

Implementation of College Cloud on Google App Engine

Prof. L. J. Sankpal¹, Ankush Kawalkar², Suhas Bhattu³, Gaurang Parnaik⁴, Akash Sagar⁵

Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, Maharashtra, India^{1,2,3,4,5}

Abstract: When we want to access the server of the college from the Cloud, then the Google App Engine provides the best building environment for application development. On reasonable rates, this app engine provides the best option for deploying the server on the Cloud. We can access it from anywhere. But for the better use, we have to understand the basics of the Google App Engine. Google also provides the Google Cloud SQL database system for implementation of the database operation such data retrieval and the data modification. We can use this service as a platform for developing applications.

Keywords: Cloud Computing, College Cloud, Google App Engine, Google Cloud SQL.

I. INTRODUCTION

When we want to access the college server from anywhere, we have to deploy it on the internet. But on the internet, the platform for developing new applications is not available and also we cannot expand or shrink the storage according to our requirement. Therefore we can switch to the Cloud solution. In the Cloud Computing, we can develop a virtual network and store the data on that. In the Cloud Computing, we can provide services as a Platform or as a Software or as Infrastructure. But we choose Google for its reasonable cost, best service and quality.

II. GOOGLE APP ENGINE



Fig. 1. Google App Engine Logo

Google App Engine is a Platform as a Service(PaaS). It is a Cloud Computing platform for developing and hosting web applications in Google-managed data centers. It offers automatic scaling for web applications. As the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.

In the Google App Engine, applications are sandboxed and run across multiple servers. Google App Engine is free up to certain level of consumed resources. Fees are charged for additional storage, bandwidth, or instance hours required by the application. It was first released as a preview version in April 2008, and came out of preview in September 2011.

III. SUPPORTED FEATURES

Currently, the supported programming languages are Python, Java, Go, and PHP. Go and PHP are in the experimental status. Google has said that it plans to support more languages in the future, and that the Google App Engine has been written to be language independent. App Engine is designed in such a way that it can sustain multiple datacenter outages without any downtime. This resilience to downtime is shown by the statistics that the High Replication Datastore saw 0%.

IV. GOOGLE CLOUD SQL

In Oct 2011, Google previewed a zero maintenance SQL database, which supports JDBC and DB-API where JDBC supports for Java and DB-API is for other languages like PHP, Python etc. This service is allows to create, configure, and use relational databases with App engine applications. The database engine is MySQL Version 5.1.59 and the database size must be no longer than 10GB, which means we can upload at max of the 10GB of the data at a time. We can shrink and expand it but limit should not be exceeded

Google App Engine requires a Google account to get started, and an account may allow the developer to register up to 10 applications. This limit can be increased by the Google staff. Google App Engine defines usage quotas for free applications. Extensions to these quotas can be requested, and application authors can pay for additional resources.

TABLE I
Hard Limit for Application

Quota	Limit
Time per request	60sec per normal request, 10 minutes for tasks, unlimited for backends
HTTP response size	32MB
Datastore item size	1MB

V. GOOGLE CLOUD SQL INSTANCE

To connect your application with Google Cloud SQL, we have to create the SQL Instance for that is used for MySQL. The steps to create the SQL Instance are as follows:

- 1) From your project's overview page, select Cloud SQL, and click NEW INSTANCE.
- 2) Enter the name for the instance and click CONFIRM.
- 3) Once the instance is running, click ACCESS CONTROL and then;
 - a) Click REQUEST AN IP ADDRESS and use this as your instance-IP in step 3.
 - b) In the SET ROOT PASSWORD box, enter the root password and click SET.
 - c) Click ADD AUTHORIZED NETWORK and add your IP address.

A. Cloud SQL command lines

- 1) Install the Cloud SDK.
- 2) Use the following command to create an instance: `$ gcloud sql instances create <your instance name>`
- 3) Modify the instance and grant access to your IP address: `$ gcloud sql instances patch <your instance name> --authorized-networks <your IP address>`
- 4) Set the root password for the instance. `$ gcloud sql instances set-root-password <your instance name> --password -file <path to file>`
- 5) Assign an IP address to the instance and retrieve it. `$ gcloud sql instances patch <your instance name> --assign-ip $gcloud sql instances get <your instance name>`
- 6) In the output, find the "IPaddress" field. Use this as your instance IP in step 3.

B. Connection with Cloud SQL using MySQL Client

- 1) To use the MySQL, start the interactive shell by using MySQL command and specifying the IP address of the Instance and a user. The command below will prompt for a password.

```
shell> mysql --host=instance-IP -user=username -password
Enter Password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.31 (Google)
.... mysql> Write your commands here.
```

VI. AUTHENTICATION AND AUTHORIZATION

Google APIs use the OAuth 2.0 protocol for authentication and authorization. Google supports common OAuth 2.0 scenarios such as those for web server, installed, and client-side applications. OAuth 2.0 is a relatively simple protocol. To obtain, you obtain OAuth 2.0 credentials from the Google Developers Console.

Then your client application requests an access token from Google Authentication Server, extracts a token from a response, and sends a token to the Google API that you want to access.

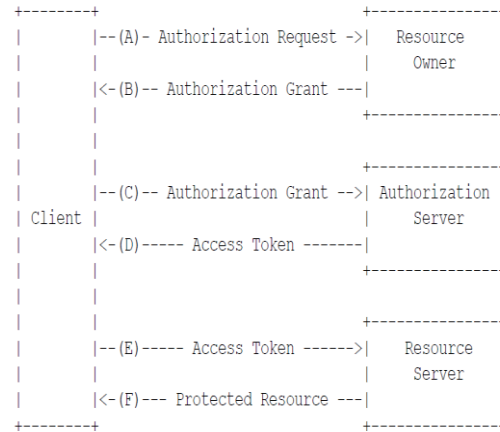


Fig 2. Basics Steps for accessing a Google API using OAuth 2.0

First step to access the Google API is to get the Authorization Grant where client sends request to the Resource Owner where Resource owner replies with the Authorization Grant. Then Client sends that Authorization Grant to the Authorization Server and it replies with the access token. Then Client sends the Access Token to the Resource Server and it replies with Protected Resource. Here Resource Owner and the Resource Server have control of all the resources.

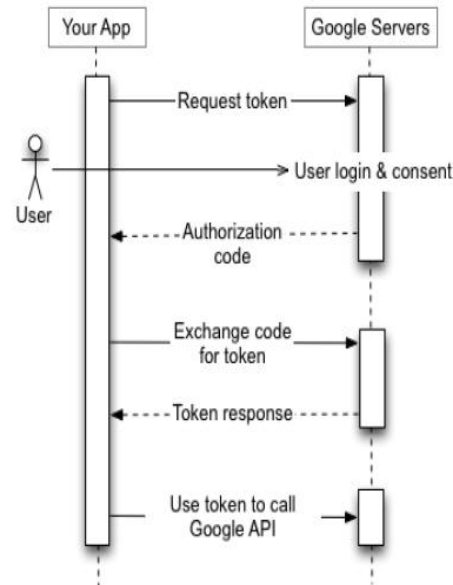


Fig. 3. Using OAuth 2.0 for web server applications

The Google OAuth 2.0 endpoint supports web server applications that use languages and frameworks such as PHP, Java, Python, Ruby and ASP.NET. The authorization sequence begins when your application redirects a browser to the Google URL; the URL includes query parameters that indicate the type of access being requested. Google handles the user authentication, session selection and user consent. The result is an authorization code, which the application can exchange for an access token and a refresh token. The application should store the refresh token for future use and use the access token to

access a Google API. Once the access token expires, the application uses the Refresh token to obtain a new one.

V. CONCLUSION

Google App Engine provides the efficient way for the application development on the Cloud. It is also in the reasonable cost and estimation requirement for application development is also less. It provides the Platform as a Service and for database Google has a facility of the Google Cloud SQL. For authentication and authorization it uses the OAuth 2.0 protocol. By shifting to Google Cloud facilities, we get the best results.

REFERENCES

- [1] Prof. L. J. Sankpal, Ankush Kawalkar, Suhas Bhattu, Gaurang Parnaik, Akash Sagar – Cloud Computing in Education System IJARCCCE Vol. 3, Issue 2, February 2014.
- [2] <https://developers.google.com/appengine>
- [3] <https://cloud.google.com/appengine>
- [4] www.wikipedia.com/Google_App_Engine
- [5] www.stackoverflow.com/google-app-engine
- [6] www.wikipedia.com/wiki/OAuth