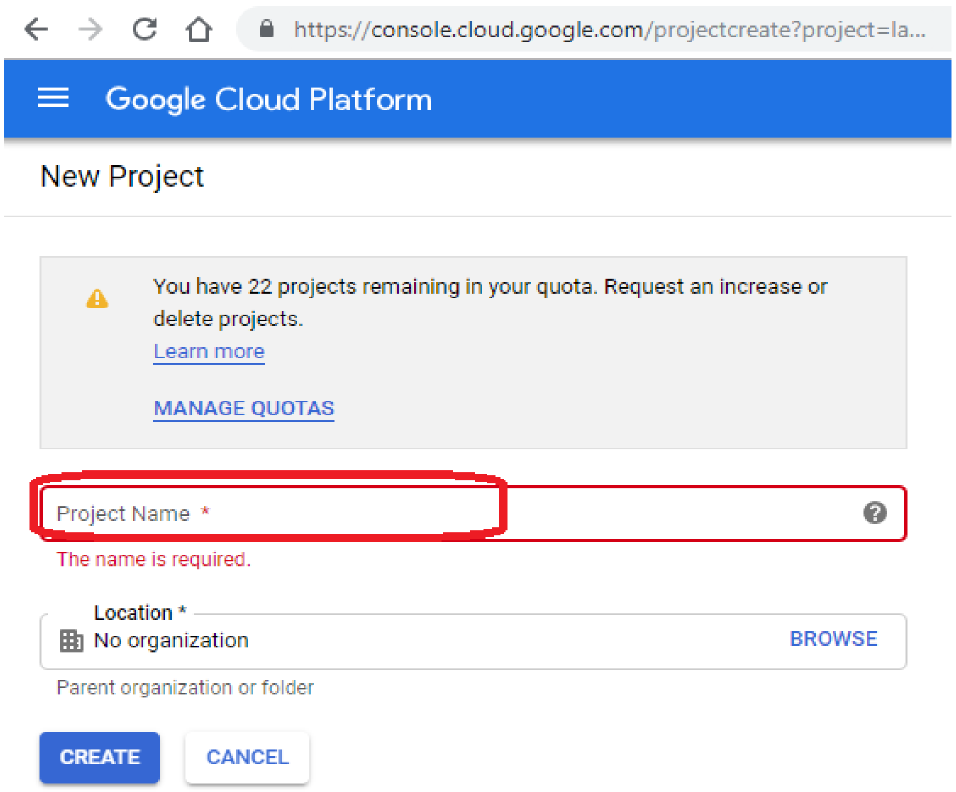
Lab 11, Part 1: Hello World Website

In this lab, we will create a couple of websites using the Google Cloud Platform (GCP) App Engine. In part 1, we will make a website that displays some text.[1] This will prepare us for part 2, where we will use Java Server Pages (JSP) to make a website that displays an image.

1. Navigate to the following page: https://console.cloud.google.com/projectcreate. In the project name field, enter "<4x4>-lab11-part1" without quotation marks. Replace <4x4> with your OUNet ID. For instance, if your ID was abcd1234, then the project name would be "abcd1234-lab11-part1".


https://console.cloud.google.com/projectcreate?project=la...

Google Cloud Platform

New Project

You have 22 projects remaining in your quota. Request an increase or delete projects.
Learn more

MANAGE QUOTAS

Project Name *

The name is required.

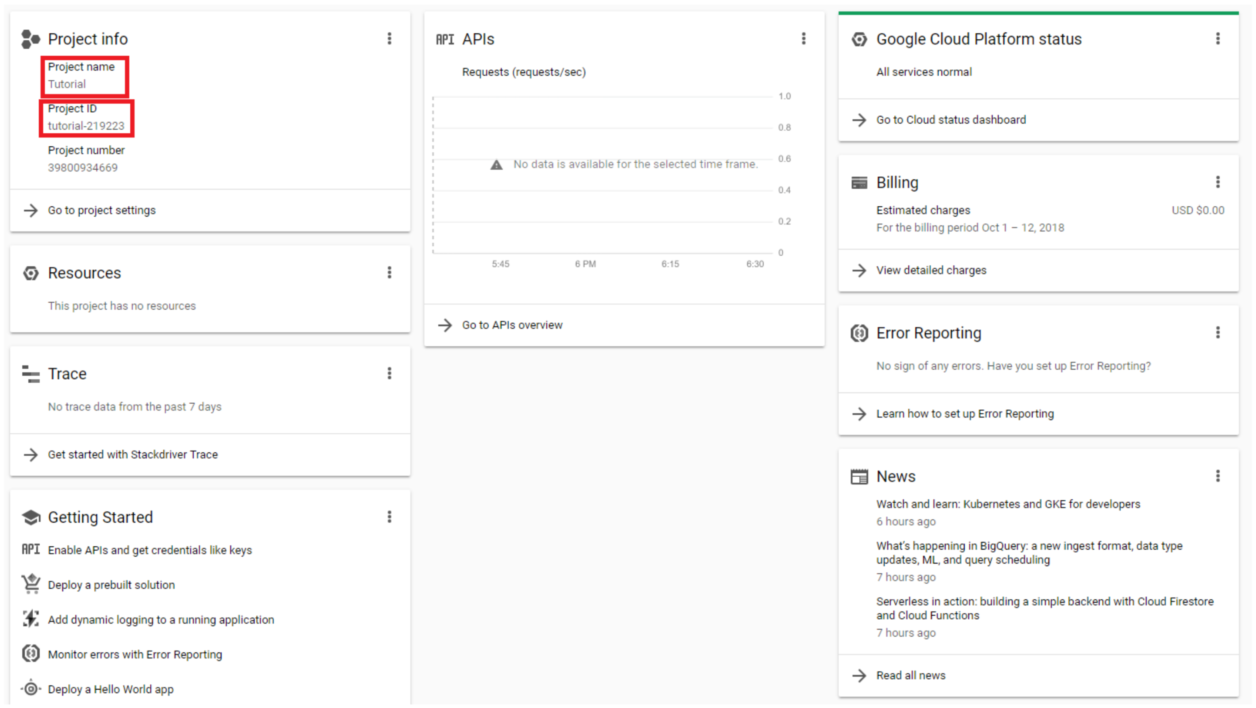
Location *
No organization BROWSE

Parent organization or folder

CREATE CANCEL

Click the Create button to create a new GCP project. This will bring you to the GCP dashboard, which looks like the image below.

[1] The instructions in this document are based on the App Engine Quickstart tutorial available here.



Project info

Project name
Tutorial

Project ID
tutorial-219223

Project number
39800934669

Go to project settings

Resources

This project has no resources

Trace

No trace data from the past 7 days

Get started with Stackdriver Trace

Getting Started

Enable APIs and get credentials like keys

Deploy a prebuilt solution

Add dynamic logging to a running application

Monitor errors with Error Reporting

Deploy a Hello World app

APIs

Requests (requests/sec)

No data is available for the selected time frame.

Go to APIs overview

Google Cloud Platform status

All services normal

Go to Cloud status dashboard

Billing

Estimated charges USD $0.00

For the billing period Oct 1 – 12, 2018

View detailed charges

Error Reporting

No sign of any errors. Have you set up Error Reporting?

Learn how to set up Error Reporting

News

Watch and learn: Kubernetes and GKE for developers
6 hours ago

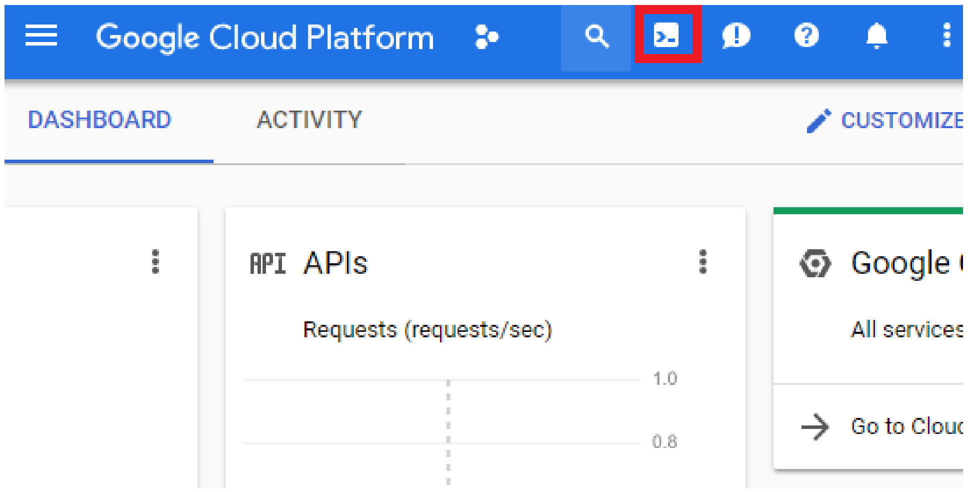
What's happening in BigQuery: a new ingest format, data type updates, ML, and query scheduling
7 hours ago

Serverless in action: building a simple backend with Cloud Firestore and Cloud Functions
7 hours ago

Read all news

The project name on the info card (outlined in red above) should match the name you entered on the previous page. If the name is different, click the dropdown menu in the top-left corner (after the text "Google Cloud Platform" on the blue bar) and select the correct name.

2. Open the Cloud Shell by clicking the icon outlined in the following picture:[2]



3. Execute the command `ls` to list the contents of your cloud directory. If the list includes the directory appengine-try-java, delete it with the following command:[3]

```
rm -rf appengine-try-java
```

[2] In addition to accessing the shell from a browser, you can also use a terminal on your computer by first installing the Cloud SDK: https://cloud.google.com/sdk/docs/install.

[3] Text can be pasted into the shell by typing Shift+Insert.

4. Now execute the command below to clone the Hello World App Engine example from GitHub. Notice that this is the same command we have used throughout the semester, but now we are using it to clone a GitHub repo in the cloud.

   ```
   git clone https://github.com/GoogleCloudPlatform/appengine-try-java
   ```

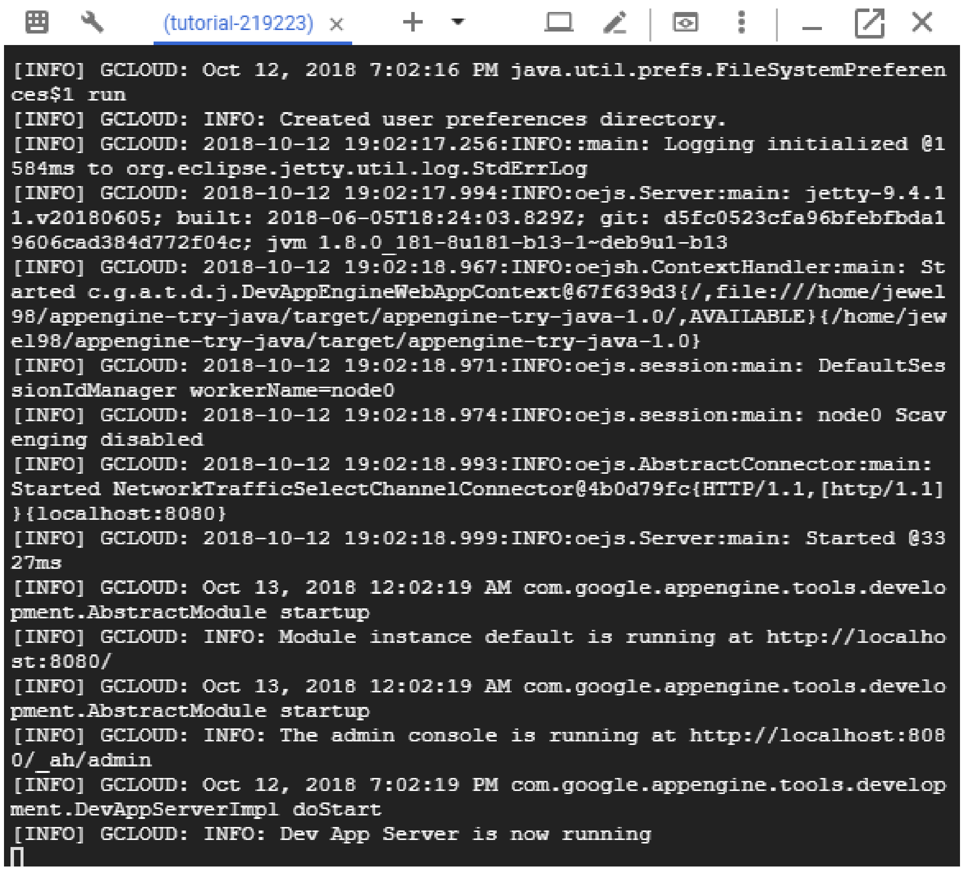
5. Change the working directory to the directory created by the clone command:

   ```
   cd appengine-try-java
   ```

6. List the contents of the new directory. Notice that the list contains a file named "pom.xml." This indicates that the application is configured to work with Apache Maven, a popular build automation tool.[4] To compile and run the program, execute the next command:

   ```
   mvn appengine:run
   ```

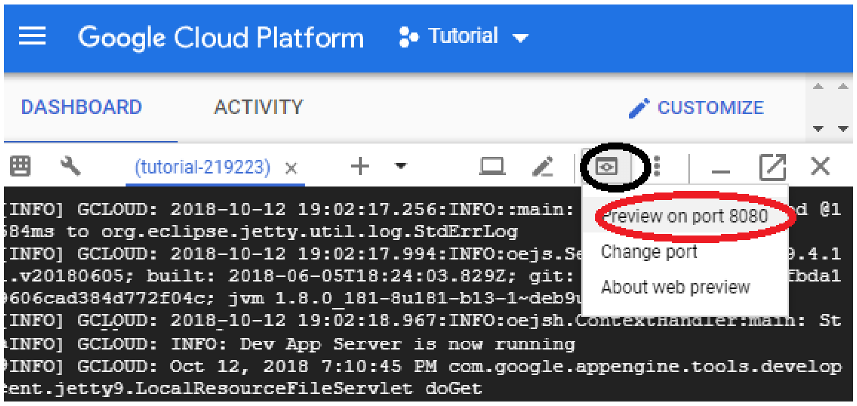
   This will produce a series of output statements like those shown in the image below. The final statement should indicate that the application is running.



```
[INFO] GCLOUD: Oct 12, 2018 7:02:16 PM java.util.prefs.FileSystemPreferen
ces$1 run
[INFO] GCLOUD: INFO: Created user preferences directory.
[INFO] GCLOUD: 2018-10-12 19:02:17.256:INFO::main: Logging initialized @1
584ms to org.eclipse.jetty.util.log.StdErrLog
[INFO] GCLOUD: 2018-10-12 19:02:17.994:INFO:oejs.Server:main: jetty-9.4.1
1.v20180605; built: 2018-06-05T18:24:03.829Z; git: d5fc0523cfa96bfebfbda1
9606cad384d772f04c; jvm 1.8.0_181-8u181-b13-1~deb9u1-b13
[INFO] GCLOUD: 2018-10-12 19:02:18.967:INFO:oejsh.ContextHandler:main: St
arted c.g.a.t.d.j.DevAppEngineWebAppContext@67f639d3{/,file:///home/jewel
98/appengine-try-java/target/appengine-try-java-1.0/,AVAILABLE}{/home/jew
el98/appengine-try-java/target/appengine-try-java-1.0}
[INFO] GCLOUD: 2018-10-12 19:02:18.971:INFO:oejs.session:main: DefaultSes
sionIdManager workerName=node0
[INFO] GCLOUD: 2018-10-12 19:02:18.974:INFO:oejs.session:main: node0 Scav
enging disabled
[INFO] GCLOUD: 2018-10-12 19:02:18.993:INFO:oejs.AbstractConnector:main:
Started NetworkTrafficSelectChannelConnector@4b0d79fc{HTTP/1.1,[http/1.1]
}{localhost:8080}
[INFO] GCLOUD: 2018-10-12 19:02:18.999:INFO:oejs.Server:main: Started @33
27ms
[INFO] GCLOUD: Oct 13, 2018 12:02:19 AM com.google.appengine.tools.develo
pment.AbstractModule startup
[INFO] GCLOUD: INFO: Module instance default is running at http://localho
st:8080/
[INFO] GCLOUD: Oct 13, 2018 12:02:19 AM com.google.appengine.tools.develo
pment.AbstractModule startup
[INFO] GCLOUD: INFO: The admin console is running at http://localhost:808
0/_ah/admin
[INFO] GCLOUD: Oct 12, 2018 7:02:19 PM com.google.appengine.tools.develop
ment.DevAppServerImpl doStart
[INFO] GCLOUD: INFO: Dev App Server is now running
```

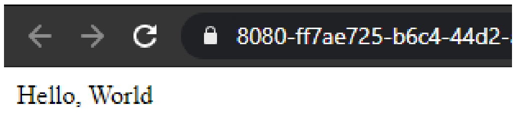
7. The application is a server program that will send text to a web browser. To preview the text, click the icon circled in the following picture and select "Preview on port 8080":

[4] The pom file contains configuration details needed to build the project, such as required libraries (e.g., JUnit).



```
[INFO] GCLOUD: 2018-10-12 19:02:17.256:INFO::main:                            d @1
584ms to org.eclipse.jetty.util.log.StdErrLog
[INFO] GCLOUD: 2018-10-12 19:02:17.994:INFO:oejs.Se                           9.4.1
.v20180605; built: 2018-06-05T18:24:03.829Z; git:                             fbda1
9606cad384d772f04c; jvm 1.8.0_181-8u181-b13-1~deb9u
[INFO] GCLOUD: 2018-10-12 19:02:18.967:INFO:oejsh.ContextHandler:main: St
[INFO] GCLOUD: INFO: Dev App Server is now running
[INFO] GCLOUD: Oct 12, 2018 7:10:45 PM com.google.appengine.tools.develop
ent.jetty9.LocalResourceFileServlet doGet
```

A new tab will open that looks like this:

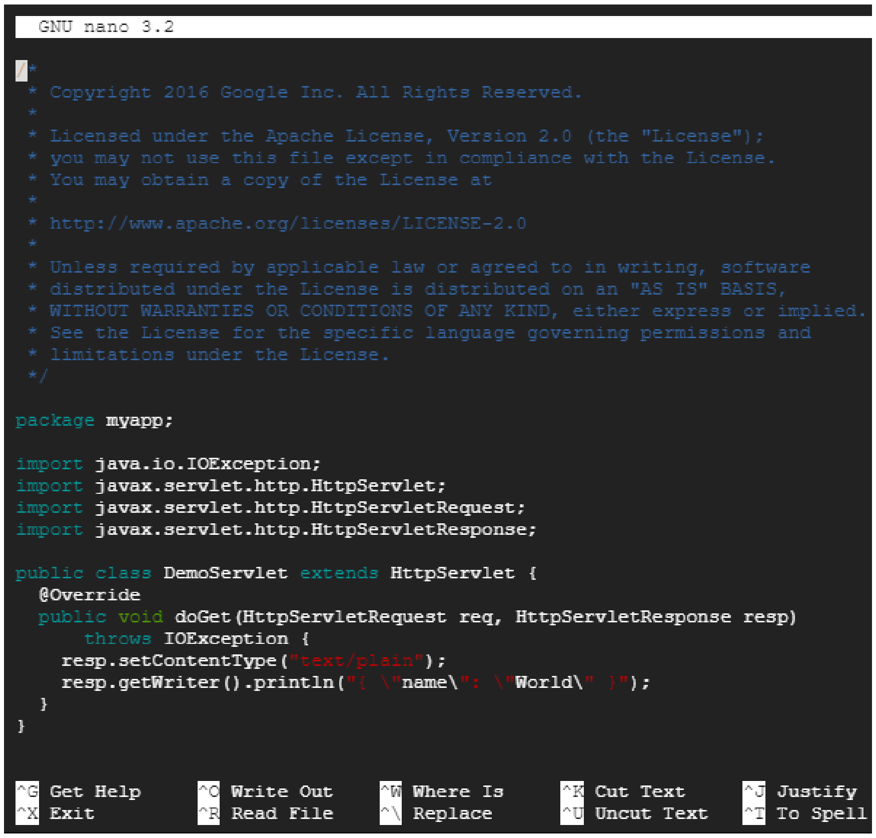


Hello, World

8. In the Cloud Shell, type Ctrl+C to terminate the program. Reload the preview tab and you will see an error that says the browser was unable to connect to the server on port 8080.

9. Now we will change the text that is output by the server. Open the source file for the program in the text editor nano using the command below. Notice that the file is located in a subfolder of the src directory, just like the Lab 10 source files.

```
nano src/main/java/myapp/DemoServlet.java
```



```
  GNU nano 3.2

/*
 * Copyright 2016 Google Inc. All Rights Reserved.
 *
 * Licensed 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 myapp;

import java.io.IOException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class DemoServlet extends HttpServlet {
  @Override
  public void doGet(HttpServletRequest req, HttpServletResponse resp)
      throws IOException {
    resp.setContentType("text/plain");
    resp.getWriter().println("{ \"name\": \"World\" }");
  }
}

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

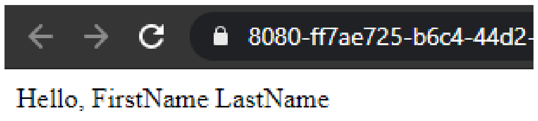
The editor should look like the picture above. If only part of the source code is visible, you can increase the size of the shell by dragging the horizontal bar that separates it from the GCP dashboard.

10. The text "Hello, World" that was displayed in the browser is specified in the source code by the string `"{ \"name\": \"World\" }"`. Replace `World` with your full name like this:

```
"{ \"name\": \"FirstName LastName\" }"
```

Type Ctrl+O and then Enter to save the file. Then type Ctrl+X to exit the editor.

11. Rerun the command from step 6 and then reload the preview tab. The browser should now show your name in place of the word "World," as in the following image:



Hello, FirstName LastName

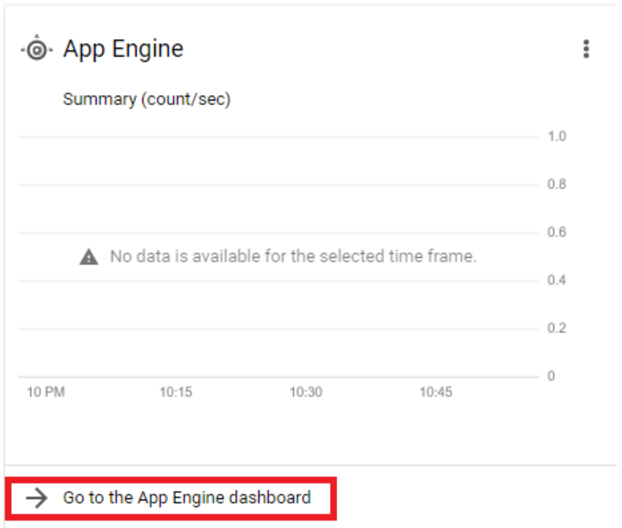
12. The last thing we need to do is deploy the app, which will make your server accessible to anyone in the world. Type Ctrl+C to terminate the program and then execute the three commands below in sequence.

```
gcloud app create
gcloud config set project <project-name>
mvn appengine:deploy
```

When you execute the first command, you will be asked to specify a region where your app will run on Google's servers. Select 14 for us-central.[5] When you execute the second command, replace `<project-name>` with the name from step 1.

The third command may take a few minutes to finish. Be patient, and you will eventually see the message "BUILD SUCCESS" in green font. You can now close the shell using the command `exit`.

13. Your new app can be monitored from the App Engine dashboard. Click the link labeled "Go to the App Engine dashboard" on the bottom of the App Engine card.



No data is available for the selected time frame.

[5] Your website can be accessed in any region, but us-central is the closest to Oklahoma.

14. In the top-right corner of the App Engine dashboard is a URL that ends with "appspot.com." You can use this URL to connect to your server from any browser, from any place on the planet. Submit this URL to Canvas along with the URL you create in part 2!

End of Part 1!