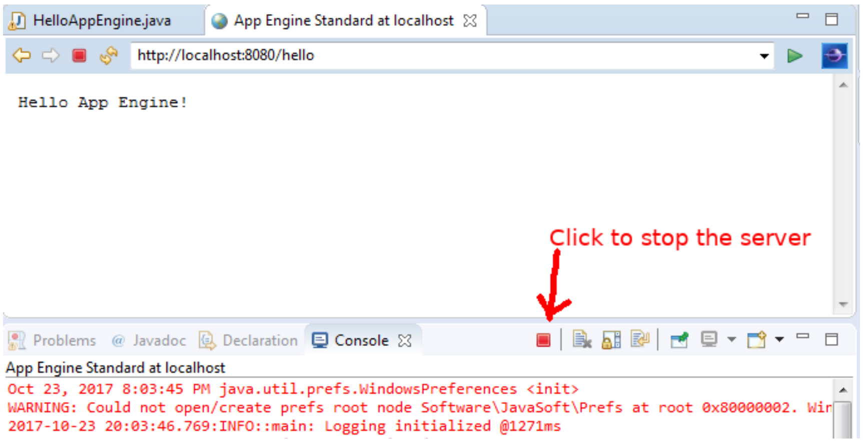


```
allel.u59xxxxx;
ion;

ine",
llo"}
gine extends HttpServlet {

pServletRequest request, HttpServletRes
n {

Type("text/plain");
erEncoding("UTF-8");

).print("Hello App Engine!\r\n");
```

2) You will be able to see the application running on http://localhost:8080/ using any web browser.

5.2 Stop the Server

When you modify the code and want to see the changes, you must **stop the server** before running the application again.



```
Hello App Engine!
```

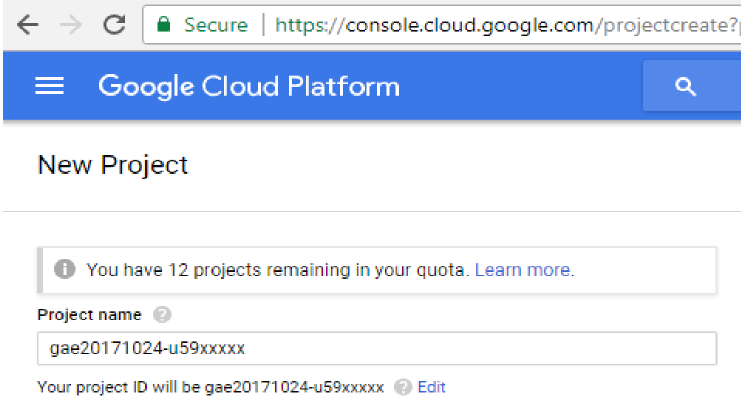
Click to stop the server

```
App Engine Standard at localhost
Oct 23, 2017 8:03:45 PM java.util.prefs.WindowsPreferences <init>
WARNING: Could not open/create prefs root node Software\JavaSoft\Prefs at root 0x80000002. Wir
2017-10-23 20:03:46.769:INFO::main: Logging initialized @1271ms
```

6. Deploy your application on Google Cloud

6.1 Create Project on Google Cloud

1) First, use a web browser to go to Google Cloud website to create a new project https://console.cloud.google.com/projectcreate
2) Pick a **project name** that is unique. And remember your "**Project ID**" (usually be the same as project name). It will be used later.



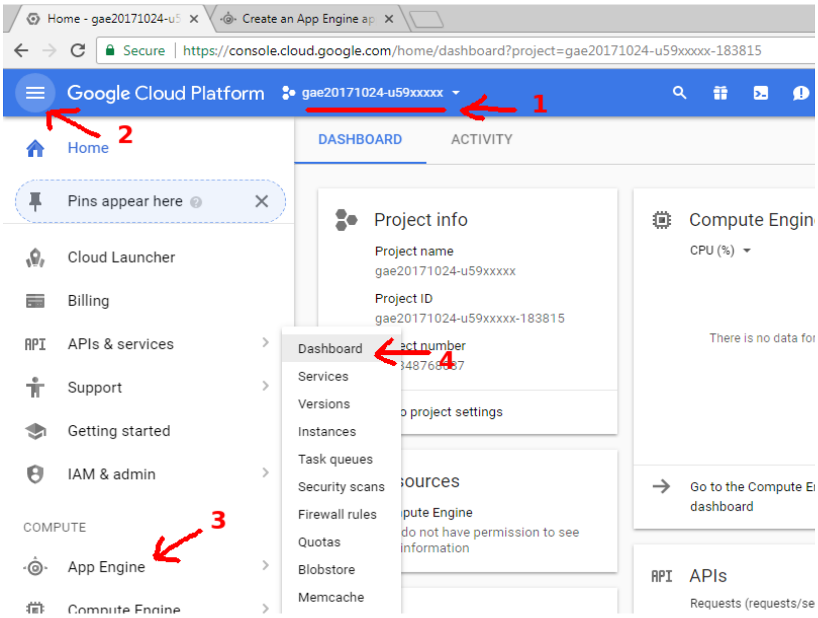
You have 12 projects remaining in your quota.

Project name: gae20171024-u59xxxxx

Your project ID will be gae20171024-u59xxxxx

3) Click "**Create**"

6.2 Create App Engine inside the Project



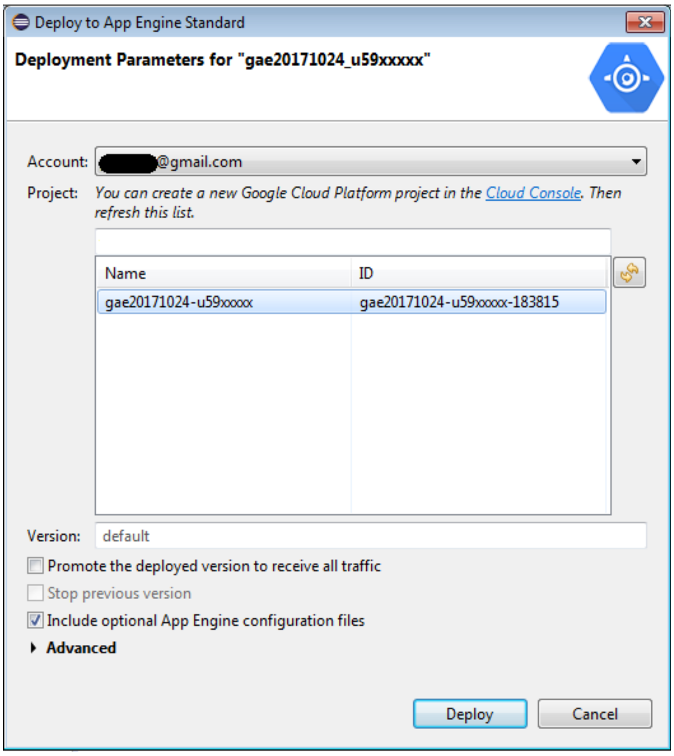
Project name: gae20171024-u59xxxxx

Project ID: gae20171024-u59xxxxx-183815

1) Select your new project
2) Click at menu
3) Select "*App Engine*"
4) Select "*Dashboard*"
5) Choose "Your first app → Java"
6) Select location of the server where the app will be run on.
   - Closest to Thailand is "**asia-northeast1**"
7) If the next page says "*Let's get started*", then your app is created. You can skip the tutorial and close the browser.

6.3 Deploy App from Eclipse to Google Cloud

1) Go back to Eclipse program
2) Select your project folder (on the left-hand side)
3) Click on the “Google Cloud Platform” button ( ) again.
4) Select “*Deploy to App Engine Standard*”.
5) Select the project name you just created.



Deploy to App Engine Standard

Deployment Parameters for "gae20171024_u59xxxxx"

Account: [redacted]@gmail.com

Project: *You can create a new Google Cloud Platform project in the Cloud Console. Then refresh this list.*

Name	ID
gae20171024-u59xxxxx	gae20171024-u59xxxxx-183815

Version: default

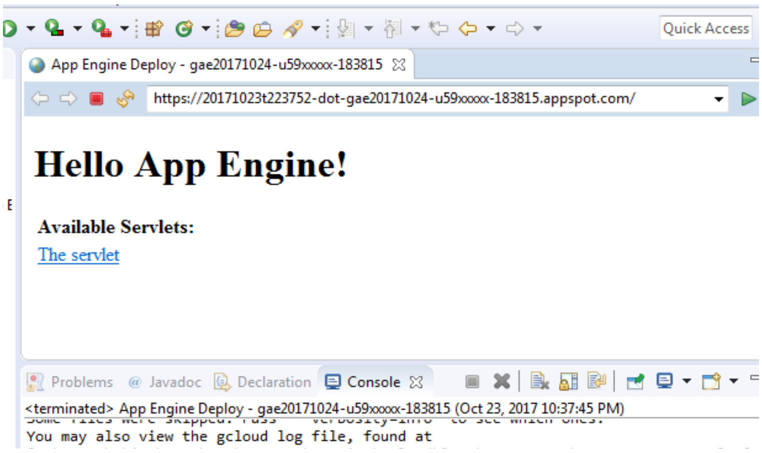
□ Promote the deployed version to receive all traffic

□ Stop previous version

☑ Include optional App Engine configuration files

▸ **Advanced**

4) **Uncheck** “*Promote the deployed version to receive all traffic*”, so not much traffic will come to your app. This prevents the app to cost you the money.
5) Click “**Deploy**” to deploy to the Google Cloud
6) Wait until it finishes, and you will be able to browse to your Google Application.



https://20171023t223752-dot-gae20171024-u59xxxxx-183815.appspot.com/

Hello App Engine!

Available Servlets:

The servlet

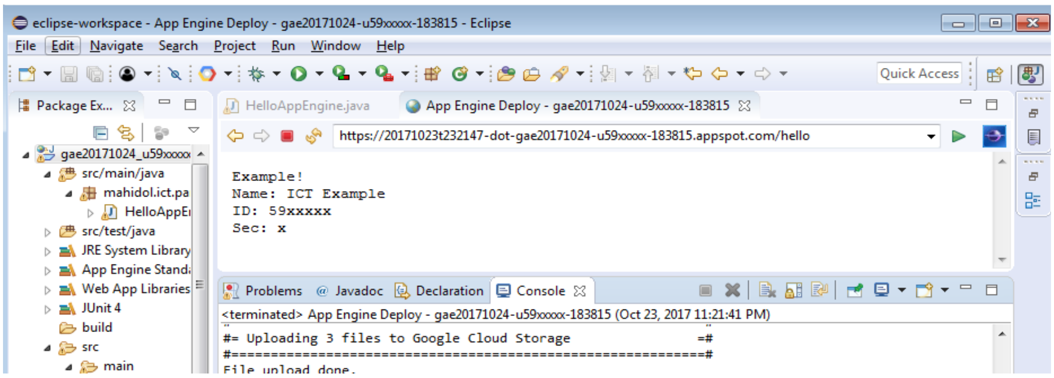
```
<terminated> App Engine Deploy - gae20171024-u59xxxxx-183815 (Oct 23, 2017 10:37:45 PM)
You may also view the gcloud log file, found at
```

Note: The URL is public. Anyone with the URL can access to this page.

Lab Assignment

1. Modify the application to print out your full name, student ID, and section.
2. Save the screen shot of your working Google Application running on Google Cloud in a Microsoft Word document, along with your application's URL.
   - The screen shot must contain your **full name**, **student ID** and **section**.
   - The screen shot must contain your **application URL** in the URL bar.

   For example (no need to follow this exactly) :



eclipse-workspace - App Engine Deploy - gae20171024-u59xxxxx-183815 - Eclipse

https://20171023t232147-dot-gae20171024-u59xxxxx-183815.appspot.com/hello

```
Example!
Name: ICT Example
ID: 59xxxxx
Sec: x
```

<terminated> App Engine Deploy - gae20171024-u59xxxxx-183815 (Oct 23, 2017 11:21:41 PM)

```
#= Uploading 3 files to Google Cloud Storage              =#
#==========================================================#
File upload done.
```

3. Save your word file as u59xxxxx.docx and submit it to the e-learning website.