Cloud-Based Intrusion Detection System Using Google App Engine and Compute Engine

George Muniz and Thomas Haugstad

Cloud Computing – Fall 2025
Instructor: Hoffman

October 26, 2025

Abstract

For our final project, we propose a cloud based Intrusion Detection System (IDS) designed and deployed using Google Cloud Platform (GCP). This project will focus on the task of comparing two deployment methods covered in class, the first one being: Google App Engine (PaaS) and the second being Google Compute Engine (IaaS). Our goal is to determine whether App Engine's automatic scaling and managed infrastructure can handle variable network traffic in a way that would be more efficient and cost-effective than a fixed virtual machine on Compute Engine, while also maintaining a consistent response accuracy. The system will process simulated network logs and identify potential intrusions using a reasonable detection logic. By using powerful cloud tools such as Cloud Storage, VPCs, and a few built-in monitoring dashboards, this project aims to demonstrates the scalability, elasticity, and pay-as-you-go characteristics of cloud computing. It directly applies course concepts that we've gone over such as virtualization, networking, containerization, and platform-as-a-service management in a real-world cybersecurity context.

Contents

1 Brief Introduction to the Project

Cloud computing allows organizations to deploy scalable, reliable, and cost effective applications without managing physical infrastructure. One of its most important applications is in the field of cybersecurity, in this case through the use of Intrusion Detection Systems (IDS), which monitor and analyze network traffic to detect malicious activity. However, many traditional IDS deployments are either located on site or hosted on fixed cloud virtual machines, which can lead to high idle time costs and limited scalability during sudden traffic spikes.

This projects goal is to address the challenge of building an IDS that is both cost efficient and dynamically scalable in the cloud. It delves into how deploying an IDS on different layers of Google Cloud specifically, Google Compute Engine (IaaS) and Google App Engine (PaaS) affects performance, scalability, and operational cost.

Originality: The originality of this project lies in its comparison analysis of these two cloud deployment models within a cybersecurity context, focusing not only on detection accuracy, but also on cost and scalability factors that are often overlooked in IDS research. Although existing studies evaluate machine learning algorithms for intrusion detection, few examine how cloud architecture choices impact the efficiency of these systems under real world conditions. This project fills that gap by analyzing how a managed, serverless platform can deliver comparable security performance with improved elasticity and reduced cost.

Research Question: Can an Intrusion Detection System deployed on Google App Engine automatically scale to handle variable network traffic more efficiently than a fixed virtual machine on Google Compute Engine, while maintaining consistent detection accuracy?

2 Critical Summary of Existing Literature

Existing research that we found shows that traditional on site IDS solutions seems to struggle with scalability and cost when traffic fluctuates from low use to high use of service. Cloud based approaches have introduced elastic scaling and data driven detection, but often at higher operational cost and more complex development.

- Studies comparing on-premise vs cloud-hosted IDS highlight the benefits of elasticity and distributed detection.[2][3]
- Machine learning–based IDS research shows strong detection capabilities, but limited focus on deployment cost or infrastructure efficiency.[1][4]
- Technical articles in Google Cloud's App Engine and Compute Engine emphasize their scalability trade-offs: App Engine simplifies management with auto-scaling, while Compute Engine offers more control at higher maintenance effort.[2]

Our project builds on that gap by implementing both deployment models side by side and measuring cost, performance, and response accuracy in a realistic test environment. This will help to prove the balance and performance of having an IDS with more scalability.

3 Hypothesis and Objectives

Hypothesis

We will try to deploy an intrusion detection system on Google App Engine that will automatically scale to handle variable network traffic more efficiently than a fixed virtual machine on Google Compute Engine, which hopefully will be resulting in a lower idle-time costs while maintaining consistent response accuracy.

SMART Objectives

1. Deploy the same IDS application to both Google Compute Engine and App Engine within two weeks.
2. Simulate variable network traffic to test automatic scaling on App Engine to ensure it's working.
3. Measure average CPU utilization, response time, and operational cost between both services. This will give us a clear view of how efficient it is.
4. Evaluate whether App Engine maintains detection accuracy within 1% of the Compute Engine baseline.
5. Create a cost and performance comparison dashboard and summarize findings in the final presentation.

4 Methodology

4.1 Cloud Provider and Services

This project uses the Google Cloud Platform (GCP), applying services we learned during lectures and labs.

- **Google Compute Engine (GCE):** A virtual machine service that provides full infrastructure control; it will host the baseline IDS deployment configured manually to measure performance and cost under a fixed resource setup.
- **Google App Engine (GAE):** A platform-as-a-service environment that automatically scales applications based on incoming traffic; it will deploy the same IDS code to evaluate how serverless scaling affects efficiency and accuracy.
- **Cloud Storage (GCS):** Google's object storage service for storing unstructured data; it will hold the simulated network traffic logs and intrusion datasets used to test the IDS.
- **Virtual Private Cloud (VPC) and Firewalls:** A secure, isolated network environment within GCP; these will control communication between the IDS components and protect data flows from unauthorized access, as practiced in Week 3 labs.
- **Cloud Monitoring and Logging:** A managed observability suite that collects metrics and logs from cloud resources; it will record CPU usage, latency, scaling activity, and detection performance for comparing App Engine and Compute Engine deployments.

4.2 System Architecture

The architecture includes:

1. A simulated traffic generator sending network requests.
2. Two IDS deployments: one on GCE and one on GAE.
3. Cloud Storage buckets holding test datasets.
4. Cloud Monitoring dashboards visualizing cost, latency, and scaling metrics.

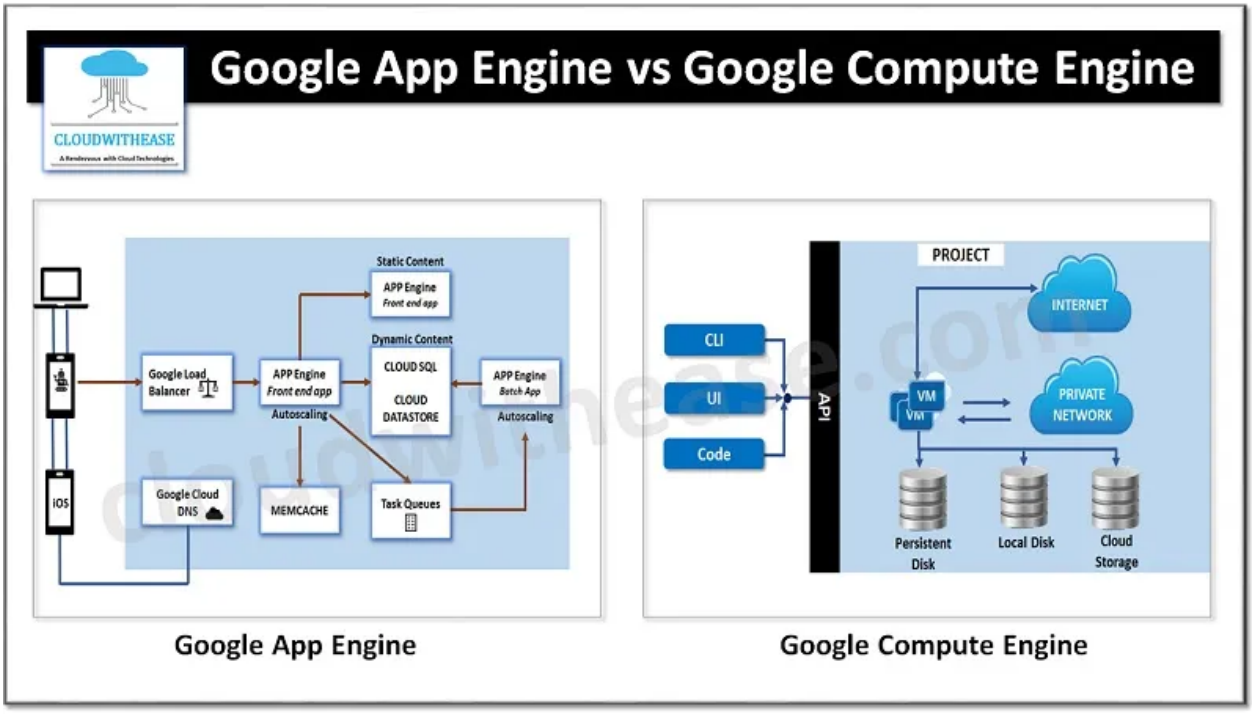

Figure 1: Cloud architecture comparing App Engine (PaaS) and Compute Engine (IaaS) deployments for IDS.

4.3 Data Management

We will have the network traffic logs stored in GCS. The IDS scans requests for suspicious patterns (e.g., repeated login attempts, SQL injection markers) and records results in a database or text file. The access will be secured with IAM roles and firewall restrictions.

4.4 Implementation Plan

1. **Week 1–2:** Configure both environments, upload test data to Cloud Storage using google cloud.
2. **Week 3:** Deploy the IDS app on Compute Engine and App Engine using `app.yaml` configuration that we will create.

3. **Week 4:** Simulate network traffic using a Python script and measure performance for analysis.
4. **Week 5:** Collect metrics (latency, CPU, scaling behavior, cost) in a report for data analysis.
5. **Week 6:** Create visual reports and finalize the comparison to be used in the presentation and final report.

5 Communication and Dissemination

Results will be presented through:

- A final report and presentation.
- A public GitHub repository with configuration files, screenshots, and documentation.
- We will communicate through Discord
- In-person meetings on a weekly schedule.

6 Researcher Preparedness

Existing Skills: Python programming, Google Cloud CLI, basic Linux commands, and prior experience from class labs deploying on GCE, GKE, and GAE. **Relation to Course:** This project integrates all major hands on lessons from Weeks 2–6 and applies them in a practical, measurable experiment.

7 Supervisory and Training Needs

- Learn how to use Cloud Monitoring dashboards effectively.
- Review App Engine scaling configurations and YAML settings like the ones we've used in class.
- Practice cost reporting using the Google Cloud Billing Export feature.

8 Ethical Considerations

No personal or private data will be used in this project and all traffic will be synthetic. Data and storage resources will be protected using IAM least privilege and encryption. Cost transparency will be maintained by publishing billing summaries that we will recieve on Google Cloud. Environmental impact will be minimized by using scale-to-zero features in App Engine.

9 Summary and Conclusions

This project demonstrates how cloud computing concepts learned in class can be applied to a cybersecurity challenge within a reasonable timeframe. By comparing Google App Engine and Compute Engine, it showcases the trade-offs between infrastructure management and automated scalability within these two. The expected outcome is to confirm that a managed PaaS service can

maintain IDS accuracy while reducing idle-time costs, which will eventually be more reasonable considering cost-efficiency This experiment will use key course concepts—such as elasticity, resource pooling, and measured service—but also highlights how cloud computing simplifies secure and scalable application deployment.

References

[1] Yu-Sung Wu, B. Foo, Y. Mei and S. Bagchi, "Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS," 19th Annual Computer Security Applications Conference, 2003.

[2] Belda Garcia, Sergi. "Viability of cloud hosting solutions: distinctions between different hosting solutions." (2025).

[3] J. Smith et al., "Comparing Serverless and Virtual Machine Deployments for Web Applications," IEEE Cloud Computing, 2022.

[4] S. Patel, "Cloud-Based Intrusion Detection: A Review," Journal of Cybersecurity Research, 2023.