Announcements

- GCP demo today
 - Ensure you have redeemed Google credits through Ed discussion post instructions
 - And downloaded Google Cloud CLI for your respective OS and installed it
- GCP assignment out today
 - Based on deployment of a simple Java logger on the Google App Engine
 - Individual Assignment. Pretty much similar to class demo.
 - Due 6/17
- REST assignment due 6/10



CS3300 Introduction to Software Engineering

Lecture 6: Tools of the Trade #5

Google App Engine, Firebase, Google APIs

Dr. Nimisha Roy ► nroy9@gatech.edu

Contents

- Backend (SpringBoot) hosting using Google App Engine
 - Demo
- Frontend hosting using Firebase
 - Demo
- Google APIs
 - Google Maps API
 - Google Routes API
 - Google Places API
 - Authentication & Security APIs
 - Al and ML APIs

Google App Engine



Google App Engine - Intro

- Google's platform to build web application on Cloud, on a fully managed serverless platform. First Version in 2008
- Manages application platform that supports any framework, language or library, worries about the infrastructure so that you can focus on the code.
 - The infrastructure scales automatically with the number of requests
 - The complexities of server provisioning, maintenance, scaling, and security are abstracted away.
- Allows you to simply deploy your code, and the platform automates everything.
 - Being "fully managed" signifies that all backend tasks like server health checks, patch management, and failovers are automatically managed by the platform.



Google App Engine vs. Google Compute Engine

Compute Engine	App Engine
Delivered as Infrastructure as a Service (IAAS)	Delivered as Platform as a Service (PAAS)

laaS provides virtualized computing resources over the internet. It offers the infrastructure – virtual machines, storage, and networking – but leaves the rest up to the user.

PaaS provides a platform allowing customers to develop, run, and manage applications without dealing with the intricacies of building and maintaining the infrastructure.

- •Provider manages the virtualization, servers, hard drives, storage, and networking.
- •User manages applications, data, runtime, middleware, and OS.

- •Provider manages everything from the infrastructure to the software applications.
- Users manage the application and its data.



Google App Engine vs. Google Compute Engine

Compute Engine	App Engine
 Unmanaged Service by Google Cloud our responsibility to configure, administer, and monitor the system. Google's responsibility to ensure that resources are available, reliable, and ready to use 	 Managed Service by Google Cloud your focus should be on the application only Google will manage the resources needed to run the application.
offers the users complete control and flexibility over resources	Google manages all compute resources
Requires high expertise level, since everything needs to be installed and configured by yourself.	Very easy to use and deploy
Autoscaling is slower	Faster Autoscaling
Cheaper in the longer run and for large instances	Expensive in the longer run
Less secure than App Engine	More Secure



Google App Engine - Functions

- All infrastructure to deploy app on cloud- end to end management
- Scalability- acquires more instances automatically if the traffic of your application becomes higher
- monitoring, logging, versioning, debugging using google stack driver diagnostics
- Traffic splitting
- big data, storage, compute, connectivity support using google cloud
- Applications run in language specific sandboxes or in docker containers, depending on environment



App Engine Environments

- *Standard*: Run in language specific sandboxes
 - Complete isolation from OS/Disk/Other Apps
 - **V1**: Java, Python, PHP, Go (OLD versions). Restricted network access & libraries for some languages not Java
 - **V2**: Java, Python, PHP, Node.JS, Ruby, Go (NEWER Versions)
 - Pricing is based on instance hours.
- Flexible- Application instances run within Docker containers
 - Makes use of Compute engine virtual machines
 - Support ANY runtime (with built –in support for Python, Java, Node.js, Go, Ruby, PHP, or .NET)
 - Provides access to background processes and local disks
 - Pricing is dependent on memory and virtual CPU



Sustainability Considerations with Google Cloud Platform (GCP) and Google App Engine

- •Google's Commitment to Sustainability: Google Cloud has been operating on 100% renewable energy since 2017. All Google data centers, which power cloud services, match their electricity use with renewable energy purchases.
- •Energy-Efficient Data Centers: Google's data centers use custom-built servers and Al-driven cooling management systems that reduce energy consumption and optimize data center temperatures, further reducing energy use.
- •Google App Engine: As a fully managed, serverless platform, GAE maximizes resource utilization by dynamically allocating resources across multiple applications. This efficient use of shared infrastructure minimizes the environmental impact per user.
- •Auto-Scaling and Right-Sizing: Google Cloud services automatically scale up or down based on demand, ensuring minimal waste of computing resources and energy.
- •Location-Based Sustainability: GCP allows users to choose cloud regions powered by carbon-free energy sources. Selecting regions with lower carbon footprints can help reduce the overall environmental impact of cloud services.
- •Tools for Monitoring and Reducing Carbon Footprint: Google Cloud provides tools that help organizations measure their carbon footprint, allowing them to make informed decisions to reduce their impact.

App Engine Demo Time!

Deploy a HelloWorld application to Cloud using App Engine Standard

- Add GCP credits to your account.
- Add Billing Information
- Create a Project on Google Cloud
- Create a New Spring Boot Application using Spring Initializr.
- Create a Controller to test your application.
- Update the pom.xml file with the Google Cloud Tools Plugin.
- Create the app.yaml file to configure deployment.
- Build your application using Maven.
- Deploy to Google App Engine using Google Cloud CLI.
- Verify the deployment and manage billing settings.



Create a SpringBoot Application

0

For your assignment

. . . .



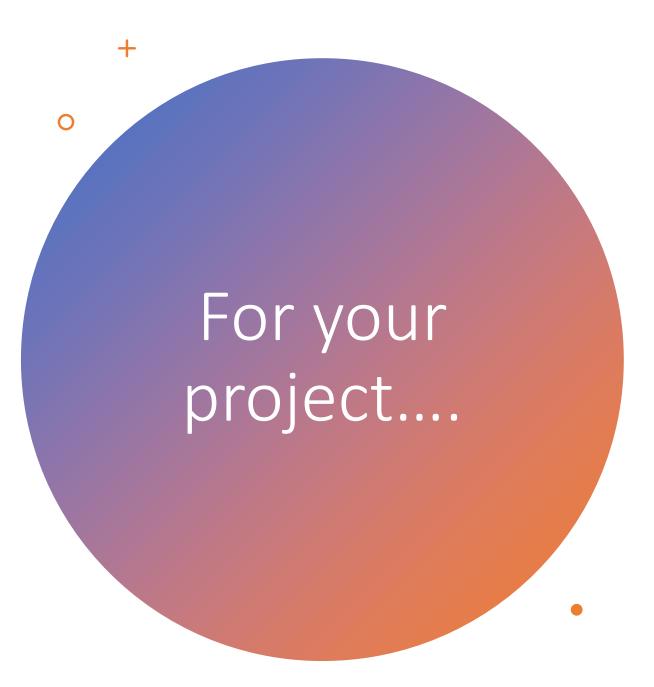
Get 2 endpoints – name and local time



Package project into a JAR



Deploy using app engine



Required for All Teams

- Backend logic must be deployed using Google App Engine (Standard or Flexible).
 - This includes REST APIs, server-side logic, or any dynamic computation.
- You may use Firebase Hosting for frontend (HTML, JS, Next.js, etc.).
- Vou may integrate with other Google APIs or Firebase services.

Help Documentation to monitor projects

Help
documentation about
setting alerts to monitor
projects in the Console

you can also use
the Cost Management
Resource Guide to learn
how to monitor and
manage costs in GCP.

Firebase

Why Use Firebase Hosting?

What is Firebase?

- A platform by Google that offers cloud-based tools for app development
- Provides services like static frontend hosting, authentication, databases, and storage
- Fully managed and integrates seamlessly with GCP projects

Why Use Firebase Hosting?

- •Blazing-fast static file hosting (HTML, CSS, JS, React, etc.)
- Free tier available perfect for student projects
- •Globally distributed CDN for low latency
- Automatic HTTPS secure by default
- Simple CLI-based deployment
 - •firebase init hosting
 - •firebase deploy
- Works great with backends on App Engine or Cloud Functions

Firebase Demo Time!

- Deploy an index.html application to Firebase
- Create a Firebase project (connect with GCP project)
- Initialize Firebase Hosting
- Add index.html or frontend code to public/Deploy using firebase deploy
- Use JavaScript to call backend at https://<your-app-id>.appspot.com/api/...
- Enable CORS

CORS

Problem: The Browser Blocks Cross-Origin Requests

- Web apps often consist of:
 - Frontend hosted on one domain (e.g., web.app)
 - Backend API on another (e.g., appspot.com)
- •Browsers enforce **same-origin policy** for security:
 - JavaScript can only call APIs from the same origin, unless the server allows otherwise.

Solution: CORS (Cross-Origin Resource Sharing)

- •CORS is a protocol that allows **controlled access** to resources from different origins.
- •The backend must include a header like:

```
Access-Control-Allow-Origin: https://your-project-id.web.app
```

What more can be done next!

You now have:

- A **Spring Boot backend** deployed on Google App Engine
- A Firebase-hosted frontend calling that backend

You can:

- Add Firebase Authentication
 - Let users sign in via email, Google, or GitHub
 - Protect access to certain backend endpoints
- Store Data in Firebase Firestore or Realtime Database
 - Save form input, comments, or app state
 - Sync data across users in real-time
- In Track Usage with Firebase Analytics
 - Monitor page views, button clicks, or custom events
- Dpload Files with Firebase Storage
 - Let users upload images, resumes, or PDFs
- Communicate Both Ways
 - Frontend (Firebase) stores data
 - Backend (Spring Boot on App Engine) reads or validates it via Firebase Admin SDK (optional)

Google APIs

Google APIs

- Developed by Google which allow communication with Google Services and their integration to other services.
- Examples include Search, Gmail, Translate or Google Maps.
- Third-party apps can use these APIs to take advantage of or extend the functionality of the existing services.
- Provides functionality like analytics, machine learning (the Prediction API) or access to user data (when permission to read the data is given).
- Usage of all of the APIs requires authentication and authorization using the Oauth 2.0 protocol.

Google Maps API

- Integrate, Embed, Query google maps in your application
- Create an interactive map
- Initiate actions like search directions
- Provide updated imagery to your users and help them find your location more easily
- Add markers, comments, icons to places of interest
- https://developers.google.com/maps/documentation
 available here to get started



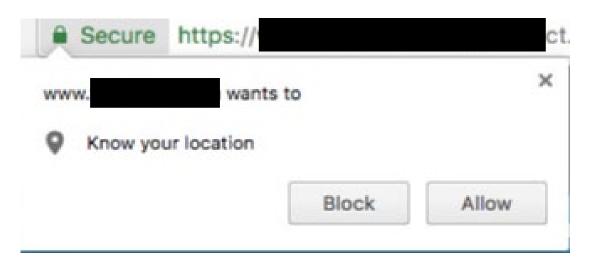
Google Routes API

- High quality directions and real-time traffic updates
- Compute travel times and distances for multiple destinations
- Precise routes for pedestrian, bikers and vehicle travels
- Enjoy snap to road benefits so you know exactly which route your asset is travelling along
- Receive speed limit information for each of those roads



Google Places API

- Access Location Data using coordinates, real-time signals, phone numbers
- Convert addresses to coordinates and vice-versa
- Get time zone, latitude, longitude etc.
- Engage your users further with contextual information about places
- Search for and receive information about local businesses, points of interests on every device with the Places autocomplete feature



Authentication

- API key required to use any Google API
- Only works in secure contexts (HTTPS)
- Generate your API key on Google Cloud Console-- API Manager--Enable API--Credentials- API key created

Authentication & Security APIs

Firebase Authentication (easy setup)

- •Email/password, Google login, anonymous users
- Built-in user management + UI

OAuth 2.0 (Google APIs)

- Needed for accessing user data like Gmail, Drive, Calendar
- User logs in and authorizes your app

Best Practices

- •Don't use unrestricted or overly-permissive API keys in frontend code. You must restrict them using Google Cloud Console settings.
- Use environment variables or Firebase config for secure key storage

Productivity APIs

🔢 Google Calendar API

- Add/view team events or reminders
- Sync with user's calendar (OAuth)

Google Sheets API

- Read/write spreadsheet data (e.g., survey results, leaderboard)
- Treat sheets like a lightweight backend

Custom Search API

- Add a personalized search box to your app
- Show curated Google results (with filters)

AI & ML APIs (No Model Training Required)

- Cloud Natural Language API
- Analyze text for sentiment, syntax, and named entities
- Cloud Vision API

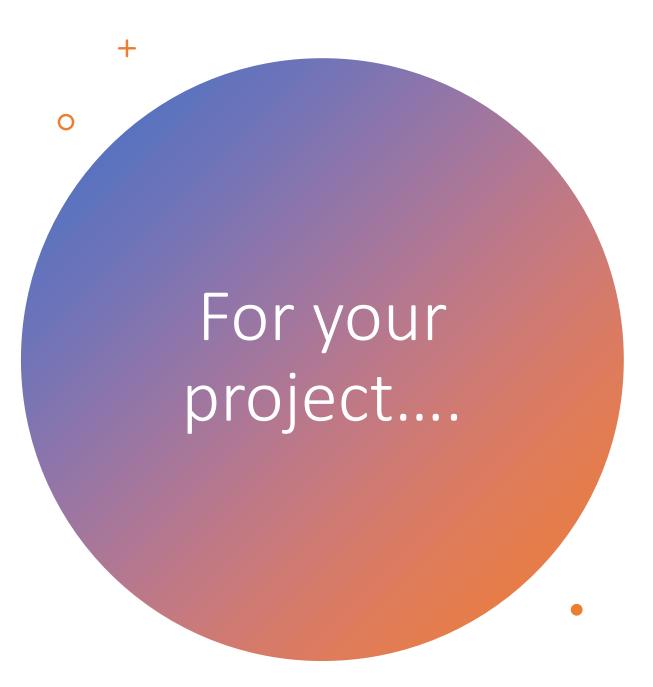
Detect labels, logos, text, and objects in images

Cloud Translation API

Translate text into 100+ languages

- Cloud Speech-to-Text / Text-to-Speech
- Convert audio to text and vice versa

These are pre-trained ML models made available via API — no training required.



Required for All Teams

- Backend logic must be deployed using Google App Engine (Standard or Flexible).
 - This includes REST APIs, server-side logic, or any dynamic computation.
- You may (should) use Firebase Hosting for frontend (HTML, JS, Next.js, etc.).
- Vou may (should) integrate with other Google APIs or Firebase services.