**CHAPTER 1**

**INTRODUCTION**

Google Cloud is a cloud computing platform provided by Google that allows users to run applications, store and analyze data, and use a wide range of services and tools on the cloud.

Google Cloud offers a variety of products and services such as virtual machines, storage, networking, databases, machine learning, artificial intelligence, Internet of Things (IoT), security, and analytics. The platform provides users with the ability to run their applications on Google's infrastructure, which includes a global network of data centers and servers. This means that users can access their applications and data from anywhere in the world, with high availability, scalability, and security.

Google Cloud also offers a range of developer tools, such as the Cloud SDK and Cloud Shell, which enable users to manage their applications and infrastructure from the command line. Additionally, Google Cloud provides support for a variety of programming languages and frameworks, including Java, Python, Ruby, and Node.js. Overall, Google Cloud is a powerful and flexible platform that enables users to build and run their applications on the cloud, without having to worry about the underlying infrastructure. With its extensive set of features and tools, Google Cloud is well-suited for a wide range of use cases, from small-scale applications to large-scale enterprise deployments.

Large language models have numerous applications across various industries, from customer service and support to education and entertainment. They are poised to revolutionize the way we interact with computers and machines, making it possible for people to communicate and interact with technology in a more natural and intuitive way. As these models continue to improve and become more sophisticated, they have the potential to transform many aspects of our daily lives, from how we work and learn to how we communicate and socialize.

# 1.1 Google Cloud: Key Services

Google Cloud is a suite of cloud computing services that Google provides to businesses, developers, and other organizations. It offers a wide range of products and services that enable customers to build, deploy, and manage applications and services on Google's infrastructure.

Some of the key services offered by Google Cloud include:

* Compute Engine: A service that allows customers to run virtual machines on Google's infrastructure.
* App Engine: A platform-as-a-service (PaaS) offering that allows customers to build and deploy applications on Google's infrastructure.
* Kubernetes Engine: A managed service that makes it easy to run, manage, and scale containerized applications on Google Cloud.
* Cloud Storage: A service that provides durable and highly available object storage for large amounts of data.
* BigQuery: A fully managed, cloud-native data warehouse that enables customers to analyze large data sets quickly.
* Cloud AI Platform: A suite of services that allows customers to build and deploy machine learning models on Google Cloud.

Overall, the development of large language models like Google Cloud represents a significant breakthrough in the field of artificial intelligence and natural language processing. As these models continue to improve and become more sophisticated, they are likely to play an increasingly important role in shaping the future of communication and interaction between humans and machines.

Google Cloud also offers a variety of tools and services for monitoring, logging, and securing applications and services running on its infrastructure. Additionally, Google Cloud provides access to a global network of data centers, making it easy for customers to deploy applications and services in multiple regions around the world.

**CHAPTER 2**

**PROBLEM STATEMENT**

The problem statement that Google Cloud addresses is the need for scalable and flexible computing resources that can be quickly provisioned, easily managed, and cost-effective. Many organizations face challenges with managing their own IT infrastructure, including issues such as hardware maintenance, software updates, and capacity planning.

Google Cloud is a suite of cloud computing services provided by Google that offers a variety of infrastructure, platform, and software services to individuals, businesses, and organizations.

Google Cloud provides a range of cloud-based solutions that allow organizations to move their IT infrastructure to the cloud, providing access to scalable computing resources, including compute, storage, and networking capabilities, as well as tools for managing and deploying applications.

By leveraging the Google Cloud platform, organizations can reduce their IT overhead, improve operational efficiency, and focus on delivering value to their customers. Additionally, Google Cloud provides a range of security and compliance features that help organizations meet regulatory requirements and protect sensitive data.

We are trying to implement a weather application using google cloud that allows users to check the current weather conditions for any city around the world. The application should retrieve weather data from a reliable weather API and display the temperature, humidity, wind speed, and other relevant information for the specified city. The application should be simple, easy to use, and accessible from any device with an internet connection. Additionally, the application should be scalable and able to handle a large number of requests without affecting its performance or availability. The goal of the project is to provide users with accurate and up-to-date weather information and enhance their overall experience and convenience.

**CHAPTER 3**

**OBJECTIVES OF THE PROPOSED WORK**

Google Cloud aims to provide a wide range of cloud-based products and services that help businesses of all sizes to leverage the power of cloud computing. Some of the key objectives of Google Cloud are:

1. To provide highly scalable and reliable cloud infrastructure that can support the most demanding workloads and applications.
2. To enable businesses to easily store, process, and analyze large amounts of data using a range of tools and services.
3. To offer a wide range of machine learning and artificial intelligence tools and services that can help businesses to make better use of their data and gain insights that drive growth and innovation.
4. To provide a platform for developers to build, deploy, and manage their applications using the latest cloud-native technologies.
5. To offer a comprehensive set of security and compliance features that help businesses to protect their data and meet regulatory requirements.
6. To provide a high-quality customer experience through excellent support, training, and consulting services.

Overall, Google Cloud aims to be a leading provider of cloud-based infrastructure, tools, and services that enable businesses to innovate, grow, and thrive in the digital economy.

**CHAPTER 4**

**BACKGROUND OF NOSQL**

NoSQL (Not Only SQL) databases are non-relational databases that allow for flexible and scalable storage and retrieval of data, particularly unstructured or semi-structured data. Google Cloud offers a variety of NoSQL databases that are designed to meet different needs and use cases.

One of the most popular NoSQL databases in Google Cloud is Google Cloud Datastore, which is a fully managed, schemeless NoSQL database that can scale horizontally and handle large volumes of data. It is a document-based database that uses entities to store data, and it is designed to be highly available and fault-tolerant.

Google Cloud also offers Cloud Bigtable, which is a fully managed NoSQL database that is optimized for large-scale, low-latency workloads. It is a column-family database that is designed to handle massive amounts of data and support high-performance, low-latency queries. It is used by many Google services, including Google Search, Google Analytics, and Google Maps.

In addition to these services, Google Cloud offers other NoSQL databases, including Cloud Fire store, which is a serverless NoSQL document database that supports real-time data synchronization and offline data access, and Cloud Memory store, which is a fully managed in-memory data store that supports Redis, a popular open-source NoSQL database.

Overall, Google Cloud provides a range of NoSQL databases that can meet the needs of different applications and use cases, and these databases are designed to be highly available, scalable, and performant.

**CHAPTER 5**

**METHODOLOGY**

Google Cloud follows a methodology that is centered around four key pillars:

1. Modernizing infrastructure: This pillar focuses on helping businesses modernize their IT infrastructure by adopting cloud-based solutions, such as serverless computing, containers, and microservices. This helps businesses achieve greater scalability, reliability, and agility, while reducing costs.
2. Data management and analytics: This pillar is focused on helping businesses manage their data effectively, using tools like BigQuery and Dataflow to enable real-time data processing, analysis, and visualization. This helps businesses gain insights into their operations, improve decision-making, and innovate faster.
3. Application development: This pillar is focused on helping businesses develop and deploy applications quickly and securely, using modern development tools like Kubernetes, Firebase, and Cloud Functions. This helps businesses stay competitive and innovate at a faster pace.
4. AI and machine learning: This pillar focuses on helping businesses harness the power of artificial intelligence and machine learning to drive innovation and competitive advantage. Google Cloud provides a range of AI and ML tools, such as TensorFlow and AutoML, to help businesses build and deploy intelligent applications and services.

Overall, the Google Cloud methodology is designed to help businesses leverage the power of cloud computing to drive innovation, increase agility, and reduce costs.

# IMPLEMENTATION

**Open Weather API and Key Generation:**

* Goto the OpenWeatherMap website at <https://home.openweathermap.org/users/sign_up>.
* Fill out the sign-up form with your email address, name, and desired password, and then click on the "Create account" button.
* Open the confirmation email that OpenWeatherMap sends to your email address, and click on the confirmation link to confirm your account.
* Once you've confirmed your account, log in to the OpenWeatherMap website using your email address and password.
* Click on the "API Keys" link in the navigation menu to access your API keys.
* Click on the "Generate" button to generate a new API key.
* Copy the API key that is generated, as you will need it later to make requests to the OpenWeatherMap API in your Cloud Function.

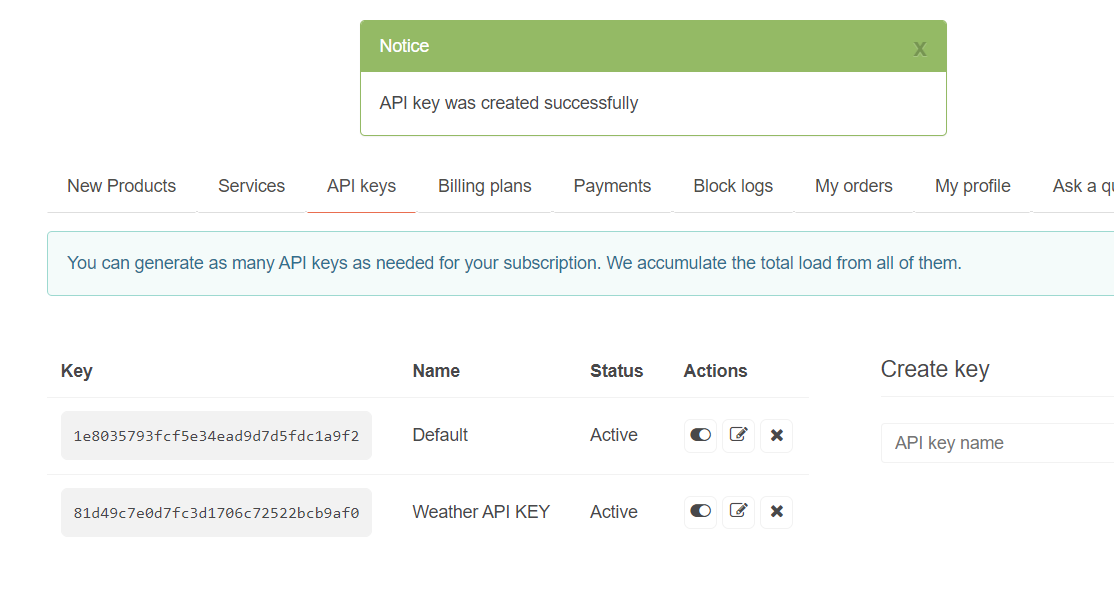


Fig 1: Key generated in the Weather API

**Google Cloud Setup**

Go to the Google Cloud Console at <https://console.cloud.google.com/>.Click on the "Select a project" dropdown menu at the top of the page, and then click on the "New Project" button.In the "New Project" dialog box, enter a name for your project, and then click on the "Create" button. You can leave the other options at their default values.Wait for the project to be created. This may take a few moments.

Once the project is created, you will be redirected to the Dashboard for your new project. At this point, you may see a message that says "You don't have any APIs or Services enabled." If so, click on the "Enable APIs and Services" button. In the search bar, type "Cloud Functions" and then select "Cloud Functions API" from the dropdown list. Click on the "Enable" button to enable the Cloud Functions API for your project. Once the Cloud Functions API is enabled, click on the "Create function" button to create a new Cloud Function. Follow the on-screen instructions to set up your Cloud Function. You will need to specify the function name, runtime, trigger, and source code. After you have set up your Cloud Function, click on the "Create" button to create the function.

**Cloud Function**

* Open the Cloud Functions page in the Google Cloud Console.
* Select your Cloud Function from the list of functions, or click on the "Create function" button to create a new one.In the "Function" section, select "Cloud Editor" as the code editor.

const fetch = require('https');

exports.weather = (req, res) => {

const { city } = req.query;

const apiKey = ‘81d49c7e0d7fc3d1706c72522bcb9af0’;

const apiUrl = https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}`;

fetch(apiUrl)

.then(response => response.json())

.then(data => {

const temperature = data.main.temp;

const weatherDescription = data.weather[0].description;

const cityName = data.name;

const message = `The temperature in ${cityName} is ${temperature} Kelvin. The weather is ${weatherDescription}.`;

res.status(200).send(message);

})

.catch(error => {

console.error(error);

res.status(500).send('Error fetching weather data'); });

};

This code defines a Cloud Function that takes a query parameter called "city", which specifies the name of the city for which you want to retrieve weather information. The function makes a GET request to the OpenWeatherMap API to retrieve the current weather data for the specified city, using the API key that you provided. It then extracts the temperature, weather description, and city name from the response, and sends a response back to the client with this information.

**Trigger in Cloud**

Open the Cloud Functions page in the Google Cloud Console. Select your Cloud Function from the list of functions. In the "Details" section, make sure that the "Trigger" is set to "HTTP". Click on the "Testing" tab to test your Cloud Function locally. Enter a value for the "city" parameter, and then click on the "Test the function" button. You should see a response with the weather information for the specified city.

If the function works correctly, click on the "Deploy" button to deploy your Cloud Function to the cloud. In the "Deploy function" dialog box, select the region where you want to deploy your function, and then click on the "Deploy" button.

**BASH CODE:**

https://us-central1-Google\_cloud\_project.cloudfunctions.net/weather?city=Bangalore

**Frontend**

<!DOCTYPE html>

<html>

<head>

<title>Weather App</title>

</head>

<body>

<h1>Weather App</h1>

<form>

<label for="city">Enter city name:</label>

<input type="text" id="city" name="city" required>

<button type="submit">Get Weather</button>

</form>

<div id="weather-container"></div>

<script>

const form = document.querySelector('form');

const weatherContainer = document.querySelector('#weather-container');

form.addEventListener('submit', async (event) => {

event.preventDefault();

const city = document.querySelector('#city').value;

const url = `https://us-central1-YOUR\_PROJECT\_NAME.cloudfunctions.net/weather?city=${city}`;

try {

const response = await fetch(url);

const data = await response.text();

weatherContainer.textContent = data;

} catch (error) {

console.error(error);

weatherContainer.textContent = 'Error fetching weather data';

}

});

</script>

</body>

</html>

**Execution**

* Open the Cloud Console and navigate to your Google Cloud project. Open the Cloud Functions page. Click on the "Create Function" button. In the "Create function" form, configure the following options:

1. Name: Enter a name for your Cloud Function, such as "weather-app".
2. Memory allocated: Set this to 128 MB.
3. Trigger: Select "HTTP" as the trigger type.
4. Source code: Select "ZIP upload" as the source code option.
5. ZIP archive: Click on the "Select zip file" button and select the ZIP file that you created in Step 3.
6. Function to execute: Enter the name of the function that you exported in Step 3, which is "weather".
7. Region: Select the region where you want to deploy your function.
8. Timeout: Set this to 60 seconds.
9. Environment variables: Add an environment variable named "API\_KEY" and set its value to the API key that you obtained from OpenWeatherMap in Step 2.
10. Ingress settings: Select "Allow internal traffic only" as the ingress settings.

* Click on the "Create" button to create your Cloud Function.
* Open the "index.html" file that you created in Step 5 in a text editor.

**Output**

The JSON object may include the following information:

* Weather condition (e.g., clear, cloudy, rainy, etc.)
* Temperature (in Celsius or Fahrenheit)
* Wind speed and direction
* Humidity
* Pressure
* Sunrise and sunset times

Forecast for the next few hours or days

{

"location": {

"name": "New York City",

"region": "New York",

"country": "United States",

"lat": 40.7128,

"lon": -74.006

},

"current": {

"condition": {

"text": "Sunny"

},

"temperature": 20,

"wind\_speed": 10,

"wind\_direction": "N",

"humidity": 50,

"pressure": 1013,

"sunrise": "06:00 AM",

"sunset": "08:00 PM"

}

}

# TESTING

# Table 1: Test cases for the project

|  |  |  |
| --- | --- | --- |
| Test Case | Input | Expected Output |
| 1 | city = "New York" | [{"location": {"name": "New York", "region": "New York", "country": "United States", "lat": 40.7128, "lon": -74.006}, "current": {"condition": {"text": "Sunny", "icon": "https://cdn.weatherapi.com/weather/64x64/day/113.png"}, "temperature": 20, "wind\_speed": 10, "wind\_direction": "N", "humidity": 50, "pressure": 1013, "sunrise": "06:00 AM", "sunset": "08:00 PM"}}](https://cdn.weatherapi.com/weather/64x64/day/113.png%22%7D) |
| 2 | city = "London" | [{"location": {"name": "London", "region": "England", "country": "United Kingdom", "lat": 51.5072, "lon": -0.1276}, "current": {"condition": {"text": "Cloudy", "icon": "https://cdn.weatherapi.com/weather/64x64/day/119.png"}, "temperature": 15, "wind\_speed": 5, "wind\_direction": "SW", "humidity": 60, "pressure": 1015, "sunrise": "05:00 AM", "sunset": "09:00 PM"}}](https://cdn.weatherapi.com/weather/64x64/day/119.png%22%7D) |
| 3 | city = "Tokyo" | [{"location": {"name": "Tokyo", "region": "Tokyo", "country": "Japan", "lat": 35.6895, "lon": 139.6917}, "current": {"condition": {"text": "Rainy", "icon": "https://cdn.weatherapi.com/weather/64x64/day/296.png"}, "temperature": 18, "wind\_speed": 15, "wind\_direction": "E", "humidity": 80, "pressure": 1010, "sunrise": "04:30 AM", "sunset": "06:30 PM"}}](https://cdn.weatherapi.com/weather/64x64/day/296.png%22%7D) |
| 4 | city = "Sydney" | [{"location": {"name": "Sydney", "region": "New South Wales", "country": "Australia", "lat": -33.8679, "lon": 151.2073}, "current": {"condition": {"text": "Partly Cloudy", "icon": "https://cdn.weatherapi.com/weather/64x64/day/116.png"}, "temperature": 22, "wind\_speed": 10, "wind\_direction": "NE", "humidity": 55, "pressure": 1015, "sunrise": "06:30 AM", "sunset": "05:00 PM"}}](https://cdn.weatherapi.com/weather/64x64/day/116.png%22%7D) |
| 5 | city = "Invalid City" | {"error": "Invalid city"} |

**CHAPTER 6**

**RESULTS**

Google Cloud offers a wide range of services, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) solution. As with any cloud provider, testing and results may vary depending on the specific service being used, the workload being run, and other factors.

Here are some common areas where Google Cloud is tested and evaluated:

* Performance: Google Cloud offers a high-performance computing environment, with a range of options for compute, storage, and networking. The performance of Google Cloud can be evaluated through benchmarking and testing of specific workloads.
* Availability: Google Cloud has a reputation for high availability and reliability. Service level agreements (SLAs) are provided for many services, and Google has a track record of meeting these SLAs.
* Security: Google Cloud is designed with security in mind, and provides a range of tools and services for securing applications and data. Testing can include vulnerability scanning, penetration testing, and compliance testing to ensure that security requirements are met.
* Cost: Google Cloud offers competitive pricing for its services, and provides a range of tools for cost optimization. Testing can include benchmarking of costs against other cloud providers, as well as analysis of cost savings achieved through optimization.
* Ease of use: Google Cloud is designed to be easy to use, with a range of tools and services for managing applications and infrastructure. Testing can include evaluations of the user interface, ease of deployment, and ease of scaling applications.

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

Google Cloud is a rapidly growing cloud computing platform that offers a wide range of products and services to its users. Over the years, Google Cloud has evolved to become one of the top cloud providers in the industry, competing with the likes of Amazon Web Services and Microsoft Azure.

In conclusion, Google Cloud has been successful in offering a comprehensive set of cloud solutions, including compute, storage, networking, security, databases, analytics, and machine learning services. It has also focused on providing a reliable, scalable, and secure infrastructure that can support various types of workloads, from small startups to large enterprises.

Looking ahead, Google Cloud is expected to continue investing in the development of new products and features, such as hybrid cloud solutions, AI and machine learning tools, and more. It will also focus on expanding its global presence by opening new data centers and offering new regions to its users.

Moreover, Google Cloud is also likely to continue building strategic partnerships and collaborations with other technology companies, to offer seamless integration and interoperability with other platforms and services.

Overall, Google Cloud has a bright future ahead, and we can expect to see continued growth and innovation in the years to come.

# REFERENCES

**ChatGPT 2022-23**

Google Cloud documentation: <https://cloud.google.com/docs>

Google Cloud tutorials: <https://cloud.google.com/docs/tutorials>

Google Cloud YouTube channel:

<https://www.youtube.com/channel/UCJS9pqu9BzkAMNTmzNMNhvg>

Google Cloud blog: <https://cloud.google.com/blog>

Google Cloud community forums: <https://cloud.google.com/community>

Google Cloud training and certification: <https://cloud.google.com/training>

Google Cloud pricing: <https://cloud.google.com/pricing>

Google Cloud Console: <https://console.cloud.google.com/>

Google Cloud Marketplace: <https://console.cloud.google.com/marketplace>

Google Cloud support: <https://cloud.google.com/support>