

Google Cloud Platform Project Website with Class Roster



A Report By:
Nathan Guenther & Zach Colby
As a Part of CIS 4932C: Cloud Computing Concepts
4/19/2016

1. Abstract

This project is a web application that is built on Google's Cloud Platform. Google's Cloud Platform is an intuitive suite with many different products. We used Google Compute Engine and Google Container Engine to create a class roster online. We wanted students to be able to sign into the class lecture by going online and entering their data to show that they were there in class. It took us two trials, first doing it with Redis database, and then doing it with MySQL database because we needed a storage that was persistent and not ephemeral. We used Docker containers to run our project and Kubernetes to manage those docker containers. We also utilized Google's tools to control the administration aspects of the website.

2. Research

2.1 Google Cloud Platform

Google Cloud Platform is a cloud computing platform that is used by everyone including business professionals and tech enthusiasts. Google Cloud Platform is actually the same infrastructure and technology that Google uses for their own Search Engine and for YouTube services as well. The entire platform has a range of products including Google App Engine, Google Compute Engine, and Google Container Engine, among numerous other services for running a cloud Infrastructure with big data. Google App Engine is an all-inclusive product that allows you to deploy web applications without having to do any of the grunt work under the hood. Google Compute Engine is useful for launching your own Virtual Machine (VM) to do whatever work necessary to do. Google Container Engine uses Docker to manage containers by spinning up separate Virtual Machines to do the work.

2.2 Docker

Docker is a Linux based platform that is used to make containers. It does this abstractly by using the same properties as a virtual machine. These containers contain all the files, tools, and code needed to run any software on any system that supports Docker. This guarantees that the software behaves the same on all types of systems.

2.3 Kubernetes

Kubernetes is a container manager that can manage Docker containers. This product was built by Google to manage their own containers and is available as an open source product for all to use. Every single thing internally at Google is run inside of a container, and has been for over 10 years. Google just recently open sourced the tools they use for others to benefit.

2.4 Redis vs MySQL

Redis is a database server that uses a NoSQL structure. Being NoSQL, it is easily scalable, and can be implemented to the project. MySQL is a relational database server that is open source as well as Redis. The main differences between Redis and MySQL is that Redis is NoSQL, whereas MySQL is not. The other main difference, and the reason we chose to use Redis initially is that it stores everything in memory. This provides for much faster read/write commands to the database.

3. Implementation

3.1 First Attempt with In-Memory Storage

Redis was used at the first attempt to create the product. It worked great until we realized we needed long term storage, and not temporary. Every time our instance was restarted, our class roster would clear because it wasn't persistent storage. This is obviously a big problem because having a class roster that clears every time the instance is restarted defeats the purpose of putting the class roster online. We needed a solution with persistent storage, and not storage stored only in temporary memory.

3.2 Second Attempt with Persistent Storage

For our second attempt, we rebuilt the class roster with persistent storage instead. We used MySQL to store the class roster information the second time. MySQL isn't as fast because it's not in-memory storage, but it is persistent which is what is needed for this solution. We were able to easily create the SQL disk by running the following command:

```
gcloud compute disks create --size 200GB mysql-disk
```

This uses Google's compute engine to spin up a small MySQL server with an attached persistent disk. The instance is separate from the other instances that are used. Similarly another instance and accompanying persistent disk are used for a PHP server and to host the WordPress files.

4. Completed Product

4.1 Class Roster Section

Below is a screenshot of the class roster that was built into the website

Attendance Sign In

Name: *

Roster Entry:

Submit

Fields marked with * are required.
For security reasons, we save the IP address 50.120.1.1.

Kevin Bacon wrote on April 19, 2016 at 1:54 pm:
I'm back! 😊

Jordan Z wrote on April 19, 2016 at 1:51 pm:
Hello 😊

Zach Colby wrote on April 19, 2016 at 1:44 pm:
I'm here 😊

4.2 Project Description Section

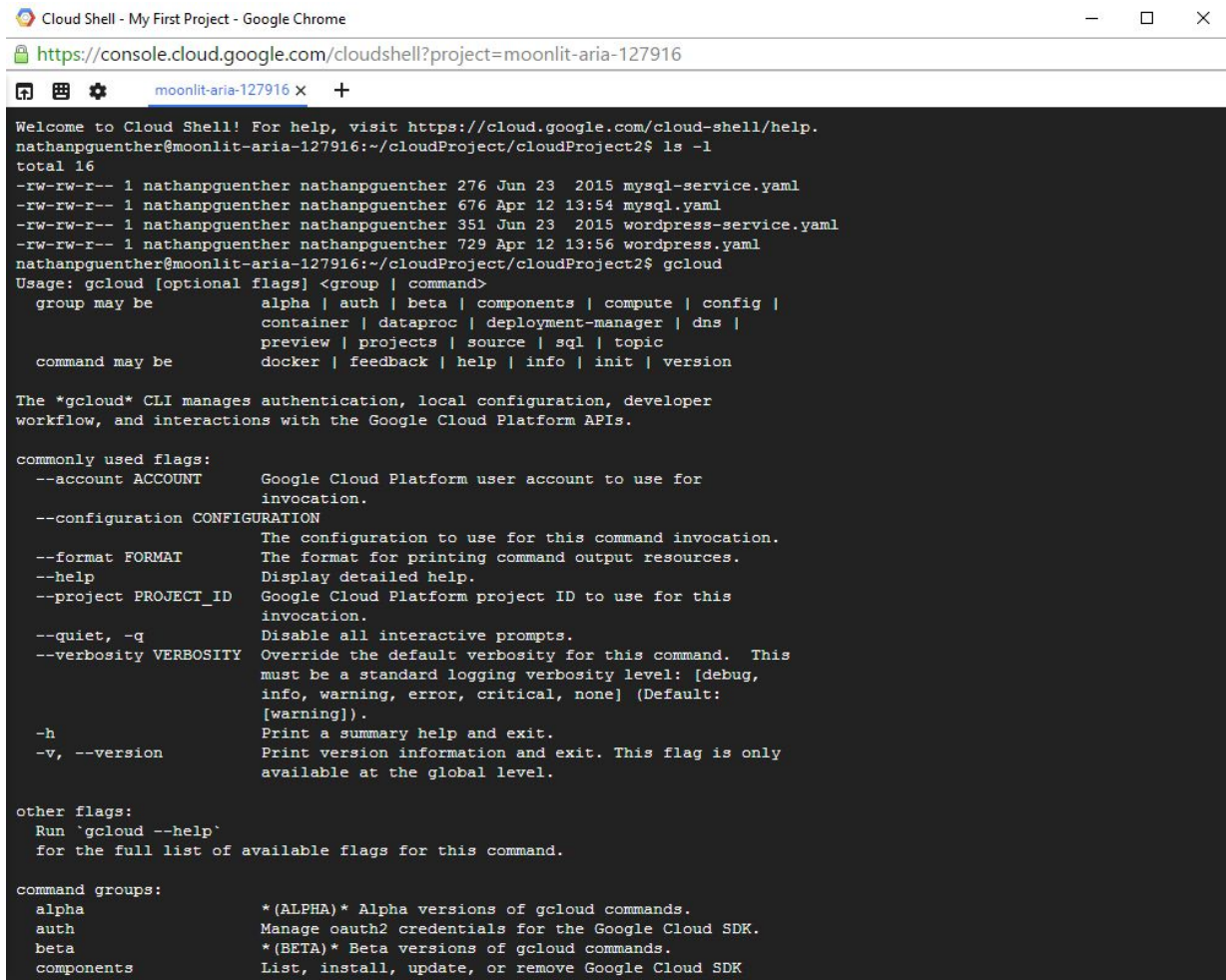
There is also a section on the website that explains the technologies used and everything that was done to create the project on the cloud. Below is a screenshot of the different pages that the website includes.



5. Google Tools

5.1 gcloud Tool

Apart of the Google Cloud SDK, gcloud is a command line tool that can be used to manage both your development workflow and your Google Cloud Platform resources. We used this to create instances, set up storage disks, and configure the cluster management.



```
Cloud Shell - My First Project - Google Chrome
https://console.cloud.google.com/cloudshell?project=moonlit-aria-127916
moonlit-aria-127916 x +
Welcome to Cloud Shell! For help, visit https://cloud.google.com/cloud-shell/help.
nathanpguenther@moonlit-aria-127916:~/cloudProject/cloudProject2$ ls -l
total 16
-rw-rw-r-- 1 nathanpguenther nathanpguenther 276 Jun 23 2015 mysql-service.yaml
-rw-rw-r-- 1 nathanpguenther nathanpguenther 676 Apr 12 13:54 mysql.yaml
-rw-rw-r-- 1 nathanpguenther nathanpguenther 351 Jun 23 2015 wordpress-service.yaml
-rw-rw-r-- 1 nathanpguenther nathanpguenther 729 Apr 12 13:56 wordpress.yaml
nathanpguenther@moonlit-aria-127916:~/cloudProject/cloudProject2$ gcloud
Usage: gcloud [optional flags] <group | command>
  group may be
    alpha | auth | beta | components | compute | config |
    container | dataproc | deployment-manager | dns |
    preview | projects | source | sql | topic
  command may be
    docker | feedback | help | info | init | version

The *gcloud* CLI manages authentication, local configuration, developer
workflow, and interactions with the Google Cloud Platform APIs.

commonly used flags:
--account ACCOUNT      Google Cloud Platform user account to use for
                        invocation.
--configuration CONFIGURATION
                        The configuration to use for this command invocation.
--format FORMAT        The format for printing command output resources.
--help                Display detailed help.
--project PROJECT_ID  Google Cloud Platform project ID to use for this
                        invocation.
--quiet, -q           Disable all interactive prompts.
--verbosity VERBOSITY Override the default verbosity for this command. This
                        must be a standard logging verbosity level: [debug,
                        info, warning, error, critical, none] (Default:
                        [warning]).
-h                    Print a summary help and exit.
-v, --version          Print version information and exit. This flag is only
                        available at the global level.

other flags:
Run 'gcloud --help'
for the full list of available flags for this command.

command groups:
alpha                *(ALPHA)* Alpha versions of gcloud commands.
auth                 Manage oauth2 credentials for the Google Cloud SDK.
beta                 *(BETA)* Beta versions of gcloud commands.
components           List, install, update, or remove Google Cloud SDK
```

6. Member Contribution

This project was completed by Nathan Guenther and Zach Colby. Nathan has an IT background so we was more familiar with the Google Cloud Platform than Zach. Nathan headed the research to figure out the best technologies to use. His research showed that the suite of tools on Google Cloud Platform was the best technology for us to use. He also came up with the idea to use Kubernetes to manage the Docker containers. Zach did research on databases and came up with the idea to use MySQL, when the team realized Redis wasn't going to work.

The initial setup was completed by Zach, by running commands to create the MySQL, and WordPress instances. Nathan went in and edited the front end aspect of the WordPress site, and he investigated how to use tools to stress test the backend of the application and monitor the metrics.

Zach handled a lot of the documentation, specifically the creation of this report, while Nathan headed the preparation for the in class demonstration. Both team members put in equal effort and used their skills to contribute to the project to make it a success.