**Experiment No: 7**

**Title:** Installation and Configuration of Google App Engine

**Aim:** To understand the concepts of Google Cloud and creating custom app by using it.

**Theory**:

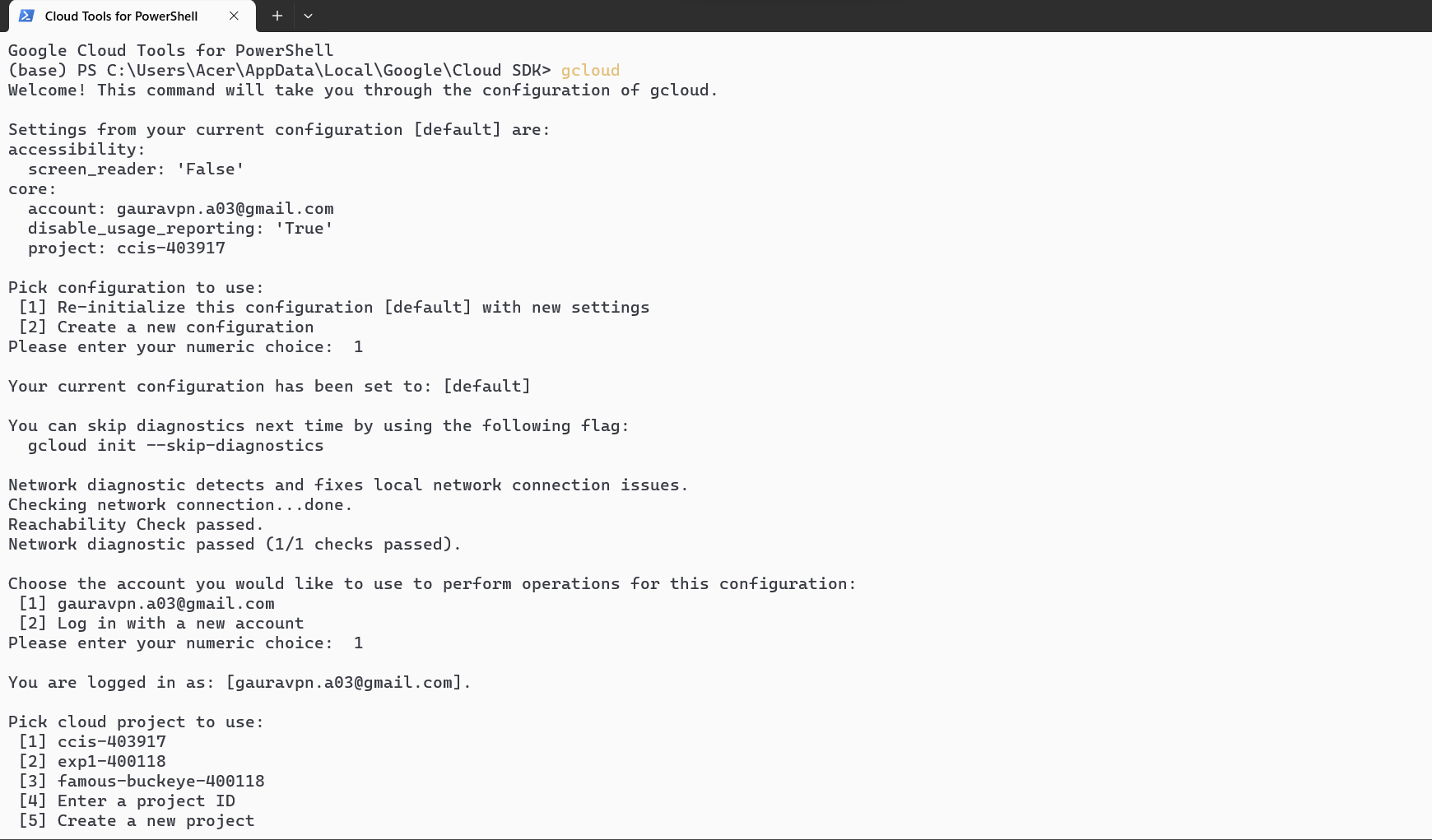
Google App Engine, often referred to as GAE, is a fully managed platform-as-a-service (PaaS) offered by Google Cloud. It allows developers to build and deploy applications on Google's infrastructure without having to manage the underlying server infrastructure. Here's some theory on Google App Engine:

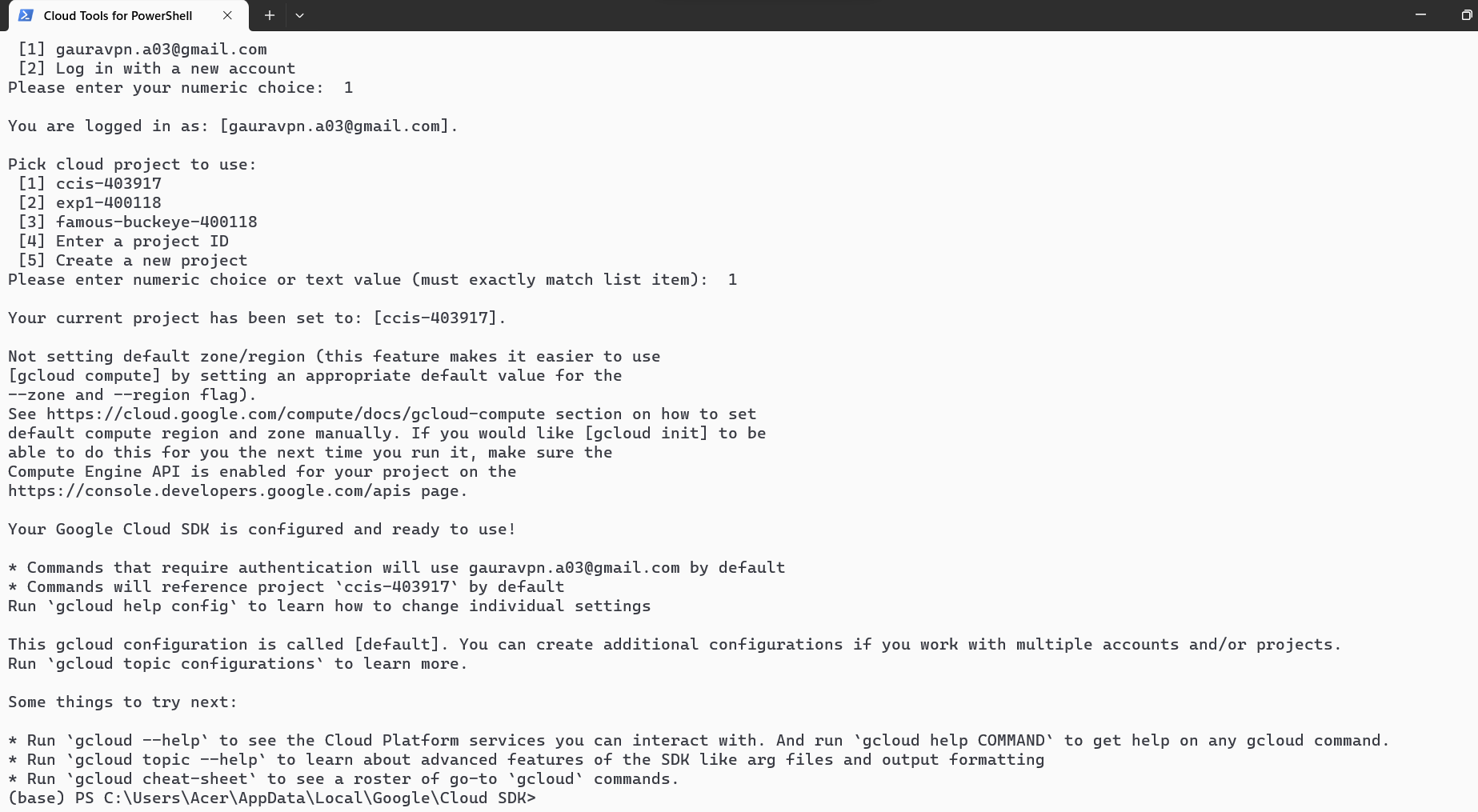
1. **Overview:** Google App Engine is designed to provide a platform for developing and hosting web applications, APIs, and microservices. It abstracts away much of the operational complexity, making it easier for developers to focus on building and deploying their applications.
2. **Scalability:** One of the key features of App Engine is its automatic scaling. Applications hosted on GAE can handle a wide range of traffic loads without manual intervention. As traffic increases, GAE automatically allocates additional resources to handle the load and, as traffic subsides, it scales down accordingly, helping to optimize costs.
3. **Languages and Runtimes:** Google App Engine supports several programming languages, including Python, Java, Go, Node.js, and PHP. Each runtime environment comes with a set of libraries and tools to help developers work with those languages effectively.
4. **Data Storage:** GAE provides data storage options, including Google Cloud Datastore, a NoSQL database for structured data, and Cloud SQL for relational databases. Developers can choose the most appropriate data storage solution based on their application's requirements.
5. **Development Frameworks:** Developers can use popular web development frameworks and tools when building applications for GAE. This includes libraries and frameworks specific to the supported programming languages.
6. **Services and APIs**: Google App Engine provides access to various Google Cloud services, such as Cloud Storage, Pub/Sub, Bigtable, and more. Developers can leverage these services to enhance the functionality of their applications.
7. **Security:** Google Cloud takes security seriously. GAE applications benefit from Google's robust security measures, including data encryption, identity and access management, and DDoS protection.
8. **Versioning and Traffic Splitting:** GAE allows you to deploy multiple versions of your application simultaneously. This feature is useful for A/B testing and gradual feature rollouts. You can also direct specific percentages of traffic to different versions of your application.
9. **Monitoring and Debugging:** Google Cloud provides comprehensive monitoring and debugging tools, including Stackdriver, which helps developers track application performance and troubleshoot issues.
10. **Pricing Model:** Google App Engine offers a pay-as-you-go pricing model. You are charged based on the resources consumed by your application, and there are free quotas available for small-scale usage.
11. **Vendor Lock-In:** As with any cloud platform, using Google App Engine can result in vendor lock-in. It's essential to consider the long-term implications of hosting your application on GAE and whether the benefits outweigh potential lock-in concerns.
12. **Custom Runtimes:** For cases where the standard runtimes don't fit your needs, GAE supports custom runtimes, allowing you to run any language or binary of your choice, which can be useful for highly specialized applications.

In summary, Google App Engine is a powerful platform for building and deploying web applications and services with a focus on scalability and ease of use. It provides a range of services, tools, and integration with other Google Cloud offerings to streamline development and deployment processes.

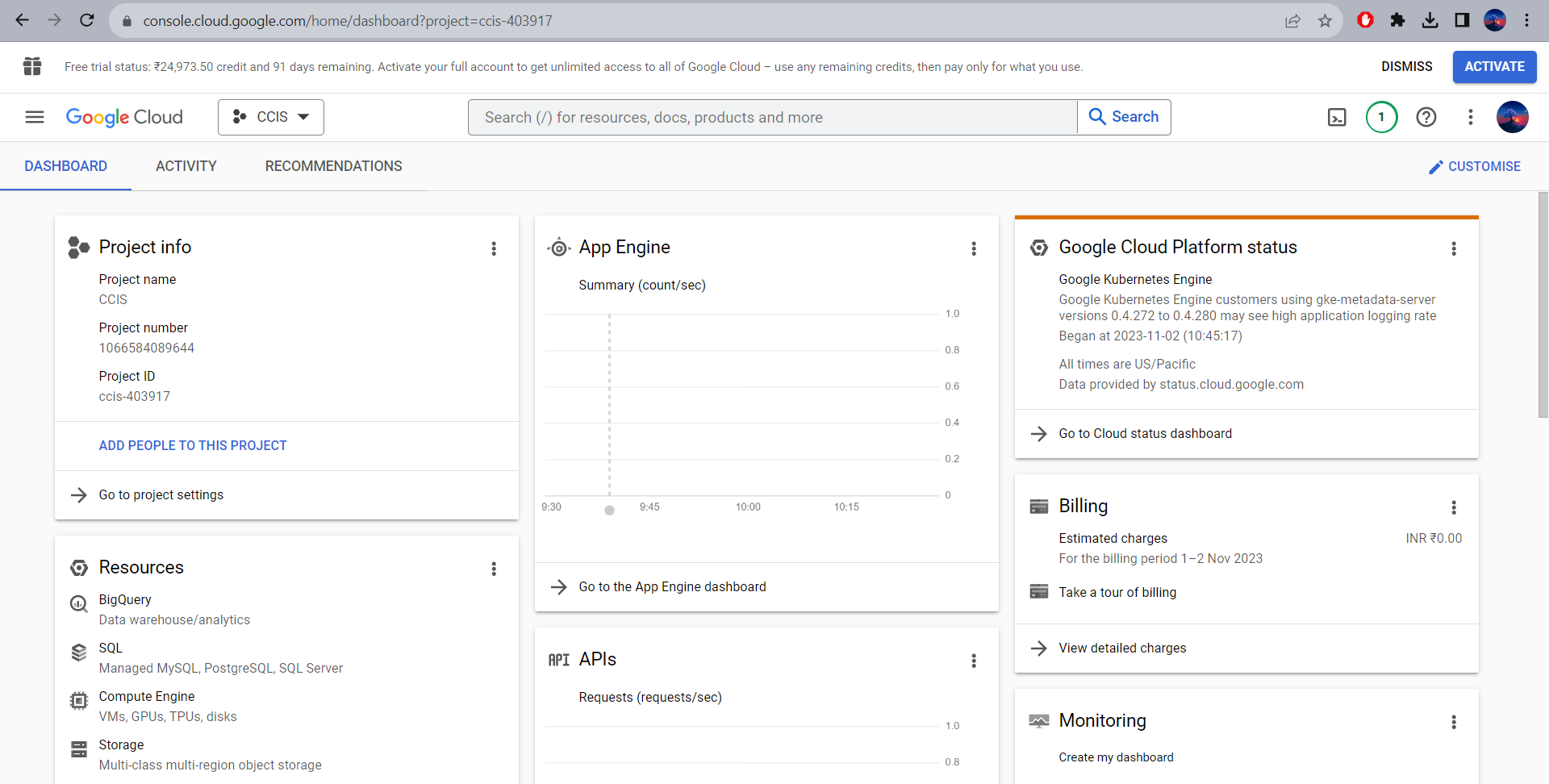
**Practical:**

1. **Gcloud SDK**

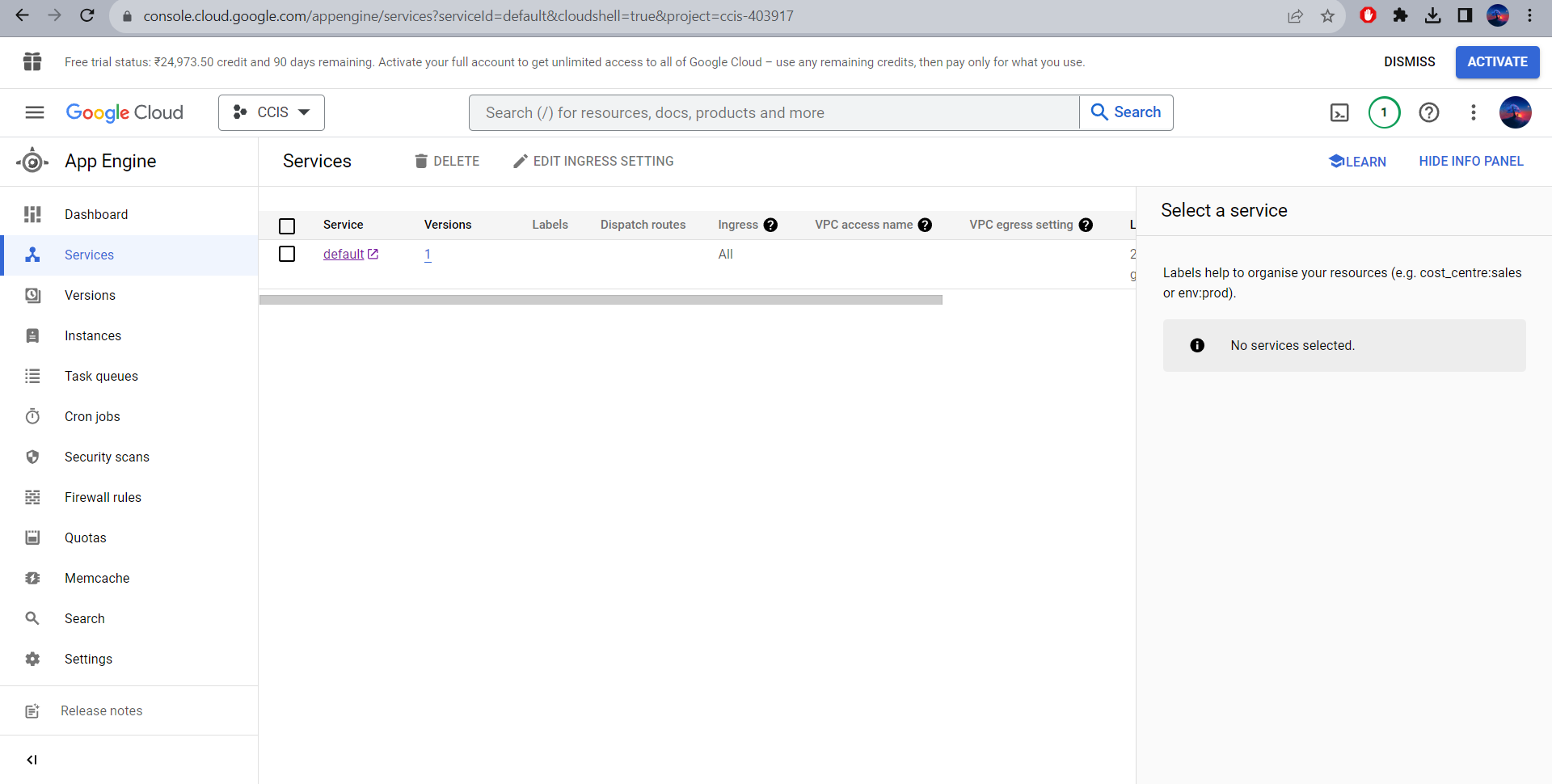
****

****

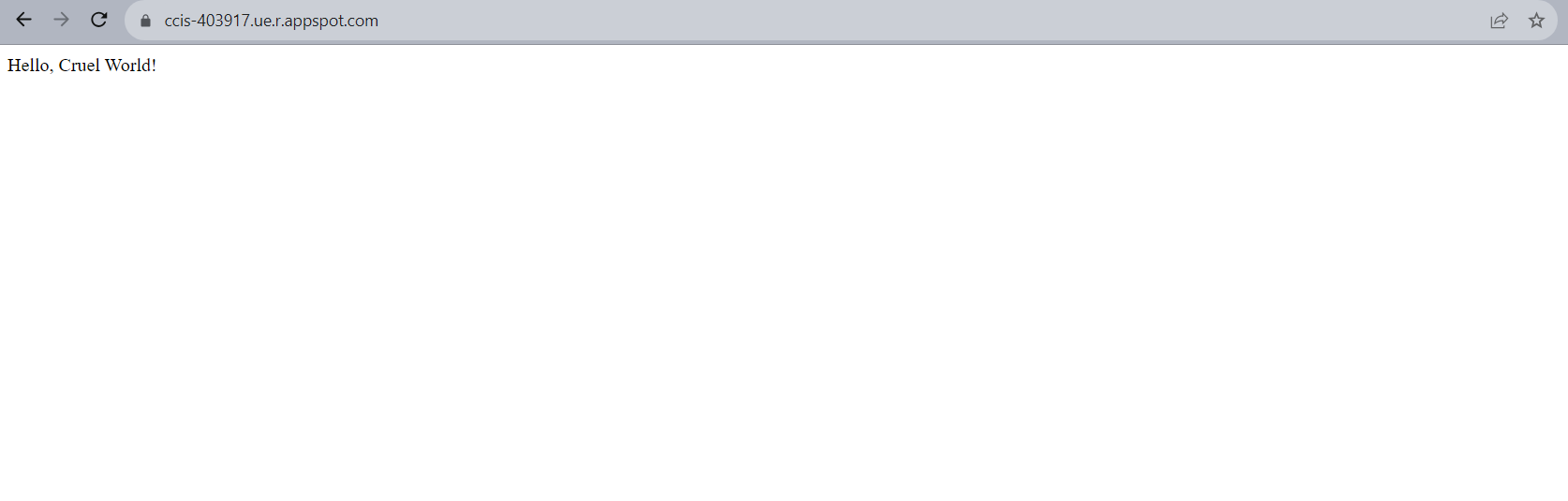
1. **Google Cloud DashBoard**

****

1. **Google Cloud App Engine**

****

1. **Hosted Python File**

****

**Conclusion:** Thus, I have successfully installed Google Cloud SDK and configured it