Cloud Computing

Internet Programming-Technology Research Paper

Huang Fan (G2002821G) Instructor: Prof. Wong Twee Wee

1. Abstract

With the development of internet technology, the design and development of network information system is becoming more and more complex, which puts forward higher requirements for the information technology industry. Web developers are required improve their efficiency, and ensure the project is easy to maintain and upgrade. Now, there are more and more professional and mature cloud service providers. Choosing a suitable cloud computing platform could largely improve the developers' development efficiency and make the project management process easier. I used, adjusted, and deployed several useful functions through API provided in Google Cloud. The settings and implementation processes are elaborated in the implementation part, the source codes and the examples are attached in the appendix part.

2. Introduction

The Platform as a service (PaaS) is a model that allows developers ignore the complexity of maintaining the infrastructures when developing and operating. The IDE plugins would make the development process with PaaS even more simpler. What is more, the Infrastructure as a Service (IaaS) is less convenient than PaaS. The PaaS also suits well with JVM. So it is better to use platforms in PaaS.[1]

Developers should also keep the following tips into consideration when choosing PaaS platform:

(1) Be flexible when choosing stacks like the server or the database.

(2) Be capable to control JVM tuning.

(3) Have proper and convenient log-in steps.[1]

(4) Be highly scalable.[3]

2.1.Cloud platforms supporting JAVA

Some major Cloud Hosting Provides nowadays are listed below.[1-3]

(1) Amazon Web Services (AWS)

AWS remains to be the largest service provider for a long time since 2002. It allows developers to run almost all kinds of application in the cloud. For example, the projects like scalable big data projects, enterprise-level applications and mobile applications. Amazon also provide AWS Toolkit for Eclipse and SDK for Java.

(2) Microsoft Azure

Azure is also a well-refined and comprehensive product. It allows developers to set virtual machines running Windows or Linux. It also support plugin for eclipse and Java stacks with a free trial to get familiar with it. It focus more on the enterprise-level

(3) Google Cloud Platform

GCP is well-known for its high security and analysis toolkits. It is written in Java with some Python and Go. It now support Java 8 or Java 11 environment. It uses Java Virtual Machine and Java Servlets interface. It has a lot tools and interfaces for developers to access. The GCP also provide development plugins for IDEs like eclipse and IDEA.

(4) Others

IBM cloud, Dell EMC, Salesforce Cloud, Oracle Cloud, BitNami, Jelastic, Heroku

2.2.Deployment preparation

I would choose Google Cloud Platform because its high security level and comprehensiveness of the toolkits, especially for its machine learning toolkits.

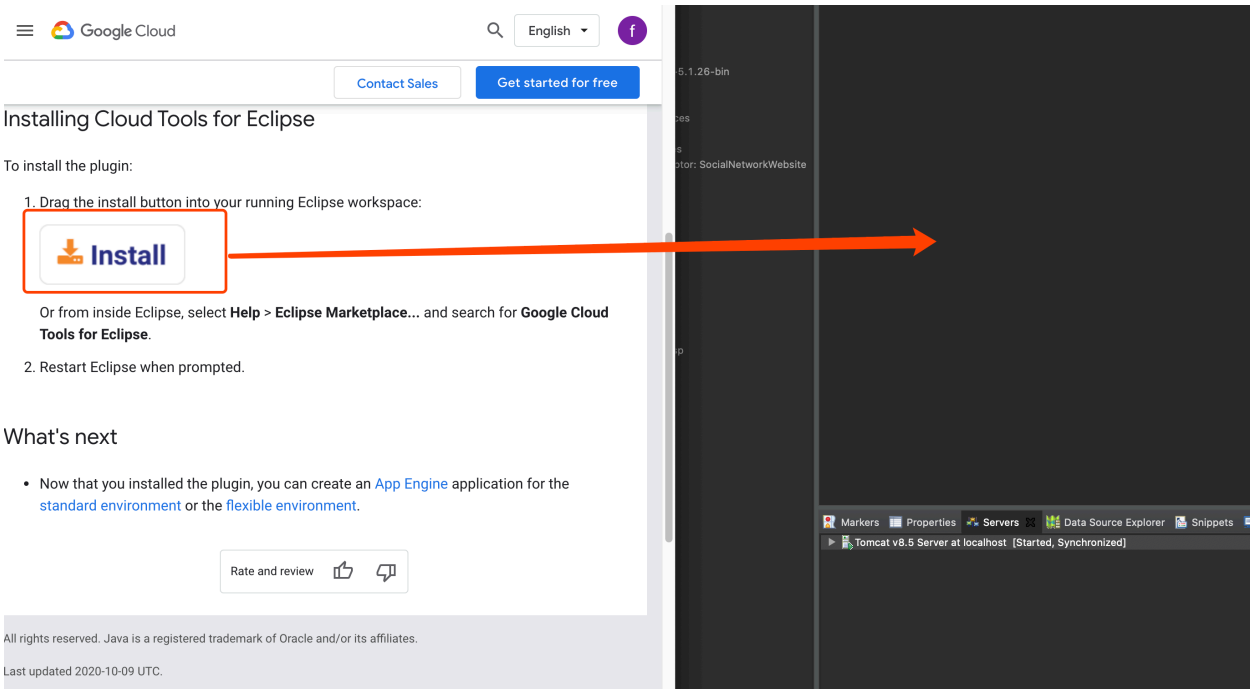
3. Implementation

The Google App Engine is a PaaS platform, which means developers have no need to maintain a server. Anyone knows the basic web development skills could create a convenient collaborate project by the usage of GAE and Eclipse with plugin.[4-5] GAE also provides standard project tuition (in Java 8 and in Java 11) and flexible project tuition.

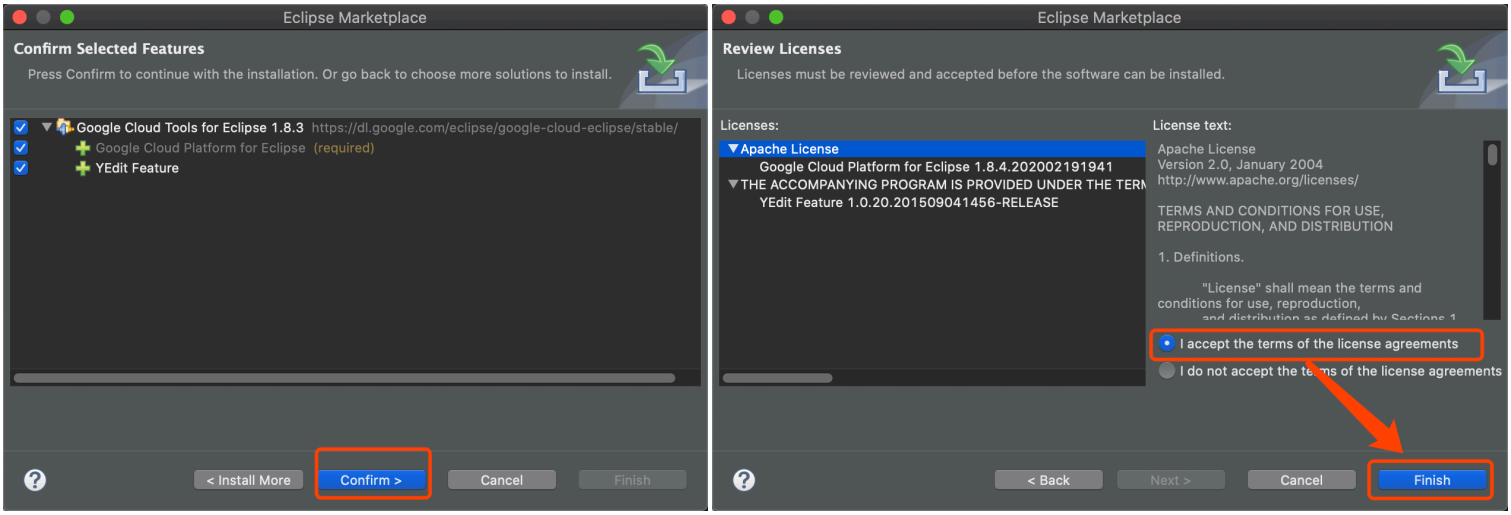
3.1.Environment and settings

MacBook Pro 2018, Eclipse 2020-06, JRE 1.8

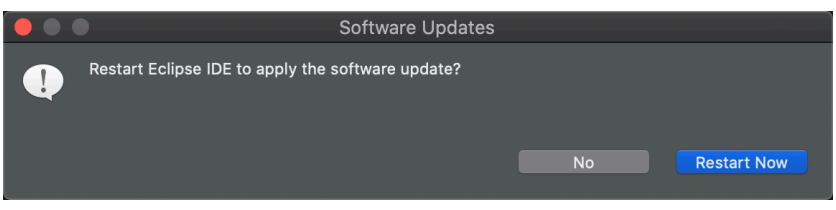
(1) Install the plugin by drag the button into the running Eclipse



(2) Click “Confirm” button. In the next page, click “I accept the terms …” and click “Finish” button.



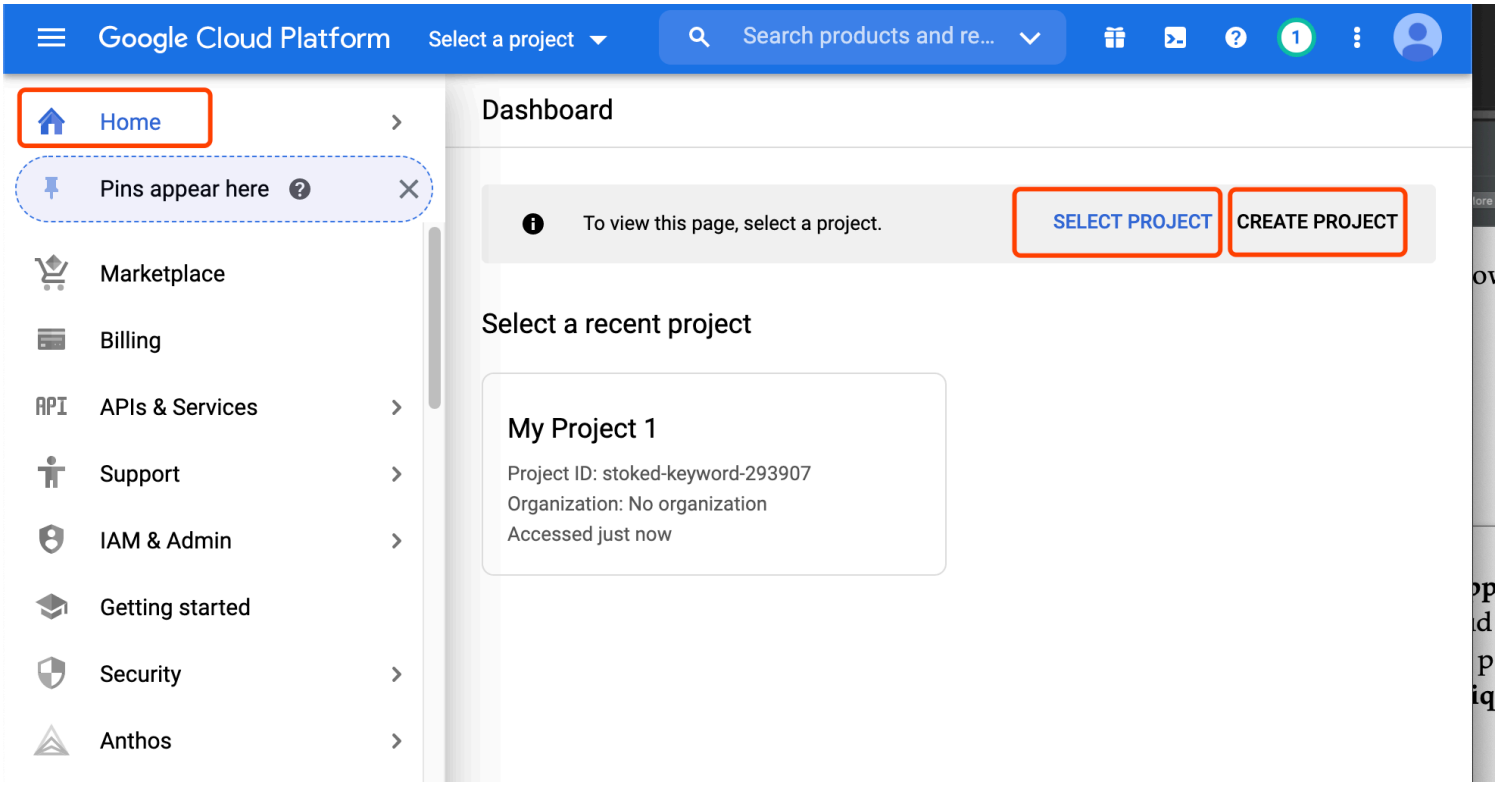
(3) Click “Restart now” button



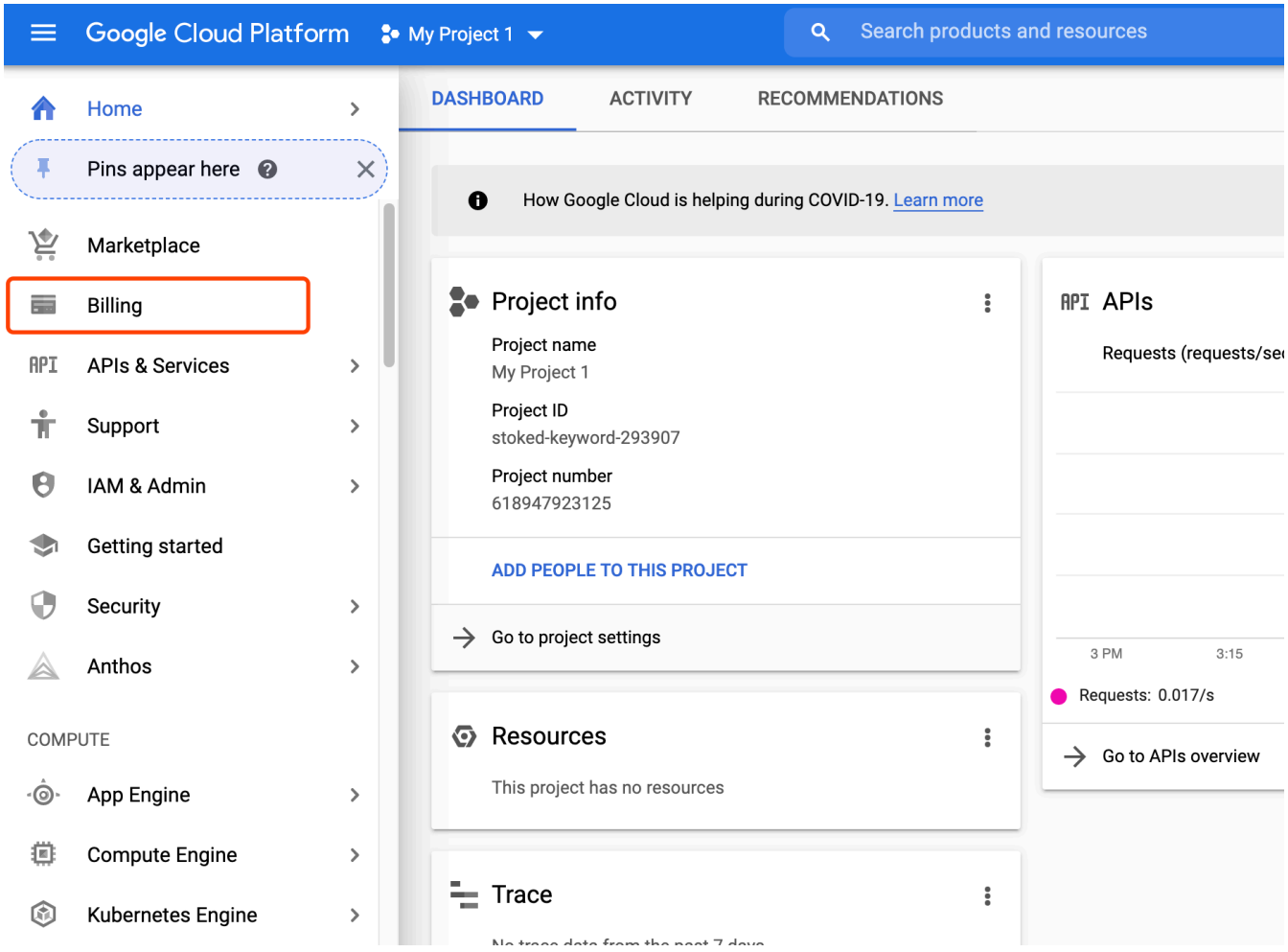
3.2.Create an App Engine Standard Application

(1) Create a cloud project in the project selector page of Cloud Console, or just click "console" button of the official guide page of Google Cloud and then choose "home" in the sidebar.

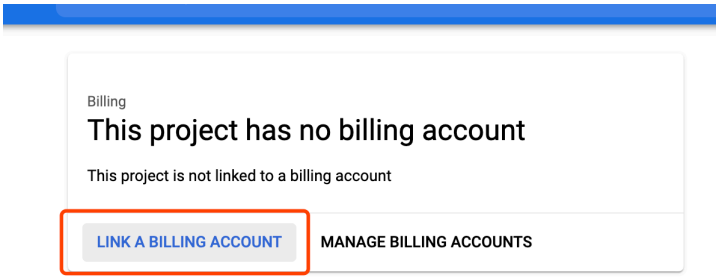
(2) Create a new project or just select a existing project like "My Project 1"

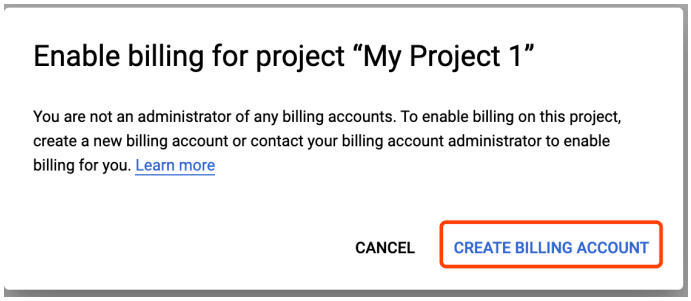


(3) Select the target project, and add billing settings in the "My Project 1" detailed page inside console

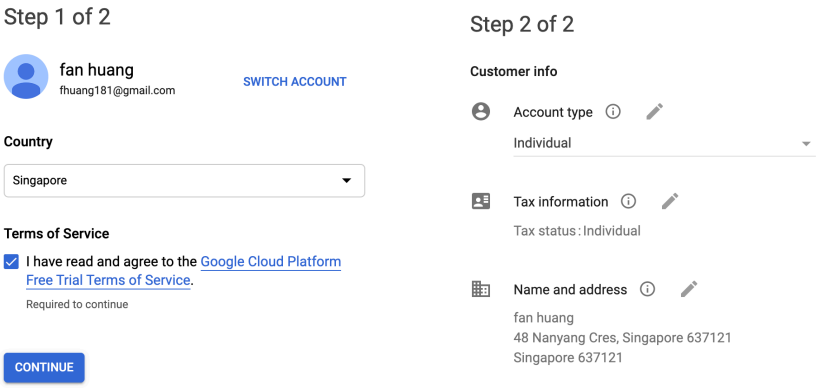


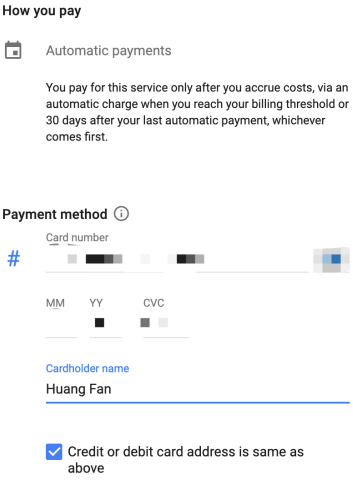
(4) If you have not set a linked billing account, then you have to click the "Link a billing account" and then add a credit card to your Google account to pay the possible charges when running program.



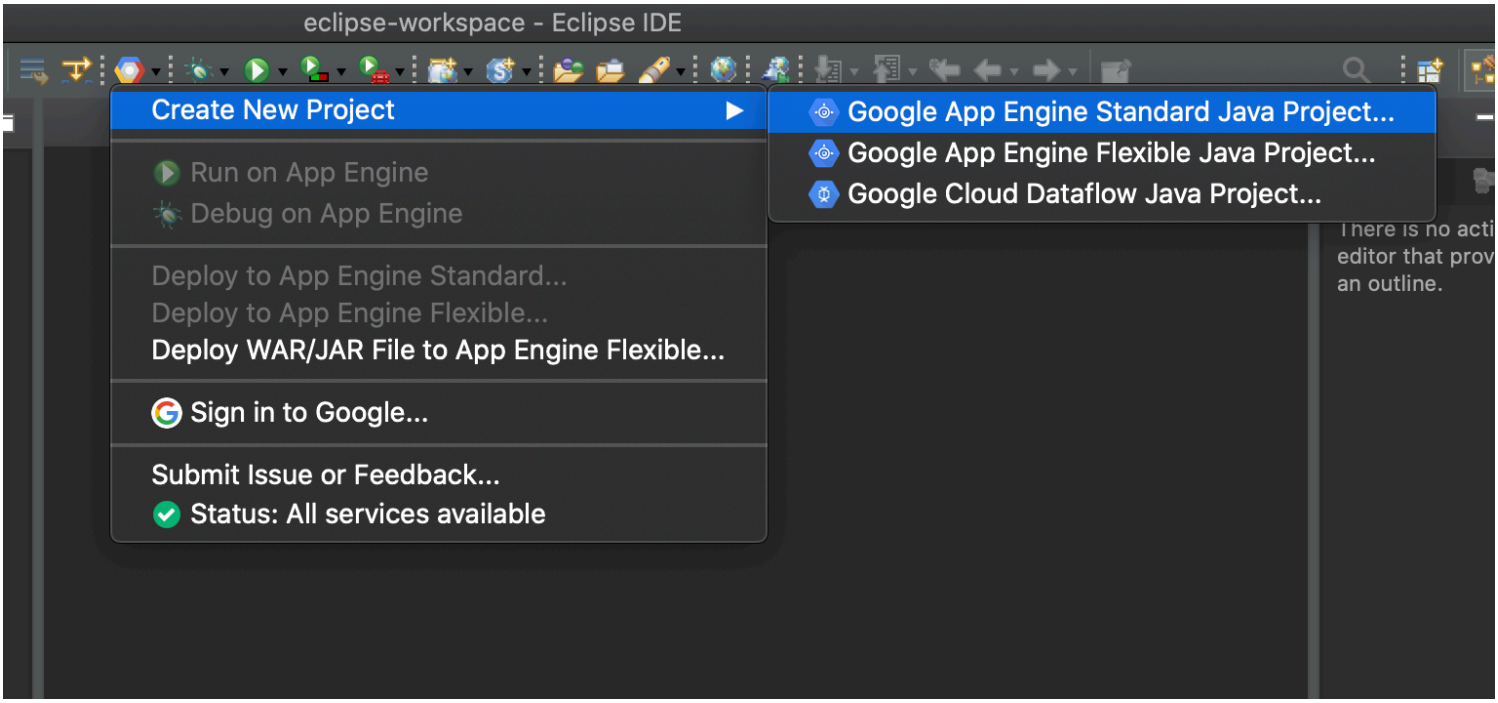


(5) (Optional for those developer who have already set their billing accounts) Create billing account. Google now provide free $300 in the account for new users to try and would not auto charge after free trail ends. The auto billing function is "No" by default.

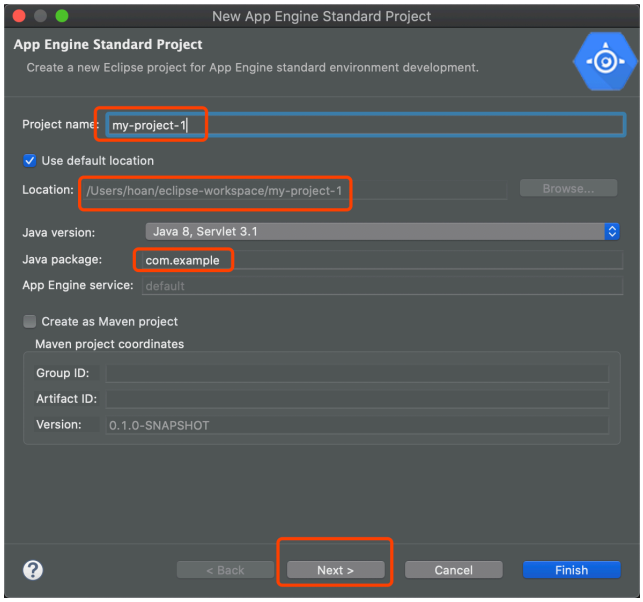


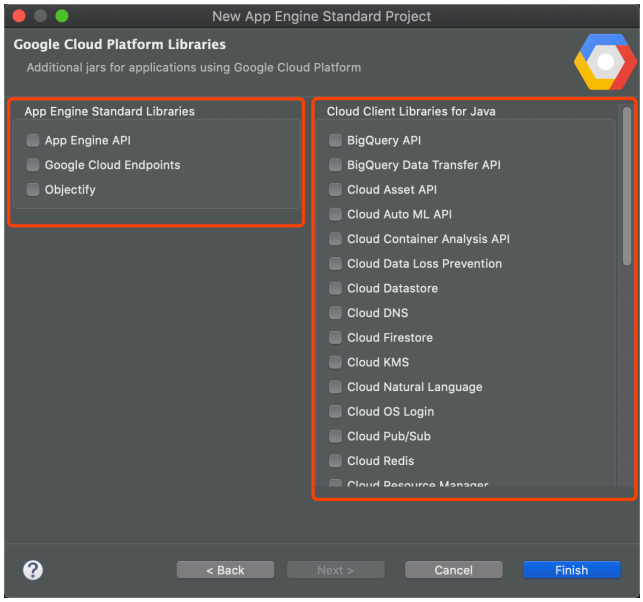


(6) Create a new project in eclipse. Click the Google Cloud Platform toolbar button. And select ""Create New Project"->"Google App Engine Standard Java Project"



(7) Config project settings. Enter the project name and a Java package(optional). Could also choose to create a Maven-based App Engine project (optional), enter the Group ID and Artifact ID. Then click "Next". In the second page, developers could choose libraries in need and click "Finish" to complete.

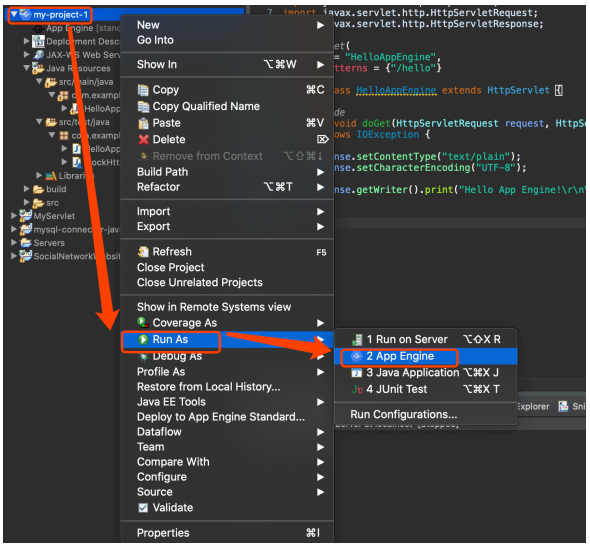


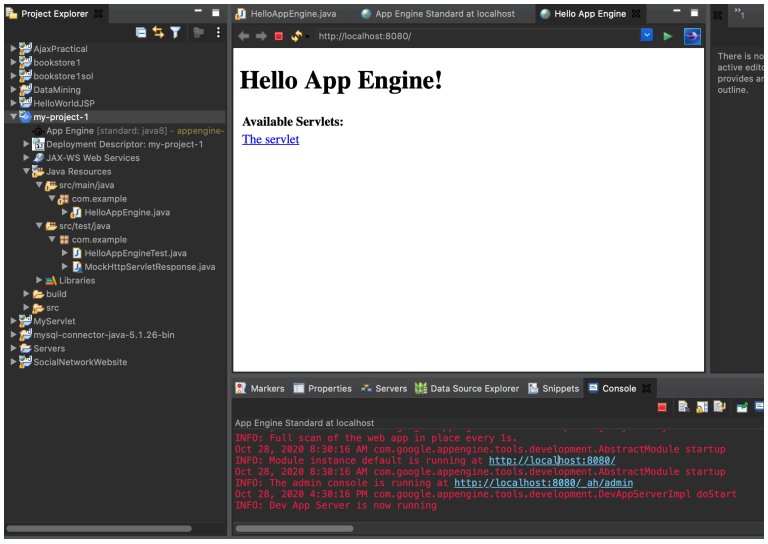


The eclipse then creates a project with a simple servlet, which developers could modify and run through eclipse.[6]

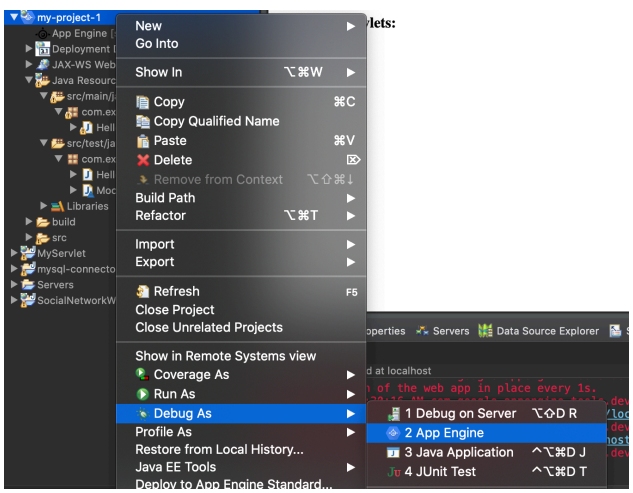
3.3.Run and debug locally inside eclipse

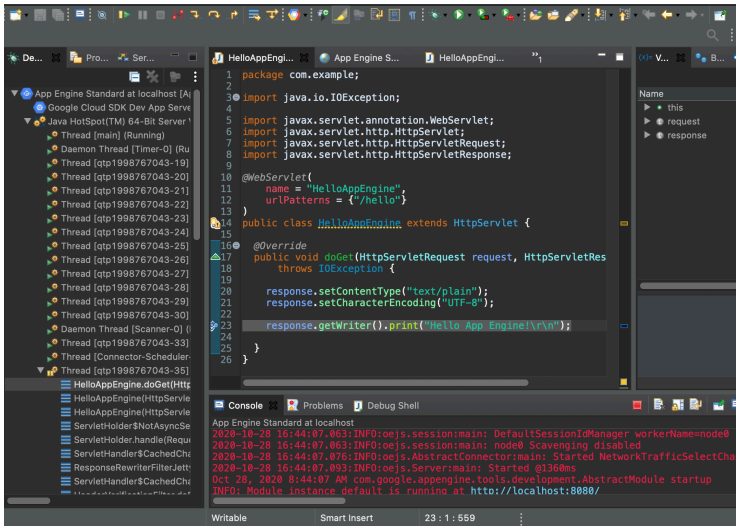
(1) Run the project locally. Select “Run As”->”App Engine”. Then the log messages appears in the console of eclipse. Developers could also use browsers through url “http://localhost:8080”.





(2) Debug the project locally. Stop the App engine in “Servers” windows and then select “Debug As”->”App Engine” after right click the project. The server would stop at the breakpoints developers set and shows the debugger view in eclipse.

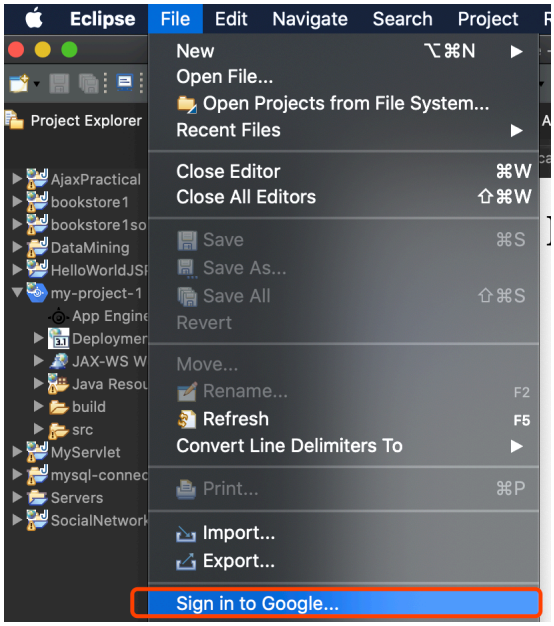


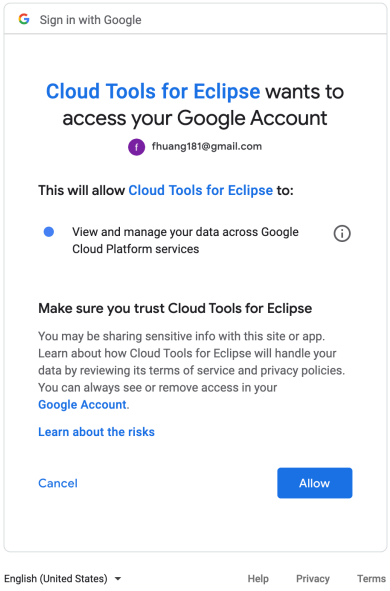


(3) Could run App Engine Standard Apps on a different host or port by manually adjust the configurations.[7]

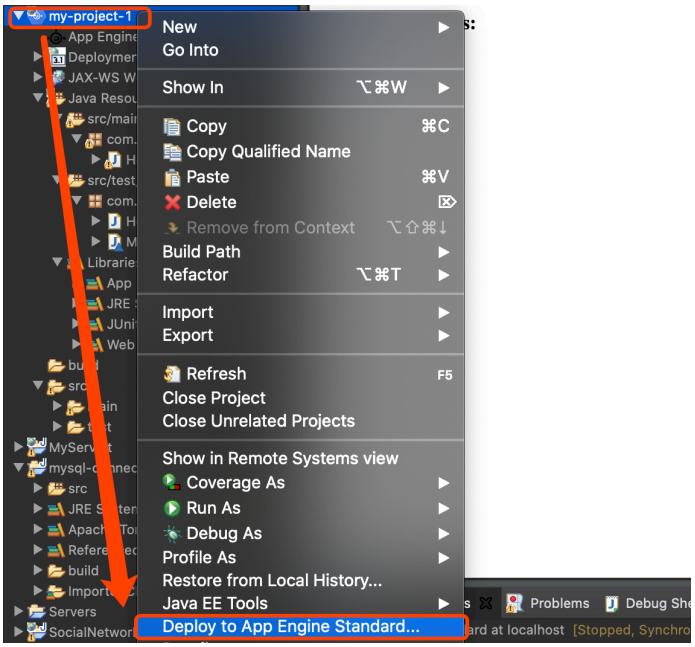
3.4.Deploy project to App Engine

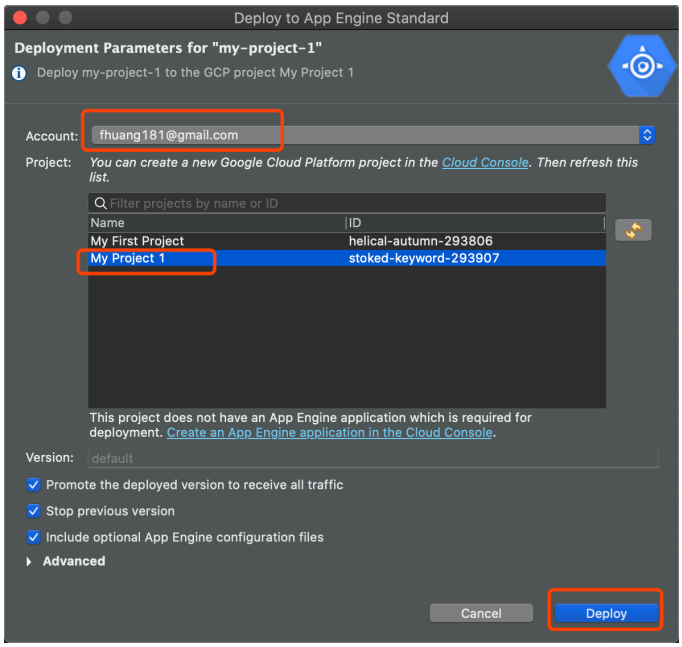
(1) Use the cloud project created in the step 5 of "3.2.Create an App Engine Standard Application". Sign in to the google account in eclipse. Select "File"->"Sign in to Google", developers could skip this if you have already signed in. Then click "Allow" in the pop up window of browser to confirm sign in.





(2) Deploy the project to App Engine. Right click the project in the Package Explorer and select "Deploy to App Engine Standard". In the pop up dialog window, developers should select the account and the cloud project. Click "deploy" to confirm.

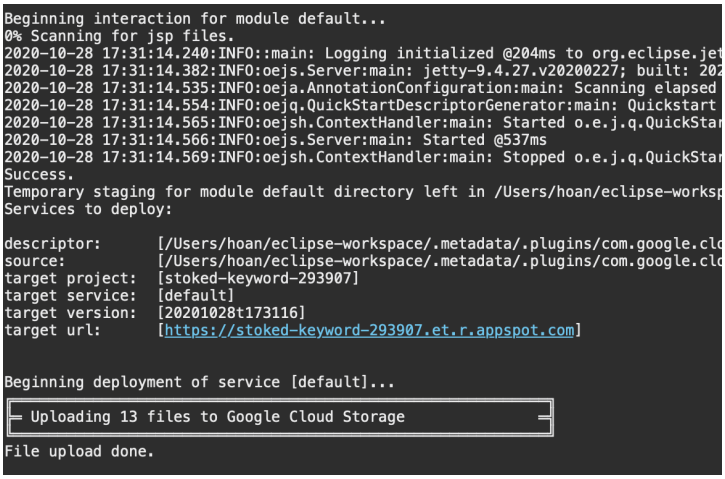


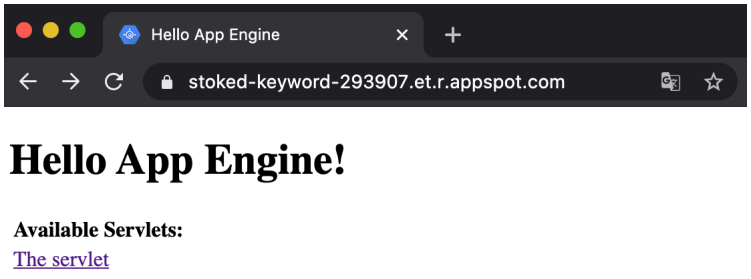


(3) GAE deployed service by default to [https://stoked-keyword-293907.et.r.appspot.com], which could show the web page of the project even after stop eclipse. Developers can stream logs from the command line by running: "$ gcloud app logs tail -s default". Developers could also application in the web browser run: "$ gcloud app browse —project=stoked-keyword-293907".

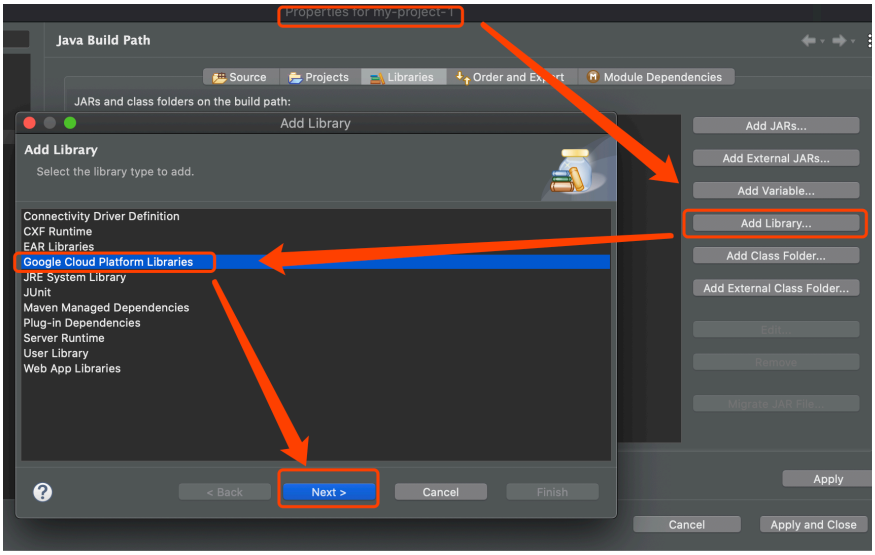
(4) After confirm to deploy, the console in eclipse would print detailed steps of the deployment process. The official guide also mentions that App Engine would stop the previous version of application and

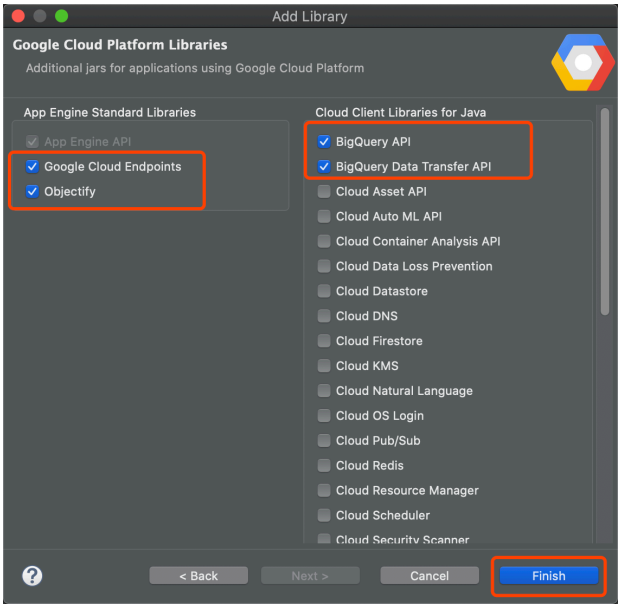
immediately promote the new code to receive all network traffic. Developers are able to change the default settings in Google Cloud Console.[8]





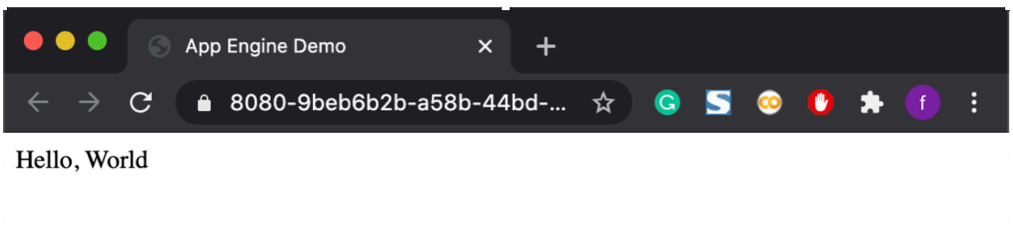
(5) Manage Cloud Client Libraries. Select the project in the package explorer. Right click and choose "Build Path"->"Add Libraries", then select "Google Cloud Platform Libraries"->"Next". After that, select libraries and click "Finish" to complete installation.





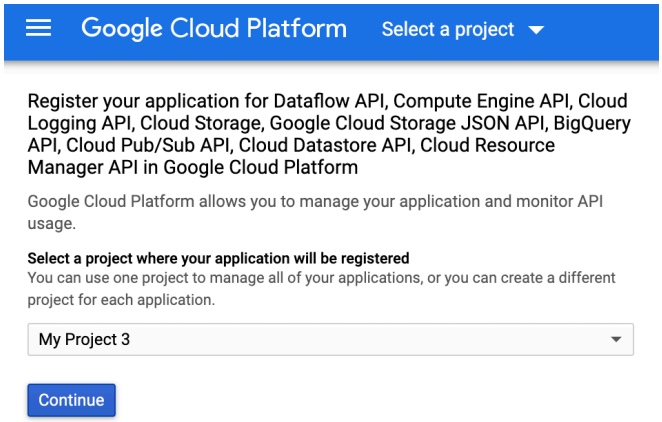
3.5.Build and run "Hello World" app by the console of Google Cloud Platform

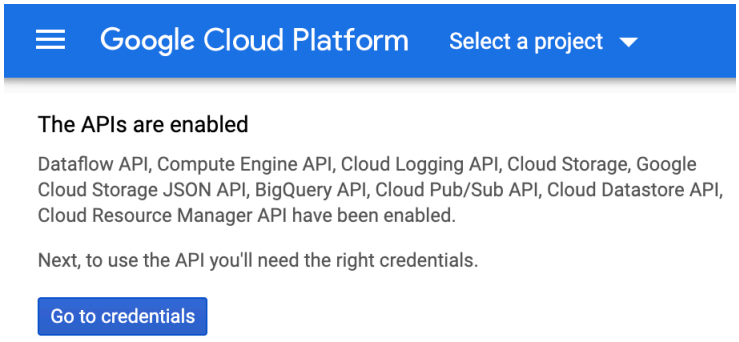
Select a project, for example "My Project 2". Activate and use cloud shell to deploy a "Hello World" sample. Then use "mvn appengine:run" command to active the server. Can see the website through this url [https://8080-9beb6b2b-a58b-44bd-9633-5a564e23deec.asia-southeast1.cloudshell.dev/?authuser=0]



3.6.Advanced techniques

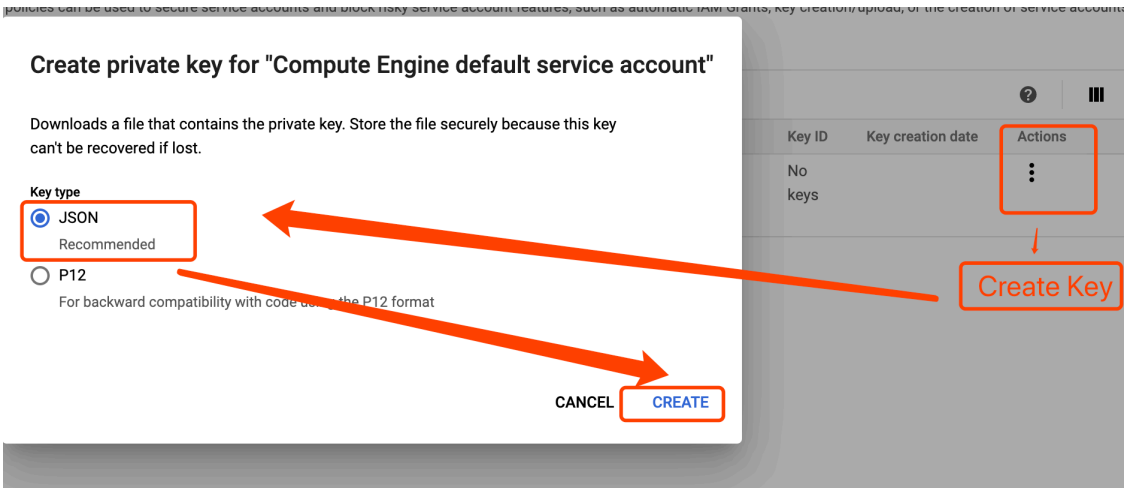
Use Cloud Dataflow in eclipse and run an example pipeline.[10] Dataflow aims to provide unified system and batch data processing services. The services are serverless, fast, and cost-effective. Detailed





information could be found in the official website. The "WordCount" sample generation processes are listed below.

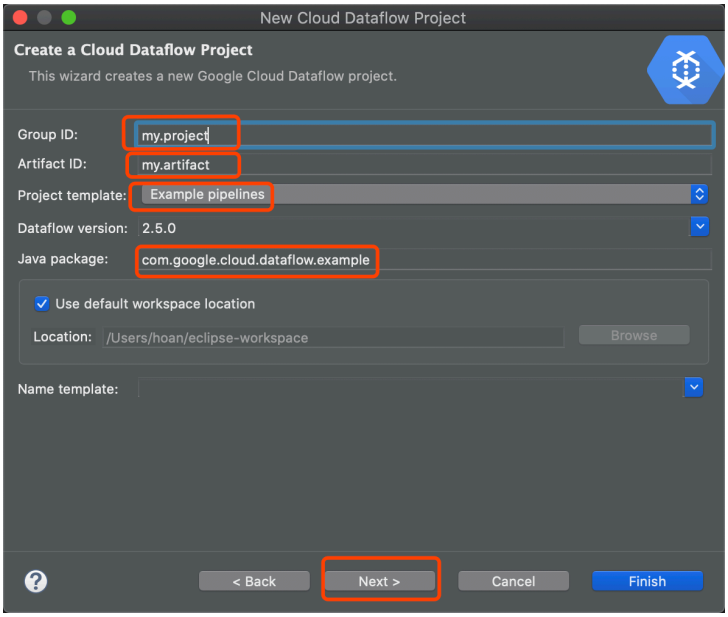
(1) Create a new cloud project "My Project 3". Enable require APIs.

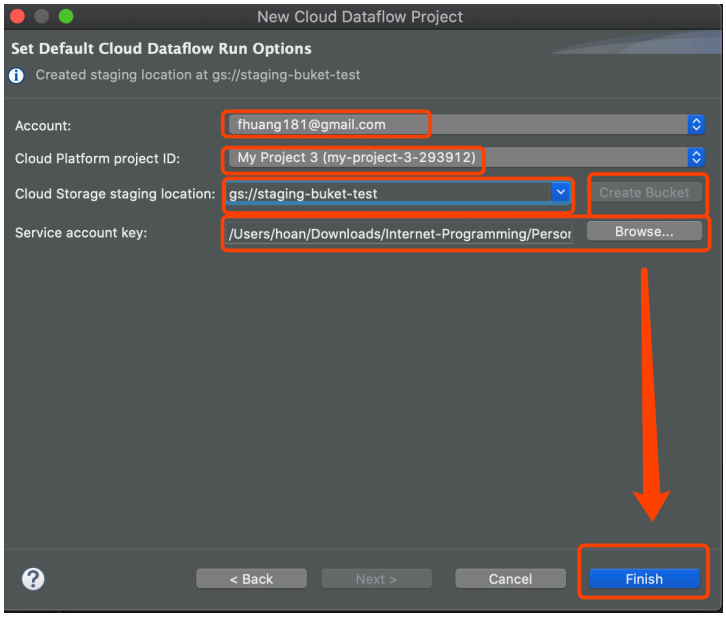


(2) Install Google Cloud SDK. The instruction is in [https://cloud.google.com/sdk/docs/install].

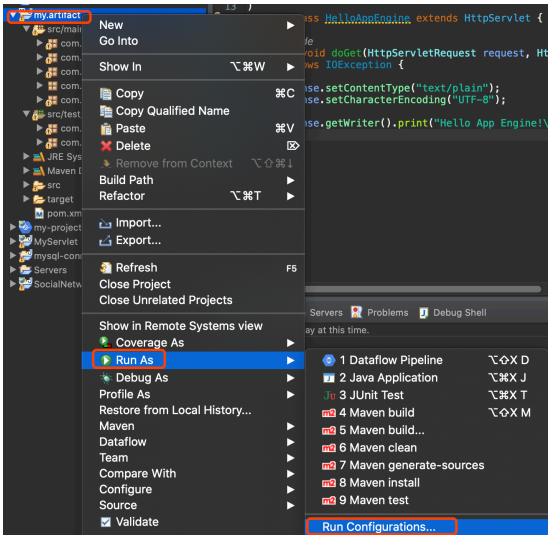
(3) Go to Google Cloud Platform console, select "My Project 3"->"IAM & Admin"->"Service Accounts", create a JSON key and download it.

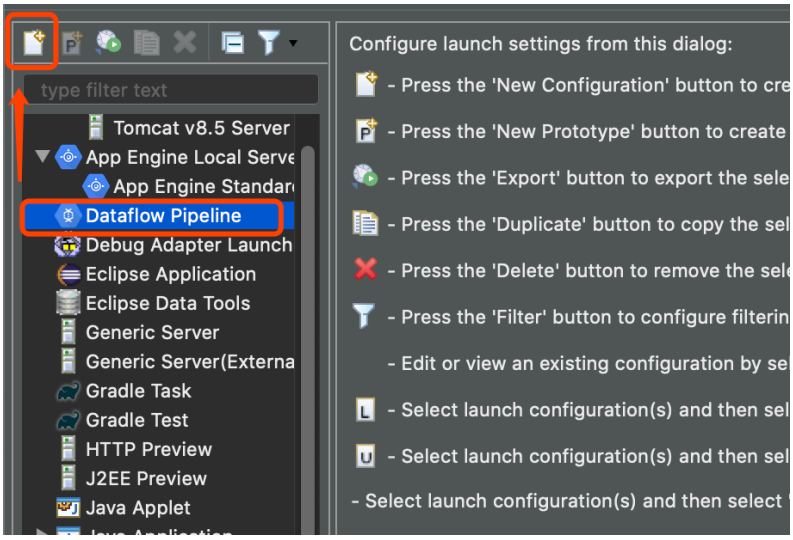
(4) Select "File"->"New"->"Project", and in "Google Cloud Platform" directory select "Cloud Dataflow





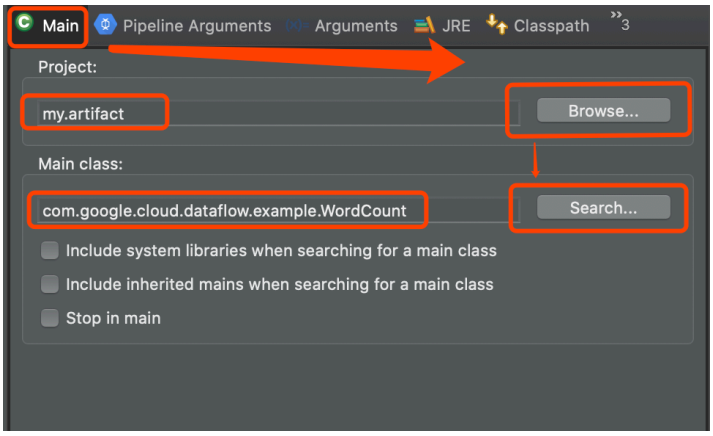
Java Project". Complete configurations like Group ID and click "Next". Fill in the account and cloud platform project ID. Then, enter a valid staging location and click "Create Bucket", which should

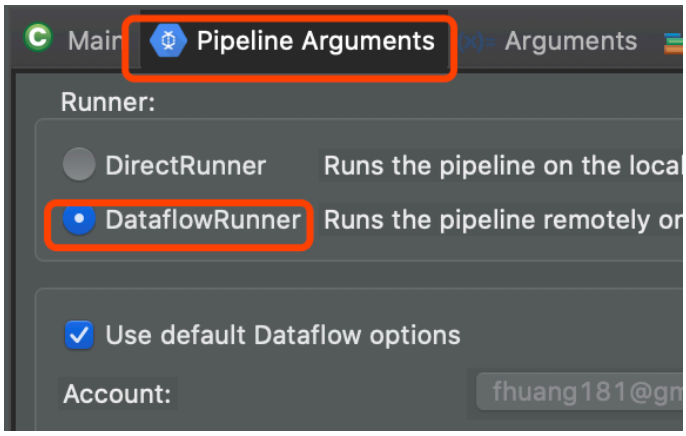




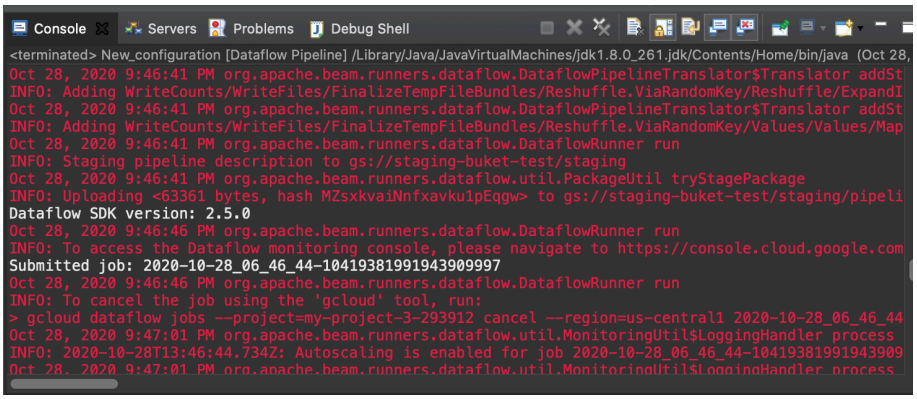
f ollo
w the official restrictions.[11] After that, click “Browse” to choose the account key generated in the step 4. Finally, click “Finish” to generate a “WordCount” sample.

(5) Run the “WordCount” example pipeline on the Dataflow service. Select “Run as”->”Run Configurations”, then choose “Dataflow Pipeline” and click “New Launch Configuration”





(6) In the “Main” tab, click “Browse” to select the project “my.artifact”. Click “Search” to select the “WordCount Main Type”. In the “Pipeline Arguments” tab, choose “DataflowRunner”. In the



“Arguments” tab, enter [-output=gs://staging-buket-test/staging/output-file-prefix] into “Program arguments” box to set the output to developers’ Cloud Storage Staging Location. Click “Run” to complete initialization.

3.7.Evaluation

(1) Advantage of Google Cloud Platform

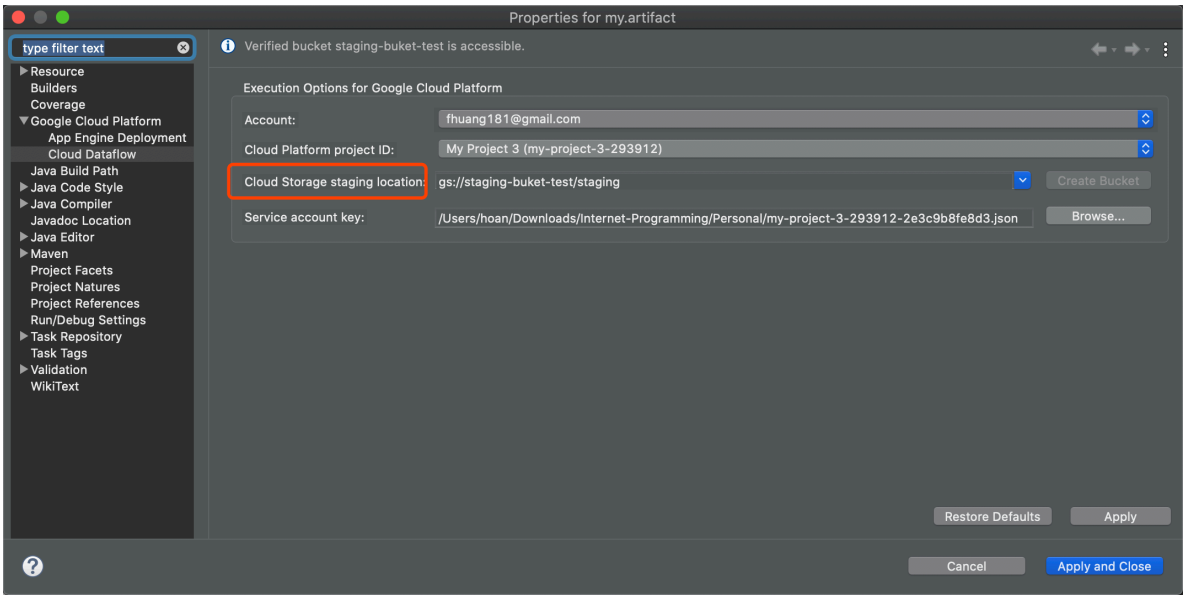
- It greatly reduces the learning cost of server-side knowledge for developers. It also improves the development efficiency and shortens the development cycle.
- It has high security level provided by Google.
- It provides a variety of service interfaces, from the most basic server interfaces to data analysis tools and even to the application deployment of machine learning methods. At the same time, there are a large number of the latest auxiliary tutorials, so that beginners can easily access the most complete mainstream technology, which helps the developers to obtain skills and experience faster and more comprehensive.
- It provides a team collaboration tool which is easy to manage and learn. It is helpful for the teams and their the developments of the projects.
- It integrates most of the server-side basic services, which enable developers to quickly follow up the development of the project after switching devices.
- Most services are charged according to internet traffic but not the time periods, which provides a cost-effective choice for some developers.
- Due to the high coupling between services in cloud platform, developers will be capable to learn and master the whole platform and stacks much more quickly.

(2) Disadvantages of Google Cloud Platform

- A high degree of integration of tools means that projects launched on this platform have a high dependence on the platform. However, sophisticated and experimental projects require developers to have a very accurate understanding and control of the overall project. In this case, it is not a good choice to choose a cloud platform with high integration
- The high degree of integration of the services provided by the system and the high coupling between the components means that the services provided by the system will have higher learning costs in the initial stage. Moreover, as an emerging platform still under development and iteration, all the specific implementation details of the cloud platform are constantly changing. The most reliable information source is the only official website. When a bug shows up when developing the project, It takes more time to spot the mistake source. Consequently, it require the developers to search and read more possible solutions, not to mention that there are not many valuable information that in a perfect correspondence with the problem, which greatly increases the difficulty of debugging.

3.8.Error analysis and solution

(1) Dataflow bucket error: “Missing object or bucket in path: 'gs://staging-buket-test/', did you mean: ‘gs://some-bucket/staging-buket-test'?"



Solution could be find in "Dataflow bucket creation #3216" in the github issues page. If the developer specify a subfolder and use [gs://staging-buket-test/staging] but not the recommended [gs://staging-buket-test/staging], the error would be fixed.

(2) In "3.5.Build and run "Hello World" app by the console of Google Cloud Platform", if use same cloud shell to deploy after successfully run in the 8080 port, it would pop up plugin error and could not deploy successfully.

Solution is that open another cloud shell for "My Project 2" and deploy in that shell.

3.9.Tips

(1) Try to use the guidance from official website of cloud platform is a precise and convenient start.

(2) Have the latest version of Could Tools for eclipse is important. Developers could check through this page.

(3) Make sure that the project has the App Engine Project facet. if not, developers could right click the project and then select "Configure"->"Convert to App Engine Project".[8]

(4) Use command line to manage GCP projects could be a very easy and fast way, developers could take the reference 9 to install gcloud.

(5) The Dataflow Eclipse plugin works only with the Dataflow SDK distribution versions 2.0.0 to 2.5.0. It does not work with the Apache Beam SDK distribution. [10]

4. Cloud platforms suitable for open source project

(1) Mainstream cloud platform providers:

Open Source Projects	Google Cloud Platform	Amazon Web Service	Microsoft Azure
Github	3K repositories / 44K Issues	3K repositories / 31K Issues	5K repositories / 247K Issues
Official Repositories	798	243	6679
Awesome Open Source	95	959	313

(2) Other platform providers:

Open Source Projects	Heroku	IBM Cloud	Alibaba Cloud
Github	84K repositories / 293K Issues	29K repositories / 20K Issues	5K repositories / 15K Issues
Official Repositories	822	339	388
Awesome Open Source	193	21	48

(3) Summary

The repository and issue number in github stands for the popularity of the cloud platform among the open source project developers, but does not directly indicate the if it is suitable for project development. The official repository numbers of platforms show how much do platform contribute to the open source community, which is a good indicator to illustrate the official concentration of platforms. The project number in "Awesome Open Source" website also helps to show the works and projects developers have already done in the specific platform, which could help to illustrate the community scale and strength.

So, it is clear to conclude that the Microsoft Azure and Heroku hold the best popularity among developers so far, and the Amazon is the most mature platform for open source project. However, different service providers start from the different time and process different features and strength in many areas. Each developer has his/her specific condition and develop requirements, so there is no the best platform but only the most suitable one.

5. Cloud platforms suitable for JAVA web application

Mainstream cloud platform providers:

Java Web App	Google Cloud Platform	Amazon Web Service	Microsoft Azure
IDE Plugin	Eclipse / IntelliJ	IntelliJ / VS Code / Eclipse	IntelliJ / VS Code / Eclipse
Environment Support	Java 8 / Java 11 / Flexible environment	Java 8 or higher / Tomcat 8 or higher	Java 8 or higher
User friendly Interfaces	Web-based or Chrome-app-based graphical interface	Web-based graphical interface	Web-based graphical interface
Basic instance	8 GB of RAM / 2 vCPUs	8 GB of RAM / 2 vCPUs	8 GB of RAM / 2 vCPUs
Largest Capability	3.75 TB of RAM / 160vCPUs	3.84 TB of RAM / 128vCPUs	3.89 TB of RAM / 128 vCPUs
Pricing per month [12]	Elastic price on traffic ; Basic $52/month ; Largest instance $5.32/hour	Basic $69/month ; Largest instance $3.97/hour	Basic $70/month ; Largest instance $6.79/hour

In summary, for the independent developers, the Azure and GCP are both good choice. Because the former has the biggest open source community and user foundation, and the latter has a more customer-friendly pricing plans with various toolkits. But for large teams or corporations, the Amazon is the better choice because it owns the biggest number of services and the largest available zones. All the platform above are suitable for Java web application, but there must be the best fitted one for each developer.

6. Conclusion

As the techniques and market environment keeping evolving, the requirements for developers are also becoming more and more complex and effective. The iteration of IDE and development toolkit put the requirements even harder for developers to accomplish by their own efforts. So, it is wise and productive for developers to use Cloud Platforms to help accelerate their development of the projects.

Among various kinds of Cloud Platform, the investigation between three mainstream service provider is conducted in the second part of this paper. In the third part, the detailed demonstration of how to deploy java web applications in the Google Cloud Platform would help the developers move the first step into serverless programming. In addition, the provided three kinds of deployment would make developer be familiar with all the basic interaction key points using GCP, which provides a comprehensive tutorial for beginners to have a correct quick-start (considering the official document does not contain all the expected errors may have happened). After that, the evaluation of GCP makes the analysis about the advantages and disadvantages from the perspective of developers, which is valuable for readers and other developer obtain a more specific analysis viewpoint. At the end of the third part, the error solution and tips play as the complement statement in case the developers encounter the same problem when deploying.

The discussions and analyses about the suitable cloud platform for open source project and Java web application are demonstrated in the fourth and fifth part of this paper. These two parts would help the developers to have a comprehensive understanding about the development status and prospect of different could computing platforms. In this part, developers could find out which platform is the most suitable for themselves and their projects.

In conclusion, the cloud platform is prevailing and playing a more and more important role in the development process. It could be a nice try to learn and get familiar with the cloud computing development method for developers. However, it depends on specific situations and projects. Stay opened to new techniques and stay focused on the details and architecture.

7. Reference

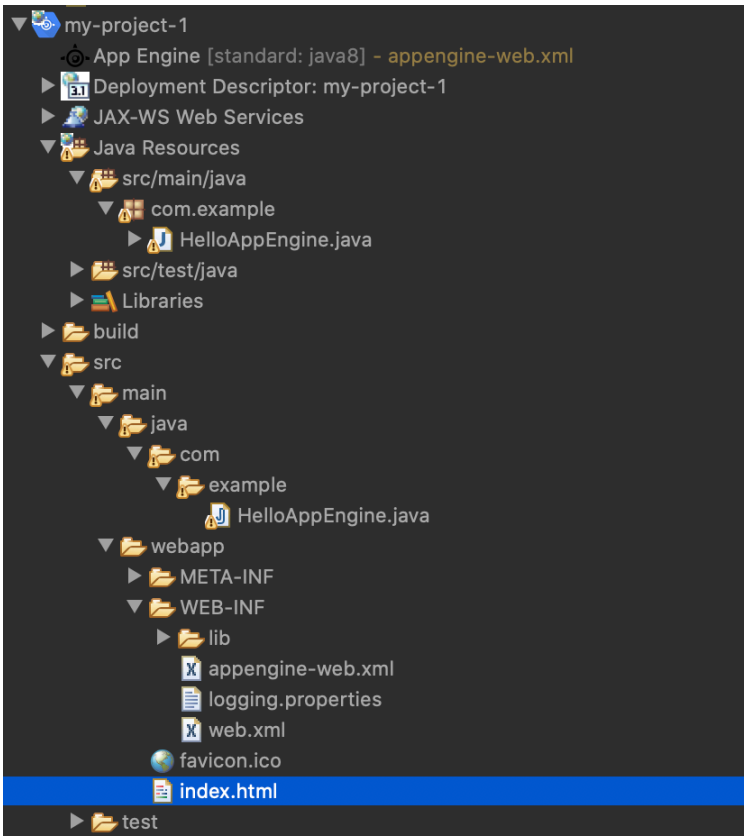
[1] Top 5 PaaS Solutions for Developing Java Applications. Site point. (URL: https://www.sitepoint.com/top-5-paas-solutions-developing-java-applications/)
[2] A List of Popular Cloud Hosting Providers for Java Web Applications. Abto Software. (URL: https://www.abtosoftware.com/blog/a-list-of-popular-cloud-hosting-providers-for-java-web-applications)
[3] How to choose the right cloud service provider. Future Mind. (URL: https://www.futuremind.com/blog/how-choose-right-cloud-service-provider)
[4] How to Create and Deploy a Java Web App to Google App Engine. wikiHow. (URL: https://www.wikihow.com/Create-and-Deploy-a-Java-Web-App-to-Google-App-Engine)
[5] Cloud Tools for Eclipse. Google Cloud. (URL: https://cloud.google.com/eclipse/docs/quickstart?hl=zh-cn)
[6] Creating an App Engine Standard Application. Google Cloud. (URL: https://cloud.google.com/eclipse/docs/creating-new-webapp)
[7] Running and Debugging an App Engine Standard Project Locally Inside Eclipse. Google Cloud. (URL: https://cloud.google.com/eclipse/docs/running-and-debugging)
[8] Deploying App Engine Standard Applications from Eclipse. Google Cloud. (URL: https://cloud.google.com/eclipse/docs/deploying)
[9] How to install gcloud SDK on the macOS and start managing GCP through CLI. Tapendra Dev. (URL: https://medium.com/@tapendradev/how-to-install-gcloud-sdk-on-the-macos-and-start-managing-gcp-through-cli-d14d2c3a8869)
[10] Setting up Cloud Dataflow in Eclipse. Google Cloud. (URL: https://cloud.google.com/eclipse/docs/quickstart-dataflow)
[11] Setting up Cloud Dataflow in Eclipse. Google Cloud. (URL: https://cloud.google.com/eclipse/docs/quickstart-dataflow)
[12] AWS vs Azure vs Google – Detailed Cloud Comparison. IntelliPaat. (URL: https://intellipaat.com/blog/aws-vs-azure-vs-google-cloud/)

8. Appendix

8.1.Core source code

(1) Java Web Application using Eclipse plug-in and run on Google Cloud Server

Program Structure:



HelloAppEngine.java

```
package com.example;

import java.io.IOException;

import org.junit.Assert;
import org.junit.Test;

public class HelloAppEngineTest {

  @Test
  public void test() throws IOException {
    MockHttpServletResponse response = new MockHttpServletResponse();
    new HelloAppEngine().doGet(null, response);
    Assert.assertEquals("text/plain", response.getContentType());
    Assert.assertEquals("UTF-8", response.getCharacterEncoding());
    Assert.assertEquals("Hello App Engine!\r\n", response.getWriterContent().toString());
  }
}
```

index.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
  <head>
    <meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8" />
    <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>
```

web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
         version="3.1">
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">

  <threadsafe>true</threadsafe>
  <sessions-enabled>false</sessions-enabled>
  <runtime>java8</runtime>
```

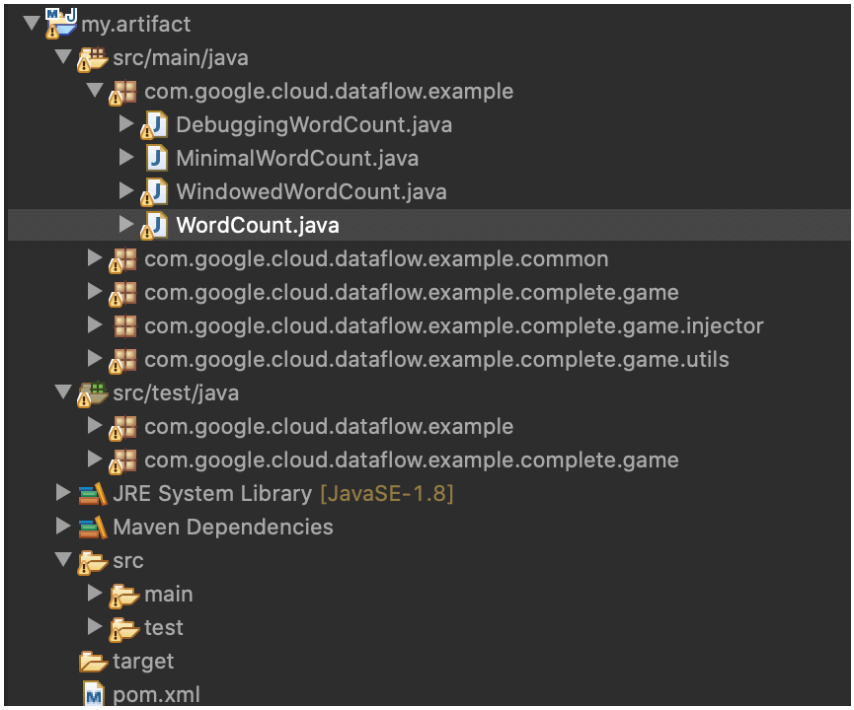
(2) Java Web Application generated and deployed directly on Google Cloud Server

- remove the previous files before cloning
  $ rm -rf appengine-try-java
- in cloud shell clone the existed project
  $ git clone \ https://github.com/GoogleCloudPlatform/appengine-try-java
- switch to the project file
  $ cd appengine-try-java

- run the project
  $ mvn appengine:run
- deploy the app
  $ gcloud app create
  $ gcloud config set project \ my-project-2-293913
  $ mvn appengine:deploy

(3) Dataflow generate in Eclipse and deployed on Google Cloud Server and Cloud Storage

Program Structure:



WordCount.java

```
/*
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements.  See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership.  The ASF licenses this file
 * to you under the Apache License, Version 2.0 (the
 * "License"); you may not use this file except in compliance
 * with the License.  You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package com.google.cloud.dataflow.example;

import com.google.cloud.dataflow.example.common.ExampleUtils;
import org.apache.beam.sdk.Pipeline;
import org.apache.beam.sdk.io.TextIO;
```

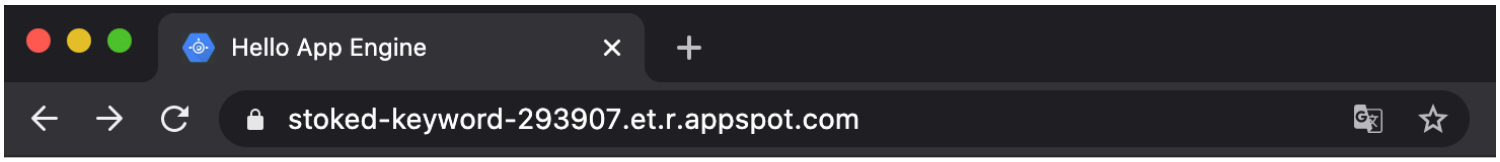
```
import org.apache.beam.sdk.metrics.Counter;
import org.apache.beam.sdk.metrics.Distribution;
import org.apache.beam.sdk.metrics.Metrics;
import org.apache.beam.sdk.options.Default;
import org.apache.beam.sdk.options.Description;
import org.apache.beam.sdk.options.PipelineOptions;
import org.apache.beam.sdk.options.PipelineOptionsFactory;
import org.apache.beam.sdk.options.Validation.Required;
import org.apache.beam.sdk.transforms.Count;
import org.apache.beam.sdk.transforms.DoFn;
import org.apache.beam.sdk.transforms.MapElements;
import org.apache.beam.sdk.transforms.PTransform;
import org.apache.beam.sdk.transforms.ParDo;
import org.apache.beam.sdk.transforms.SimpleFunction;
import org.apache.beam.sdk.values.KV;
import org.apache.beam.sdk.values.PCollection;

public class WordCount {

  static class ExtractWordsFn extends DoFn<String, String> {
   private final Counter emptyLines = Metrics.counter(ExtractWordsFn.class, "emptyLines");
   private final Distribution lineLenDist = Metrics.distribution(
       ExtractWordsFn.class, "lineLenDistro");

   @ProcessElement
   public void processElement(@Element String element, OutputReceiver<String> receiver) {
    lineLenDist.update(element.length());
    if (element.trim().isEmpty()) {
     emptyLines.inc();
    }

    // Split the line into words.
    String[] words = element.split(ExampleUtils.TOKENIZER_PATTERN, -1);

    // Output each word encountered into the output PCollection.
    for (String word : words) {
     if (!word.isEmpty()) {
      receiver.output(word);
     }
    }
   }
  }

  public static class FormatAsTextFn extends SimpleFunction<KV<String, Long>, String> {
   @Override
   public String apply(KV<String, Long> input) {
    return input.getKey() + ": " + input.getValue();
   }
  }
```

```
 public static class CountWords extends PTransform<PCollection<String>,
    PCollection<KV<String, Long>>> {
  @Override
  public PCollection<KV<String, Long>> expand(PCollection<String> lines) {

    // Convert lines of text into individual words.
    PCollection<String> words = lines.apply(
       ParDo.of(new ExtractWordsFn()));

    // Count the number of times each word occurs.
    PCollection<KV<String, Long>> wordCounts = words.apply(Count.perElement());

    return wordCounts;
  }
 }
public interface WordCountOptions extends PipelineOptions {

  @Description("Path of the file to read from")
  @Default.String("gs://apache-beam-samples/shakespeare/kinglear.txt")
  String getInputFile();
  void setInputFile(String value);

  /**
   * Set this required option to specify where to write the output.
   */
  @Description("Path of the file to write to")
  @Required
  String getOutput();
  void setOutput(String value);
 }

 static void runWordCount(WordCountOptions options) {
  Pipeline p = Pipeline.create(options);
 p.apply("ReadLines", TextIO.read().from(options.getInputFile()))
   .apply(new CountWords())
   .apply(MapElements.via(new FormatAsTextFn()))
   .apply("WriteCounts", TextIO.write().to(options.getOutput()));

  p.run().waitUntilFinish();
 }

 public static void main(String[] args) {
  WordCountOptions options = PipelineOptionsFactory.fromArgs(args).withValidation()
    .as(WordCountOptions.class);

  runWordCount(options);
 }
}
```

(4) Manage and shutdown the application in Google Cloud Platform
Go to sidebar in the left, select “App Engine”-> “Dashboard” to watch the project status.
Select “IAM & Admin” in the sidebar, choose “Settings”->”Shut Down” to shutdown the project.

8.2.Samples

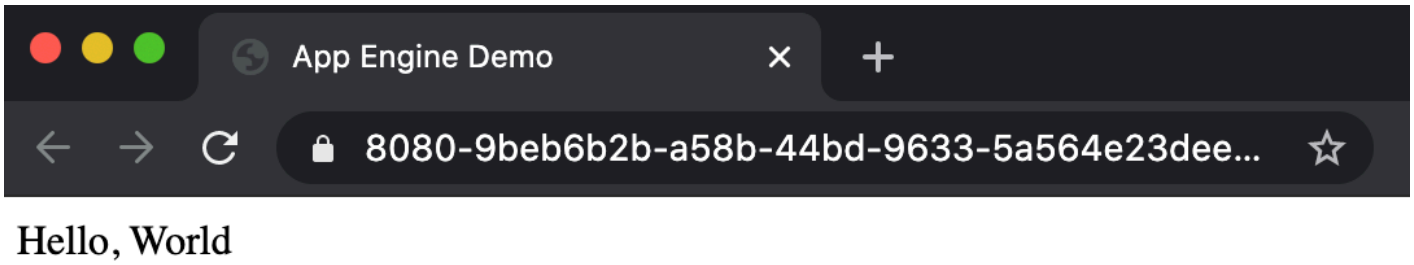
(1) Java Web Application using Eclipse plug-in and run on Google Cloud Server
Could be visited in: https://stoked-keyword-293907.et.r.appspot.com/



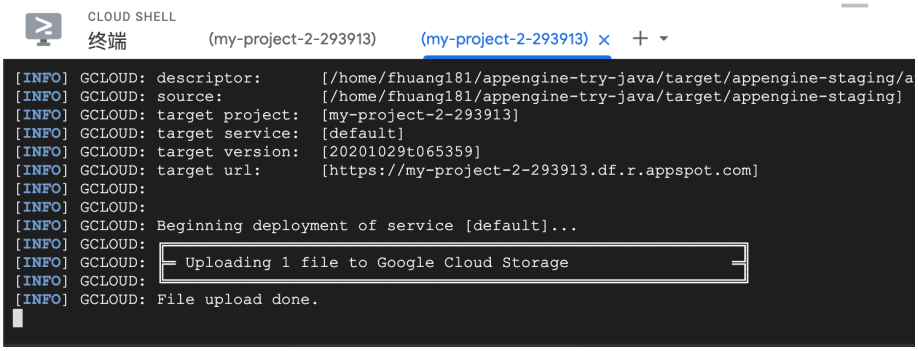
Hello App Engine!

Available Servlets:
The servlet

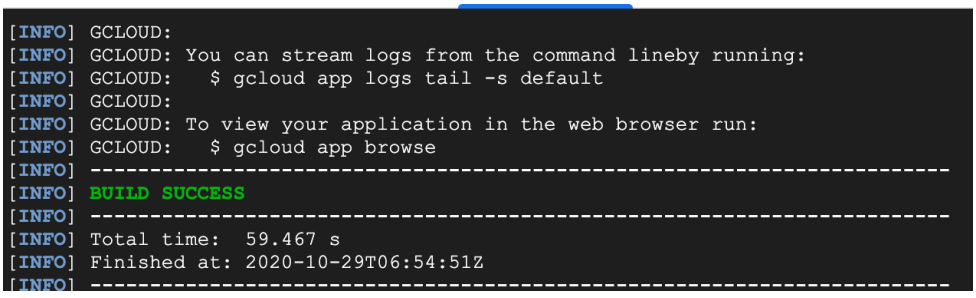
(2) Java Web Application generated and deployed directly on Google Cloud Server
Preview after run the “mvn appengine:run”



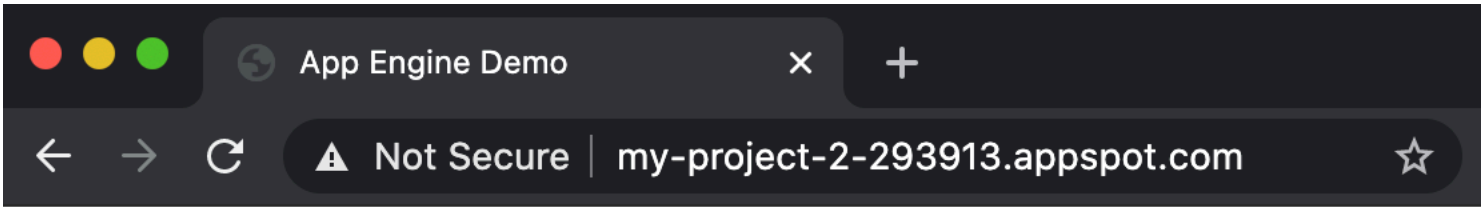
Deploy the app:



Build successfully:

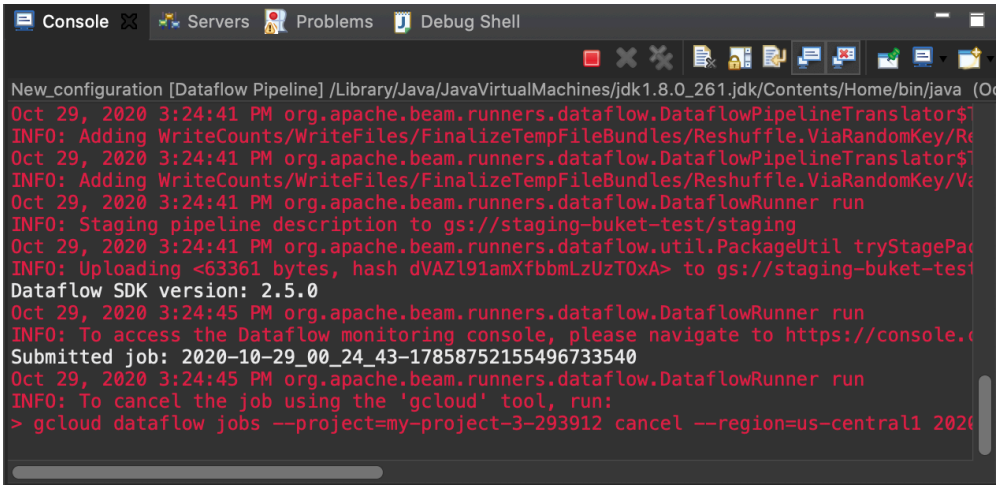


Could be visited in: http://my-project-2-293913.appspot.com/

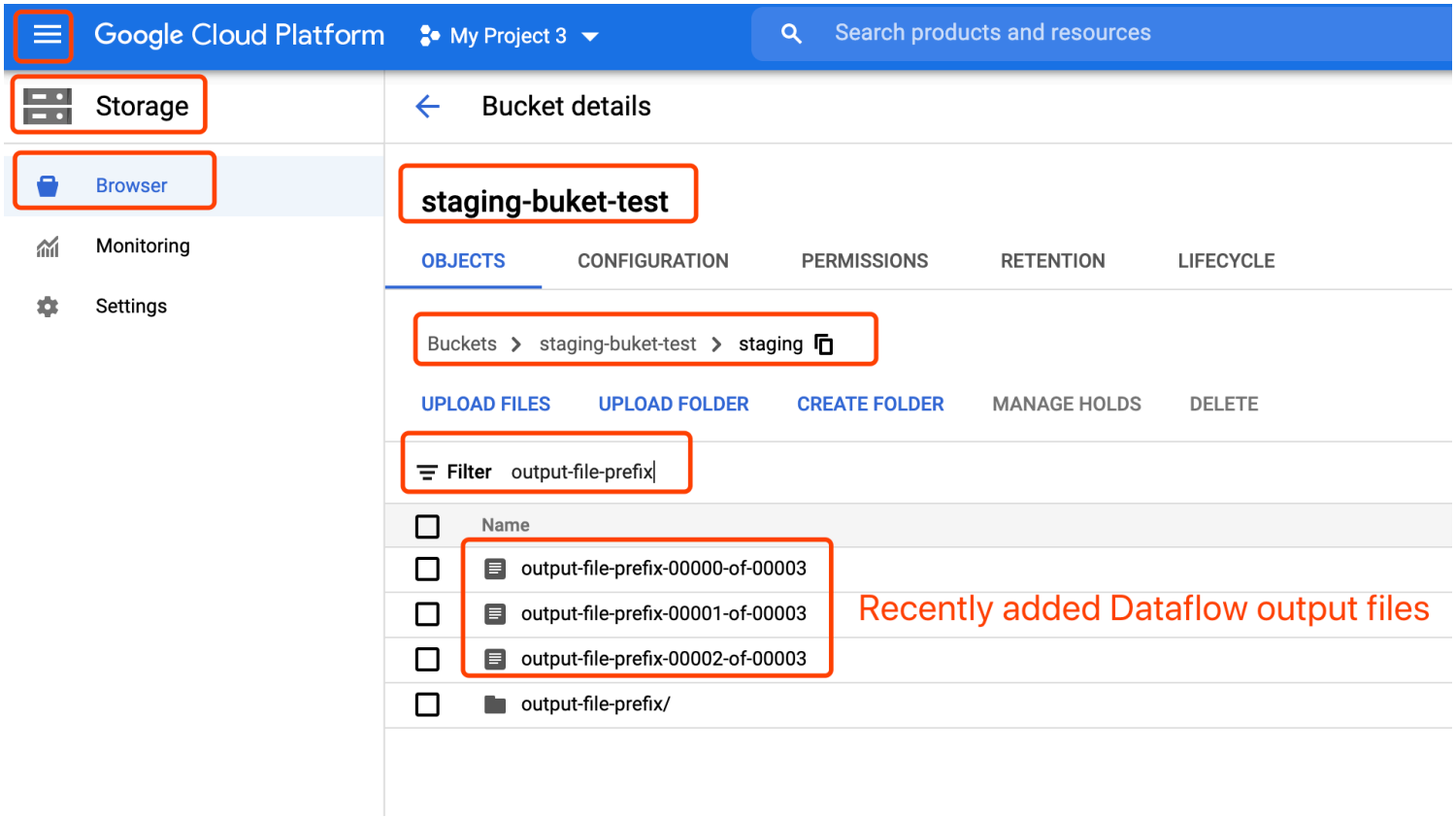


Hello, World

(3) Dataflow generate in Eclipse and deployed on Google Cloud Server and Cloud Storage Successfully run with console output in eclipse:



In Google Cloud Platform, go to sidebar and choose “Storage”->”staging-buket-test”->”staging”, then search “output-file-prefix”. The files would show on the page.



(4) Manage and shutdown the application in Google Cloud Platform