Project Proposal

Two-Step Email Verification: A GCP Implementation

Under the supervision of:

Prof. Jaspreet Bhatia

CPSC-5207EL-01-2023F

LAURENTIAN UNIVERSITY

BHARTI SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

Team members:

Oluwatomisin Taiwo
Khandakar Asef Erfan
Prathyusha Papysettypally
Mahmudul Hasan
Md Ismail Hossain

Table of Contents

1. Abstract

Today’s generation has become so accustomed to digital interactions and transactions that it is essential to protect user data and implement robust authentication procedures. The aim of this project is to enhance user login security through the implementation of a Two-Factor Authentication (2FA) system and the use of Google Cloud Platform (GCP).

Two-Factor Authentication stands as a crucial shield against unauthorized access by adding an additional layer of verification beyond traditional password protection. By requiring usersto authenticate themselves through a combination of something they know (password) and something they possess (e.g., a mobile device), 2FA significantly fortifies the security posture of online platforms. This approach mitigates the risks associated with compromised passwords, phishing attacks, and unauthorized access, thus safeguarding sensitive user information. Google Cloud Platform emerges as the ideal solution for this purpose, providing a suite of services designed to handle the demands of modern authentication systems. The auto-scaling capabilities of GCP services allow the authentication system to seamlessly adapt to fluctuating workloads,ensuring optimal performance during peak usage periods without compromising on security.

This project not only focuses on the implementation of a secure and user-friendly 2FA system but also underscores the pivotal role that cloud platforms, specifically GCP, play in achieving the scalability and reliability required for the modern digital landscape. Further columns in this paper outline the project's introduction, pipeline, components, and architecture's workflow on Google Cloud Platform with the focus on scalability, security, and user satisfaction as crucial design considerations.

2. Introduction

2.1 Background

Two-factor authentication (2FA) is a security process in which a user provides two different authentication factors to verify their identity. The first factor is typically a password, and the second factor can be a fingerprint, a smart card, or a one-time code sent to the email in this

case. 2FA is needed to provide an additional layer of security to protect against unauthorized access to sensitive information [1]

While the implementation of 2FA significantly elevates the security posture, it introduces challenges related to scalability, particularly in environments characterized by dynamic user volumes and varying authentication demands. Traditional authentication infrastructures often struggle to efficiently scale to accommodate peak loads, leading to potential latency issues and compromised user experiences.[2]

This project acknowledges the inherent scalability challenges associated with implementing 2FA and aims to address these concerns through the utilization of Google Cloud Platform (GCP). GCP's cloud-native architecture provides a scalable and elastic foundation, enabling the authentication system to dynamically adapt to changing workloads while ensuring consistent and reliable performance. The seamless scalability offered by GCP becomes a pivotal element in the successful implementation of a secure and responsive 2FA system.[3]

2.2 Objectives

- Finding the feasibility of cloud integration into two factor authentication
- Estimating the cost/performance ratio of the implementation
- Noting Potential limitations of using cloud vs on-premises implementation

2.3 Scope

This project focuses on the development and demonstration of a Two-Factor Authentication (2FA) system using Google Cloud Platform (GCP) within the controlled environment of our university's authentication page. The scope is intentionally limited to a simulated demonstration, and no real user data is utilized to ensure compliance with privacy and security standards.

The project involves the cloning of our university's authentication page for demonstration purposes only [4]. The cloned page serves as the backdrop for integrating and showcasing the 2FA system. The primary objective is to implement a 2FA system on the cloned authentication page. This involves integrating GCP services, such as Cloud SQL and App

Engine, to enhance the security of user logins. Simulated user interactions will be employed.

to demonstrate the 2FA process. Users will experience the additional layer of verification beyond passwords, showcasing the effectiveness of 2FA in enhancing security. No real user data from the university's authentication system is used in the project. Emphasis is placed on data privacy and compliance with ethical standards. All user interactions are simulated to maintain the confidentiality and integrity of personal information.

While the project is implemented within the university's infrastructure, the report will include a discussion on the scalability implications of the 2FA system in a commercial scenario [5]. This analysis will be based on theoretical considerations and industry best practices.

3. Project Model

We recreated the self-service Laurentian login page and worked on implementing 2FA authentication on cloned page to demonstrate this project. We utilized node.js and express for the backend and HTML, angular for the front end to accomplish this.

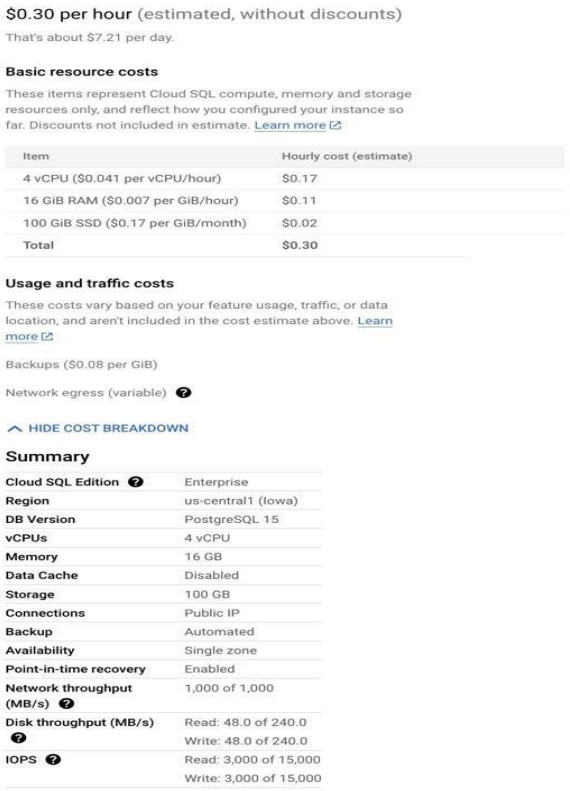
We have deployed this on GCP containers registry and used APP Engine and Cloud RUN to deploy the backend and front end, respectively [6]. App Engine is connected with Cloud SQL to store user relational data and Cloud Bucket for storing photos uploaded by users during their initial registration. We then integrated the APP Engine with Pub sub to push the randomly generated codes that are generated in the backend in order to provide an additional layer of authentication. After that, Pub sub delivers the message to the cloud function so that the user can receive authentication. We have successfully implemented 2FA authentication in GCP by using this method.

3.1 Cost Estimation

The cost estimation for this project relies on Google Cloud Platform's (GCP) Pricing Calculator, a comprehensive tool designed to provide accurate and transparent cost projections for utilizing GCP services. The calculator facilitates a detailed analysis of the financial implications associated with deploying various GCP resources, ensuring that organizations can make informed decisions regarding infrastructure, services, and scalability.

It has key benefits for the project including Service-Specific Cost Breakdown, Custom Configuration Options, Real-Time Updates, Project Lifecycle Considerations, Multi-Cloud Cost Comparison [7].

For our demonstration project we have run the services for 5 days from 7/12/2023 to 12/12/2023. Below is the estimation and cost consumption for the 4 core services we used



$0.30 per hour (estimated, without discounts)

That's about $7.21 per day.

Basic resource costs

These items represent Cloud SQL compute, memory and storage resources only, and reflect how you configured your instance so far. Discounts not included in estimate. Learn more

Item	Hourly cost (estimate)
4 vCPU ($0.041 per vCPU/hour)	$0.17
16 GiB RAM ($0.007 per GiB/hour)	$0.11
100 GiB SSD ($0.17 per GiB/month)	$0.02
Total	$0.30

Usage and traffic costs

These costs vary based on your feature usage, traffic, or data location, and aren't included in the cost estimate above. Learn more

Backups ($0.08 per GiB)

Network egress (variable)

HIDE COST BREAKDOWN

Summary

Cloud SQL Edition	Enterprise
Region	us-central1 (Iowa)
DB Version	PostgreSQL 15
vCPUs	4 vCPU
Memory	16 GB
Data Cache	Disabled
Storage	100 GB
Connections	Public IP
Backup	Automated
Availability	Single zone
Point-in-time recovery	Enabled
Network throughput (MB/s)	1,000 of 1,000
Disk throughput (MB/s)	Read: 48.0 of 240.0 Write: 48.0 of 240.0
IOPS	Read: 3,000 of 15,000 Write: 3,000 of 15,000

Figure 1: *Pricing Estimate for current Services*

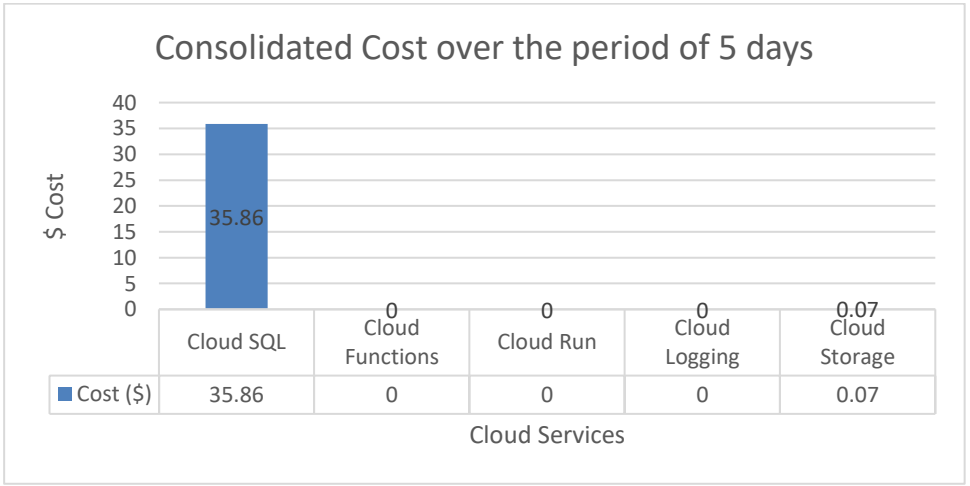


Figure 1 *Consolidated Cost Consumption*

4. GCP Block Diagram and Discussion

This block diagram provides a high-level overview and the interactions of the key components in a Two-Step Email Verification system implemented on the Google Cloud Platform. In the pipeline, we have used many GCP components to serve various purposes in the system implementation right from storing the data to delivering the message. Not just keeping the function usage we have also considered scalability and cost effectiveness while choosing the components.

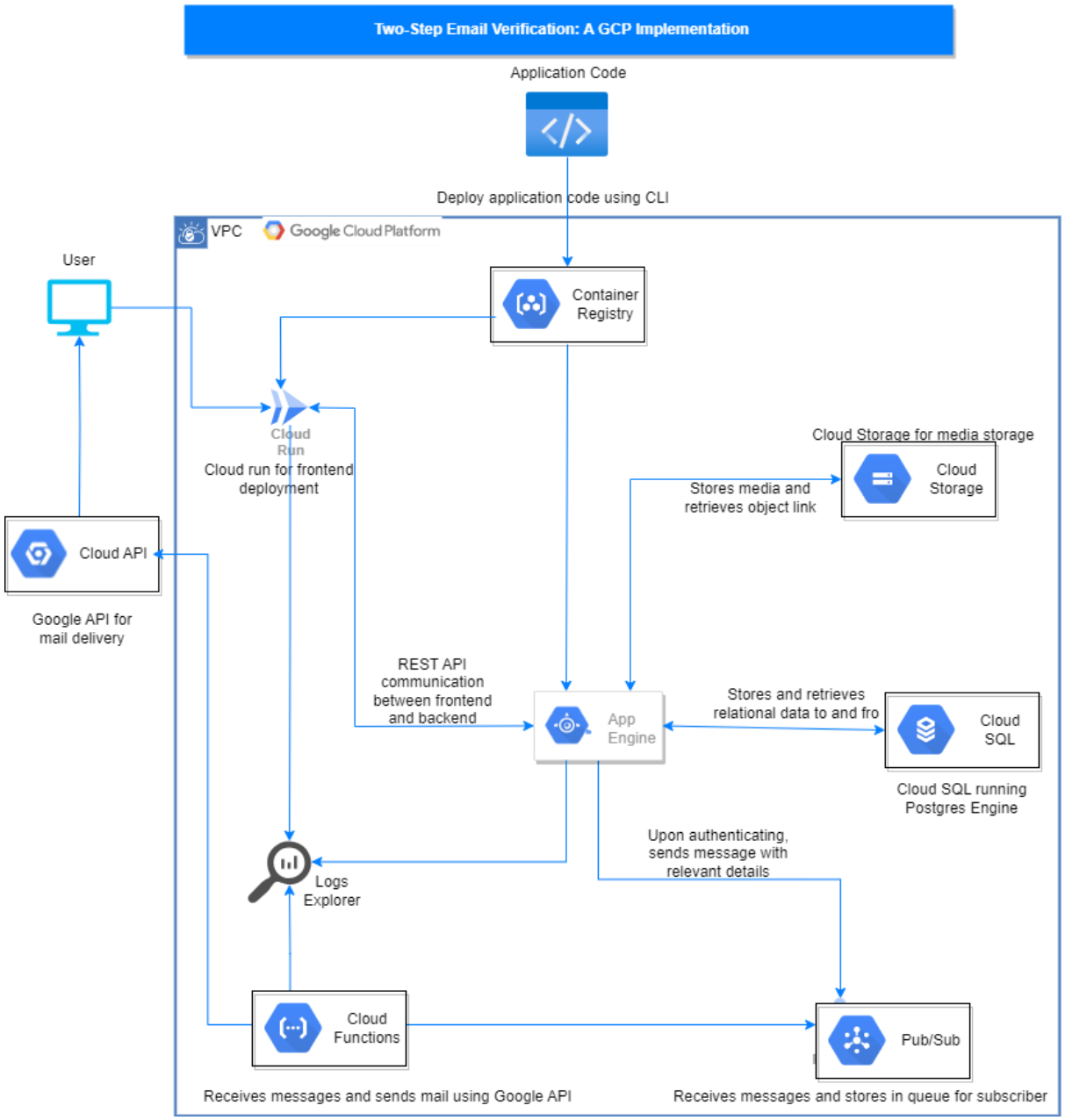


Figure 3: *GCP Block Diagram*

4.1 Cloud Storage

Google Cloud Platform (GCP) Cloud Storage stands as a versatile and scalable object storage service designed to meet the storage needs of modern cloud-based applications. At its core is the concept of a "bucket system," a fundamental organizational structure that facilitates efficient storage and retrieval of data within GCP Cloud Storage [8]. The features that were used in this project include Object storage model, integration with other GCP services including Cloud SQL.

First the User enters the user id and password on the front-end login page, which itself is stored in GCP Cloud Storage bucket.

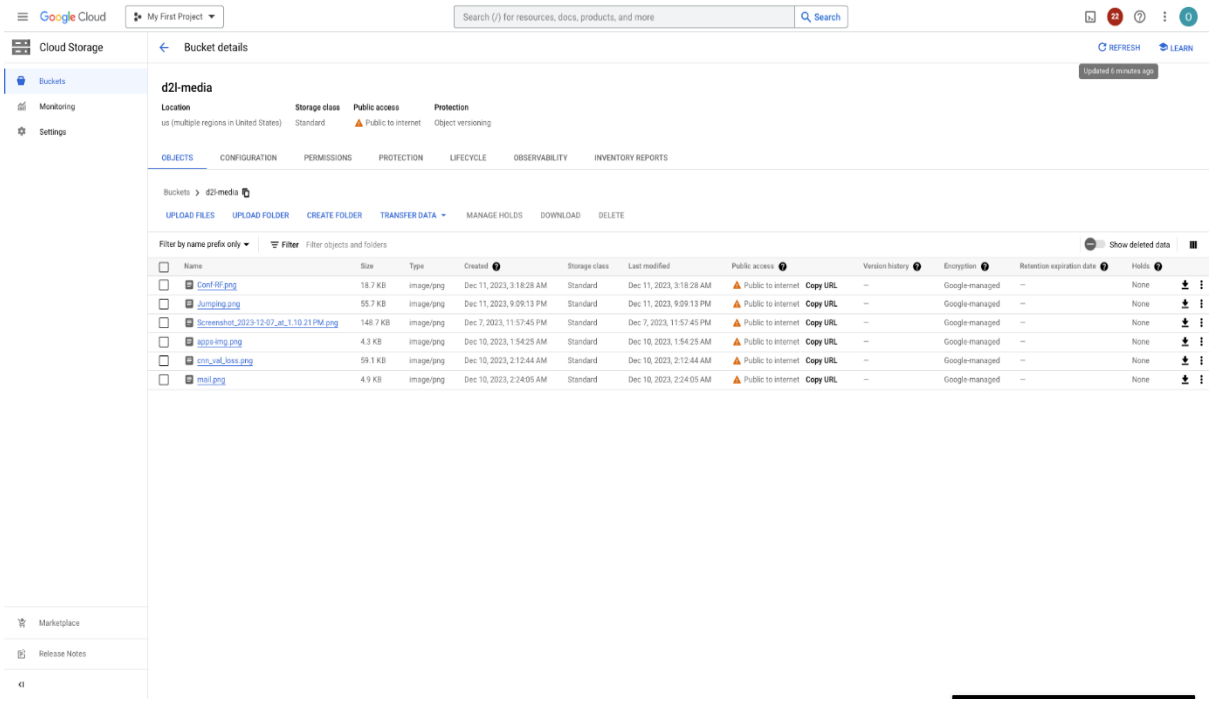


Figure 2: *Static Media Images stored in the Bucket.*

The cloud storage then relays the information to App Engine which validates with Cloud SQL for the correct relational data and matches the user image and further stores in another bucket in Cloud storage.

4.2 App Engine

Google Cloud Platform's (GCP) App Engine represents a fully managed and serverless platform designed to simplify the deployment and scaling of applications. This abstract

explores the key features and utility of GCP App Engine, particularly its role in backend development for modern cloud-based applications. [9] App Engine provides a range of built-in services, such as data storage with Cloud Datastore, authentication with Identity-Aware Proxy (IAP), and caching with Cloud memory store. These services streamline application development and reduce the need for external dependencies. Leveraging Google's global infrastructure, App Engine ensures low-latency access forusers worldwide. Applications can be deployed in multiple regions, enhancing user experience and responsiveness.

App Engine follows a pay-as-you-go pricing model, allowing organizations to optimize costs based on actual resource usage. The serverless nature of the platform eliminates costs associated with idle resources.

For this project we have deployed the website JS code on the app engine as well as the 2FA codegenerator onto it.

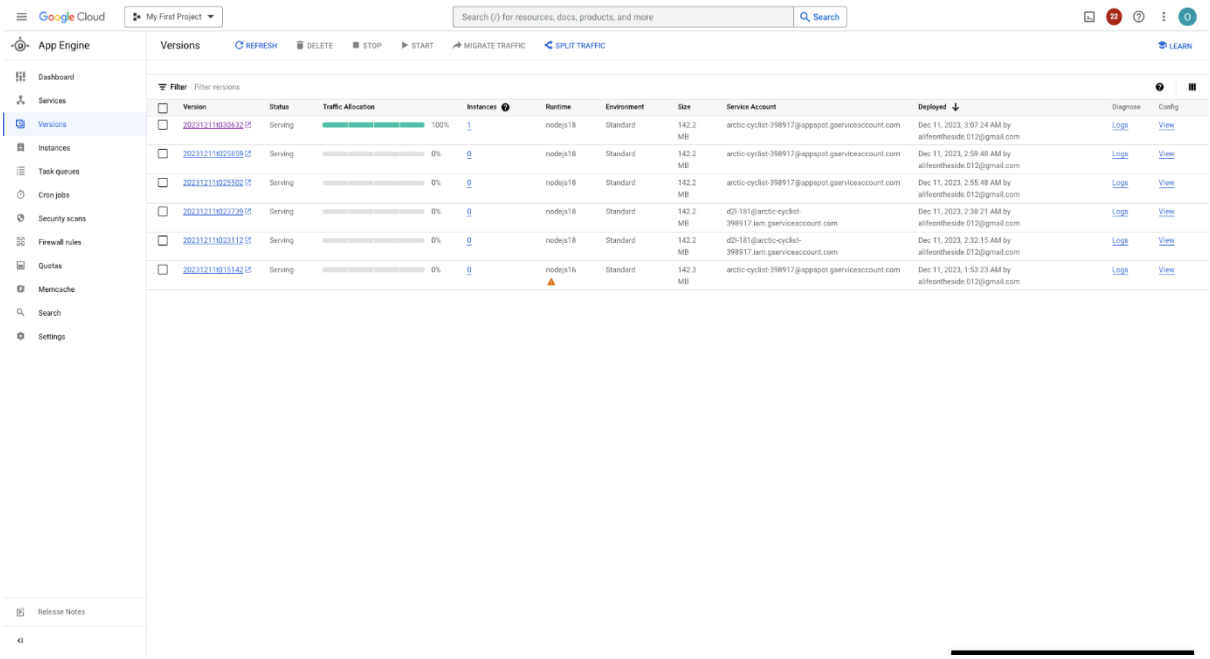


Figure 3 : *App engine with backend*

4.3Cloud SQL

Google Cloud SQL, a fully managed relational database service on Google Cloud Platform (GCP), provides a reliable and scalable solution for storing and managing data in a relational database format. This abstract explores the capabilities of GCP Cloud SQL, particularly its role in securely storing user IDs and passwords.[10]

In the context of user authentication and security, GCP Cloud SQL plays a crucial role in securely storing user IDs and passwords. Utilizing a relational database schema, sensitive user credentials can be stored with encryption at rest, ensuring the confidentiality of authentication information. Cloud SQL's support for secure connections and IAM-based access controls further enhances the protection of stored data.

In this project we used SQL functionality to store our test user ID and passwords.

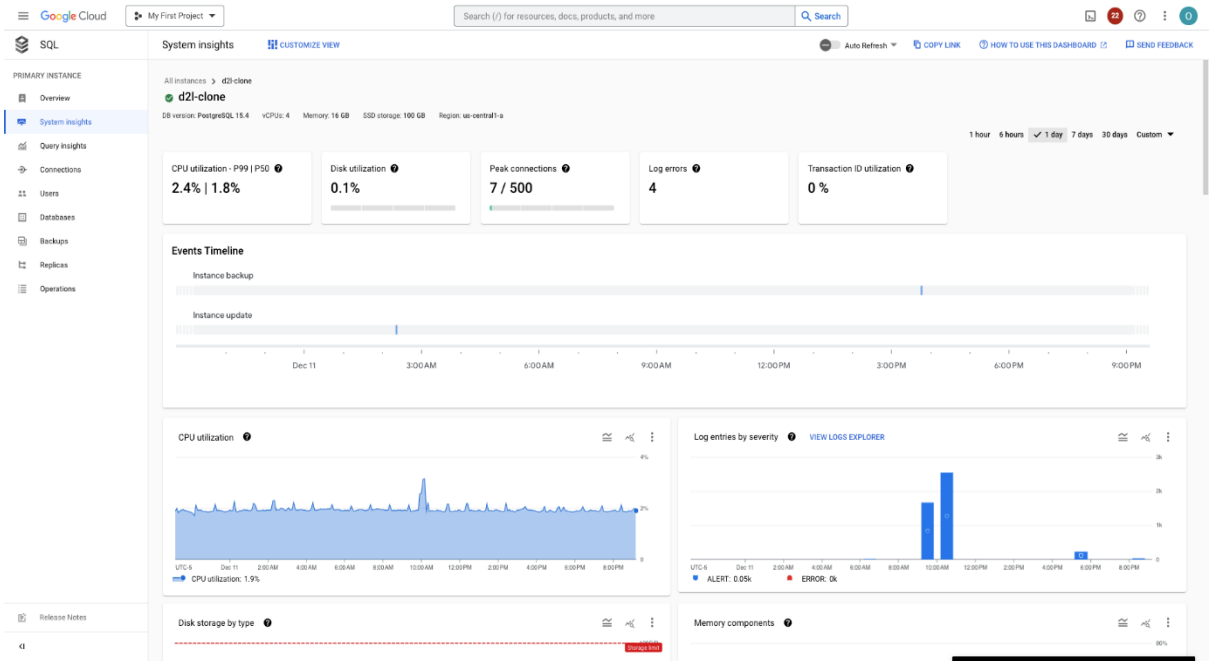


Figure 4: *Cloud SQL for storing User Data*

4.4 Pub/Sub

In the context of email queuing, GCP Pub/Sub serves as a middleware for coordinating the delivery of emails across an application.[11] Publishers can send messages to a Pub/Sub topic, and subscribers (such as email processing services) receive and process these messages, ensuring efficient and reliable email delivery.

For managing Two-Factor Authentication (2FA) codes, Pub/Sub can be employed to handle the distribution of codes securely. Upon user authentication, the system can generate a 2FA code and publish it to a Pub/Sub topic. Subscribers, such as authentication services, can then consume and validate these codes, enhancing the security of user access.

The decoupled nature of Pub/Sub allows for flexibility in scaling email processing and 2FA validationindependently, ensuring optimal performance and responsiveness. By leveraging Pub/Sub, developers can implement a resilient and scalable architecture for handling critical aspects of user communication and authentication in a cloud-based environment.

For this Demonstration, A Subscription was employed in pub/sub to queue the 2FA codes email.

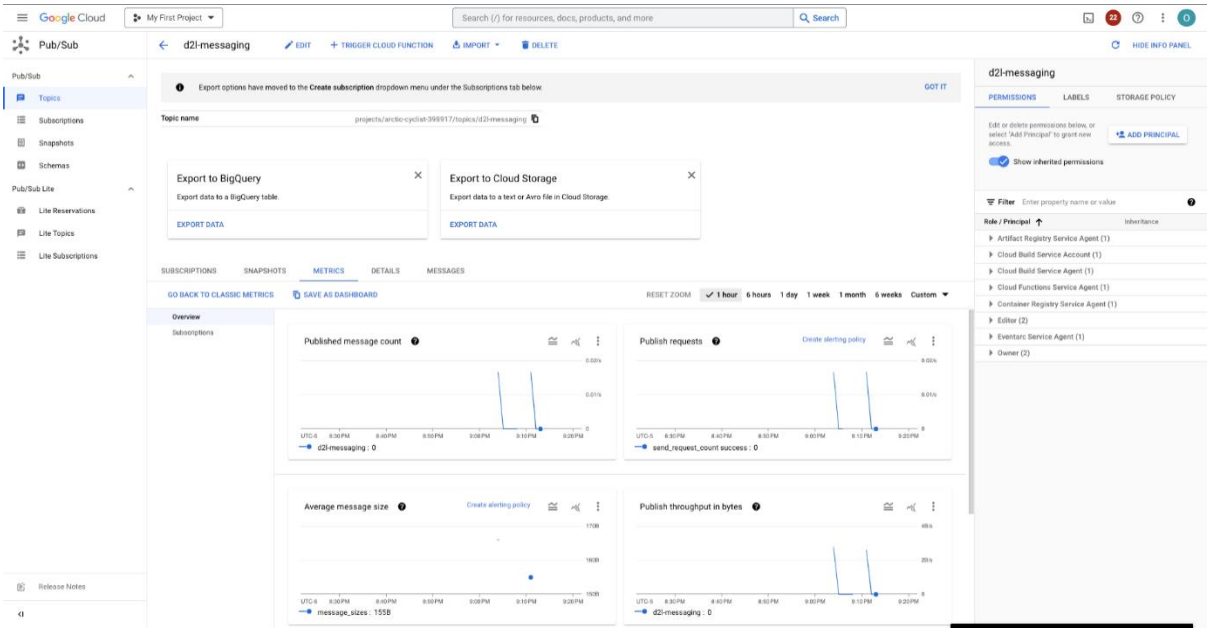


Figure 5: *Pub/Sub for Queuing the Requests*

In addition to the services mentioned above, we used additional services such as Container Registry, Cloud Run, Cloud Functions, VPC, and log explorer to execute this pipeline on GCP. App Engine is connected to Clouds Run, which is mainly used to deploy the Front-End portion of the webpage that is faced to the user. When the App Engine receives the user data, it processes the requests to Cloud SQL and Pub/Sub to finish the login.

5 Implementation

The implementation of the Two-Factor Authentication (2FA) system within the "Self Service Laurentian" login page involves a streamlined workflow to enhance the security of user access. The complete successful sign in involves many stages and the process begins as follows:

- It begins with the user entering a dummy ID and password, which are stored securely in Google Cloud SQL.
- Upon successful entry of the dummy credentials, the App Engine, a fully managed serverless platform on Google Cloud Platform (GCP), processes the user input.
- The App Engine retrieves thestored user information from Cloud SQL and initiates the 2FA code generation.
- The 2FA code is generated using a secure algorithm, ensuring uniqueness and unpredictability. Thiscode is an additional layer of verification beyond the entered dummy password and is designed to enhance the security of the authentication process.
- The generated 2FA code is then sent to the user via email. Google Cloud Pub/Sub, a fully managedmessaging service, facilitates the queuing and distribution of this code. The Pub/Subtopic is designed to handle the communication between the App Engine and the email sending service.
- A dedicated service responsible for sending emails subscribes to the Pub/Subtopic. Upon receiving the 2FA code message, it dynamically generates an email with instructions for the user and dispatches it to the specified email address associated with the dummy ID.
- The user is prompted to check their email for the 2FA code. Upon receiving the email, the user enters the code back into the login page. This creates a seamless and secure interaction, requiringboth the dummy password and the dynamically generated 2FA code for successful authentication.
- The App Engine validates the entered 2FA code against the generated code. If the codes match, theuser is granted access, and a successful login is confirmed. This two-step verification process significantly enhances the security of user authentication.

6 Future Work

The implemented Two-Factor Authentication (2FA) system on Google Cloud Platform (GCP) for the "Self Service Laurentian" login page lays a foundation for secure user

authentication. To ensure future scalability in a commercial environment, several factors should be considered and few of the main limitations are as follows:

6.2 Future Scalability and Cost

As the user base expands, the system needs to efficiently handle a higher volume of authentication requests. Utilize GCP's auto-scaling capabilities to dynamically adjust resources based on demand, ensuring optimal performance during peak usage periods. Monitor and analyze traffic patterns to identify peak usage times. Implement load balancing strategies to distribute incoming requests evenly across multiple instances, preventing bottlenecks and ensuring a responsive system.

Cost estimation for a larger user base involves a holistic analysis of various factors, including infrastructure, networking, storage, and security measures. By employing proactive cost management strategies, continuous optimization, and leveraging GCP's pricing tools, the project canstrike a balance between scalability and cost-effectiveness, ensuring that the Two-Factor Authentication system remains financially sustainable as it scales to accommodate a growing and dynamic user community.

6.3 Optimization and Caching

As the user base grows, consider optimizing the 2FA code generation process for efficiency. Evaluatealgorithms and methodologies to maintain the speed and reliability of code generation while accommodating a larger number of concurrent users. Implement caching mechanisms for frequentlyaccessed data, such as user information and previous 2FA codes. This reduces the load on the system, enhances response times, and supports scalability by minimizing redundant data processing.

6.4 Security and Availability

Continuously assess and enhance security measures to withstand evolving threats. Regularly updateencryption protocols, strengthen code generation algorithms, and stay informed about the latest security best practices to ensure robust protection against unauthorized access. Consider deploying the system in multiple geographic regions to ensure global availability and low-latency access for users worldwide.

Utilizing GCP's global infrastructure to replicate data and services across multiple regions, enhancing reliability and resilience. Keep abreast of regulatory requirements related to user authentication and data privacy. Ensure that the implemented system complies with industry standards and legal frameworks to avoid compliance issues as the user base expands.

6.5 Better User integration and availability

Embrace continuous integration and deployment (CI/CD) practices to streamline the development lifecycle. Automated testing and deployment pipelines help maintain code quality, facilitate rapid updates, and ensure the seamless introduction of new features or security enhancements.

Implement comprehensive monitoring and logging solutions to gain insights into system performance, user interactions, and potential security incidents. Leverage GCP's monitoring and logging services to proactively address issues and optimize resource utilization. Focus on enhancing the user experience by continuously optimizing the authentication process. Evaluate user feedback, conduct usability testing, and implement improvements to make the 2FA system intuitive and user-friendly.

In summary, the future scalability of the implemented GCP-based Two-Factor Authentication system in a commercial setting relies on proactive measures, continuous optimization, and adherence to best practices. By addressing the evolving needs of a growing user base and leveraging the scalability features of Google Cloud Platform, the system can maintain its effectiveness and security in a dynamic and expanding environment.

7 Conclusion

An effective implementation of the 2FA secure login is made in this test environment. By adding a second layer of verification and maximizing the usage of GCP services, this pipeline aimed to increase the security of user authentication.

A scalable and secure authentication procedure has been developed by using GCP services including Cloud SQL, App Engine, and Pub/Sub, Cloud Storage. The integration of Cloud SQL ensures the secure storage of user credentials, while the App Engine orchestrates the 2FA code generation and validation seamlessly. Pub/Sub facilitates the reliable queuing and

distribution of 2FA codes via email, contributing to a holistic and efficient user authentication workflow. While Cloud storage enabled the raw images and user images storage very convenient.

The project not only demonstrates the successful implementation of a secure 2FA system but also highlights the importance of GCP in providing a foundation for scalability, reliability, and ease of integration. The serverless architecture of App Engine, coupled with Pub/Sub's messaging capabilities, ensures that the system is equipped to handle varying workloads and scale dynamicallybased on user demand.

As we look toward the future, considerations for scalability in a commercial setting have been outlined, encompassing aspects such as increased cost, increased user base, optimized code generation, global availability, and continuous security measures. These considerations underscore the adaptability and sustainability of the implemented solution as it evolves to meet the demands ofa growing and diverse user community.

8 References

[1] K. Reese, T. Smith, J. Dutson, J. Armknecht, J. Cameron, and K. Seamons, "A Usability Study of Five {Two-Factor} Authentication Methods," presented at the Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019), 2019, pp. 357–370. Accessed: Dec. 09, 2023. [Online]. Available:https://www.usenix.org/conference/soups2019/presentation/reese

[2] A. Al-Said Ahmad and P. Andras, "Scalability analysis comparisons of cloud-based software services," *J. Cloud Comput.*, vol. 8, no. 1, p. 10, Jul. 2019, doi: 10.1186/s13677-019-0134-y.

[3] S. Kaur, G. Kaur, and M. Shabaz, "A Secure Two-Factor Authentication Framework in Cloud Computing," *Secur. Commun. Netw.*, vol. 2022, p. e7540891, Mar. 2022, doi: 10.1155/2022/7540891.

[4] "Laurentian Student Application." Accessed: Dec. 09, 2023. [Online]. Available:https://selfservice.laurentian.ca/Student/?hideProxyDialog=false

[5] G. Griffis, "How to scale in the cloud," Work Life by Atlassian. Accessed: Dec. 09, 2023.[Online]. Available: https://www.atlassian.com/blog/platform/how-to-scale-in-the-cloud

[6] "How instances are managed | Google App Engine standard environment docs," Google

Cloud. Accessed: Dec. 09, 2023. [Online]. Available: https://cloud.google.com/appengine/docs/standard/how-instances-are-managed

[7] "Google Cloud Pricing Calculator," Google Cloud. Accessed: Dec. 09, 2023. [Online]. Available: https://cloud.google.com/products/calculator

[8] "Cloud Storage," Google Cloud. Accessed: Dec. 09, 2023. [Online]. Available:https://cloud.google.com/storage, https://cloud.google.com/storage

[9] "App Engine Application Platform," Google Cloud. Accessed: Dec. 09, 2023. [Online]. Available: https://cloud.google.com/appengine, https://cloud.google.com/appengine

[10] "Cloud SQL documentation | Cloud SQL Documentation," Google Cloud. Accessed: Dec. 09,2023. [Online]. Available: https://cloud.google.com/sql/docs

[11] "Pub/Sub documentation | Cloud Pub/Sub Documentation," Google Cloud. Accessed: Dec. 09, 2023. [Online]. Available: https://cloud.google.com/pubsub/docs

9 GitHub Links

- Frontend Page : https://github.com/thomsyne/d2l-clone
- Email Configuration : https://github.com/thomsyne/d2l-email
- Backend - https://github.com/thomsyne/d2l-clone-backend
- Report , PPT and the Demo Video is uploaded in GitHub - thomsyne/cloud-group-10