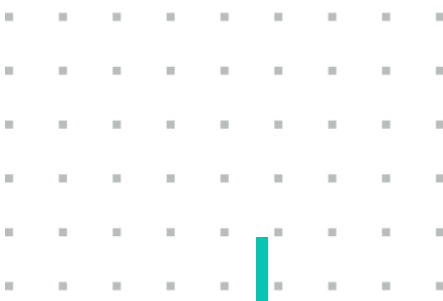


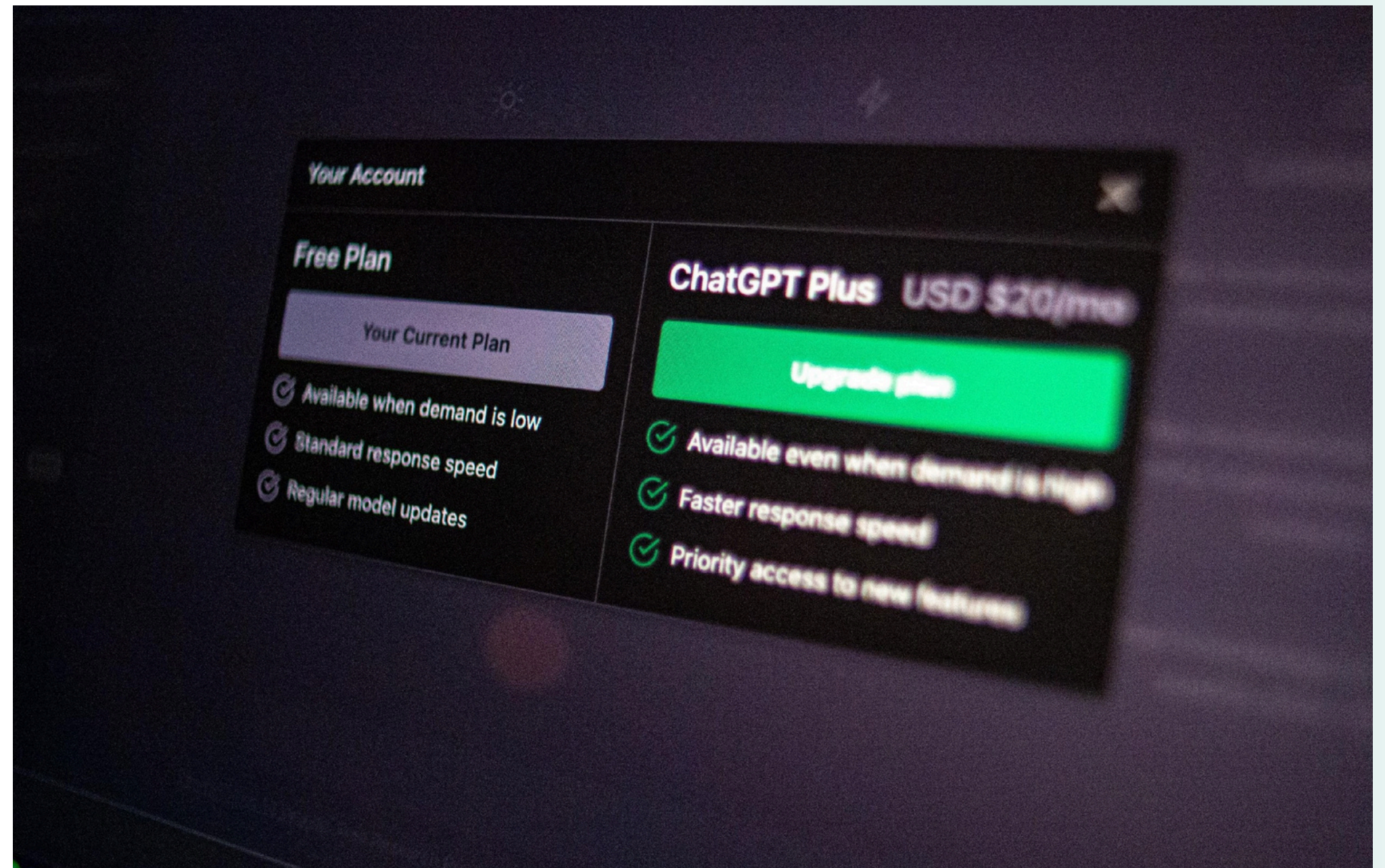
# Hosting Chatbot on an EC2 instance





# Index

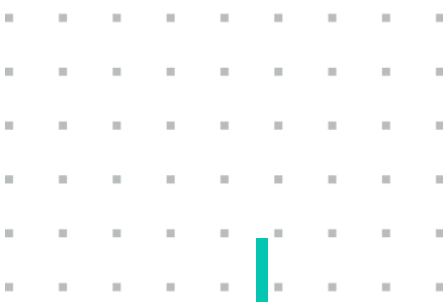
- Introduction
- Chatroom Application
- Working of the Chatroom Application
- Amazon EC2
- Hosting of our Project
- Unveiling Our Chatbot





# Introduction

In today's digital world, a lot of information is frequently found in PDF documents, making it difficult for users to effectively extract insights. Our research provides a web-based chat application that aims to solve this issue by enabling smooth interaction with PDFs. Our application enables users to ask questions, upload PDF files, and get real-time conversational responses. Utilising new technologies like vector embeddings, text extraction, and similarity search, our platform improves PDF content's usability and accessibility. This project demonstrates the development process, key features, and overall objective of providing customers with an easy-to-use platform for natural language interaction-based insight extraction from PDF documents.

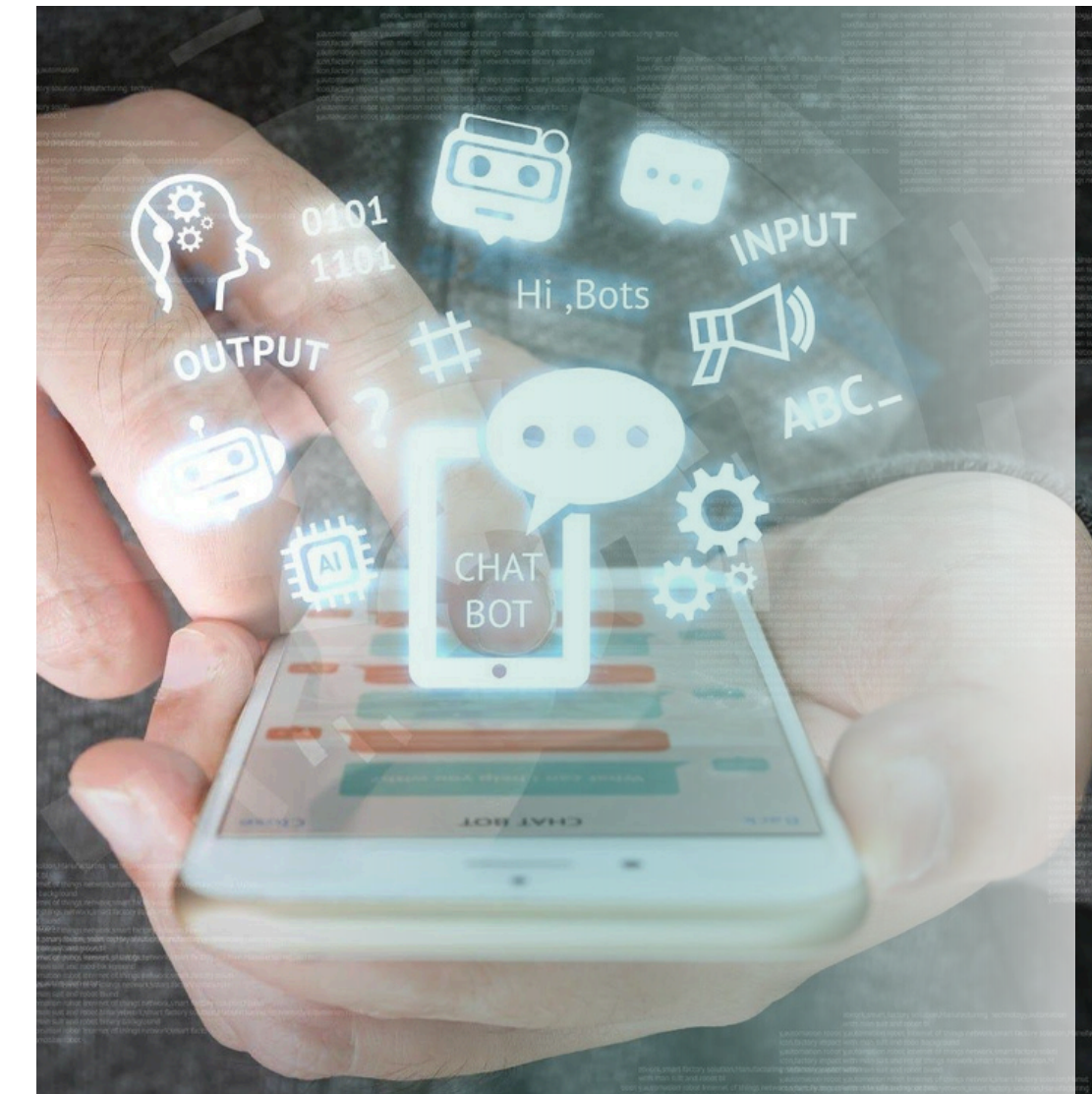


# Chatbot Application

Chatbots are AI-powered virtual assistants you can chat with! Imagine a conversation window where you ask questions and get answers. Langchain, a developer toolkit, simplifies building these chatbots. Here's the magic:

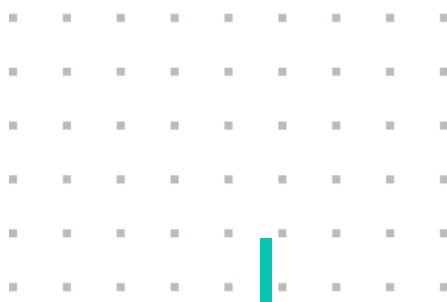
1. You Ask a Question: You type your question in the chat window.
2. Langchain Captures Input: Langchain receives your question and acts as a bridge.
3. Connects to Powerful AI: Langchain connects to a large language model (like me!) that can understand your question.
4. AI Analyzes and Responds: The AI analyzes your question and generates a response.
5. Langchain Delivers Answer: Langchain delivers the AI's answer back to you in the chat.

Langchain streamlines building chatbots, letting you focus on the conversation design, while the AI handles the complex understanding and response generation.



# Working of the Chatbot

- Upload PDFs and Extract Text: You upload PDFs, and the code extracts all text content.
- Split Text into Chunks: The text is split into smaller manageable pieces.
- Create a Search Index: The code creates a searchable index using a powerful AI model (FAISS) for efficient searching.
- User Asks a Question: You type your question about the PDFs.
- Search for Relevant Parts: The system searches the index using your question to find related parts of the PDFs.
- Answer the Question: A large language model, like Bard, analyzes the retrieved parts and answers your question in detail, referencing the PDFs.



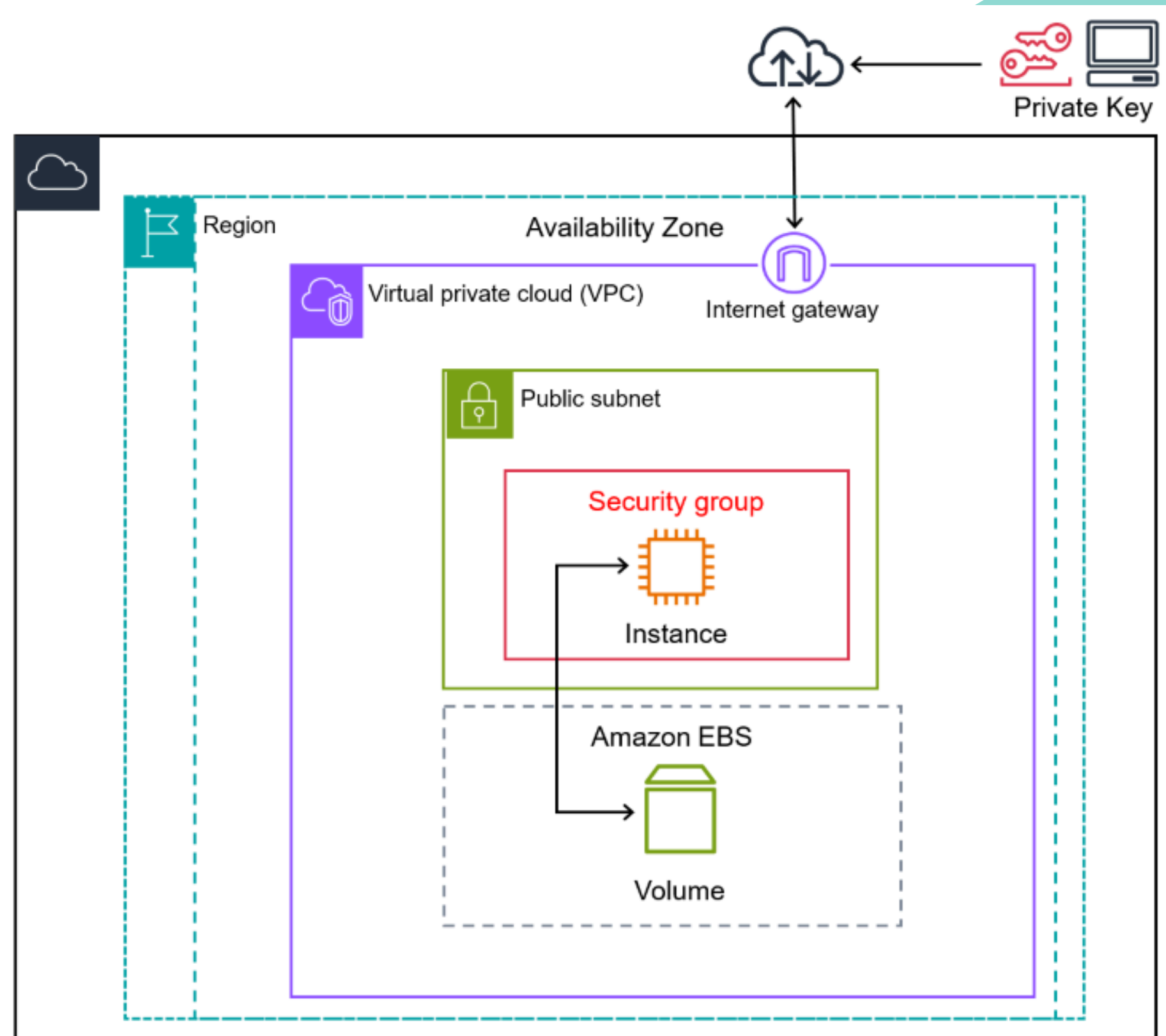


# Amazon EC2

Amazon EC2, or Elastic Compute Cloud, is a service by Amazon Web Services (AWS) that lets you rent virtual machines (instances). These virtual computers, called instances, can be booted up with various configurations of processing power, memory, storage, and operating system. This allows you to quickly set up and scale compute capacity for your applications without having to manage physical hardware. Here are some of the reasons why you might use EC2 in your presentation:

- Scalability: Easily add or remove computing power as needed.
- Cost-effective: Pay only for what you use.
- Variety of options: Choose from a wide range of instance types to meet your specific needs.
- Fast setup: Get virtual servers up and running in minutes.

EC2 is a popular choice for businesses of all sizes looking for a flexible and cost-effective way to run their applications in the cloud.



EC2 Basic Architecture

# Steps to Deploy our Chatbot on AWS EC2 Instance:

1

## Create an EC2 instance

Creating an EC2 instance on AWS is the first step of deployment in the Cloud. We enter the specifications of the instance and save the .pem (key-pair) file required to access the instance

2

## Create a virtual environment

Using the .pem file and SSH command, we access the EC2 instance and create a virtual environment and eventually install all the required libraries and dependencies.

3

## Clone the Github Repository

Now, we create and clone our Git with following commands thus, push our code to Github:

- `git add {project folder}`
- `git commit -m "model added"`
- `git remote add origin{repo clone link}`
- `git push -u origin master`

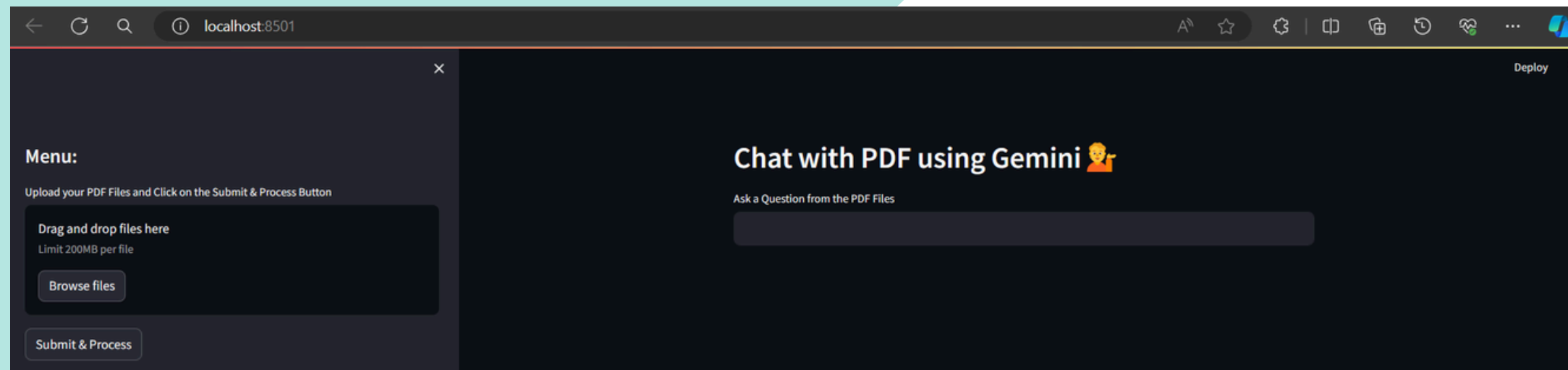
4

## Checking the performance

Finally, we run the Chatroom py file and open the UI. To check whether the working is fine, we upload a simple PDF file and enter related query, to which we get pretty accurate and satisfactory results, meaning success.



# Unveiling our Chatbot



## Chat with PDF using Gemini 🧠

Ask a Question from the PDF Files

What is BART? Describe its features.

Reply: BART (Bidirectional Auto-Regressive Transformers) is a denoising autoencoder that maps a corrupted document to the original document it was derived from. It is implemented as a sequence-to-sequence model with a bidirectional encoder over corrupted text and a left-to-right autoregressive decoder. For pre-training, BART optimizes the negative log likelihood of the original document.

BART has several features that distinguish it from other pre-trained language models:

- **Bidirectional encoder:** BART uses a bidirectional encoder to process the input text, which allows it to capture both left-to-right and right-to-left dependencies. This is in contrast to models like GPT, which only use a left-to-right encoder.
- **Autoregressive decoder:** BART uses an autoregressive decoder to generate the output text, which allows it to condition on the previously generated tokens. This is in contrast to models like BERT, which use a masked language model objective that does not require autoregressive generation.
- **Denoising pre-training:** BART is pre-trained on a denoising objective, which involves corrupting the input text and then training the model to reconstruct the original text. This objective encourages the model to learn to identify and correct errors in the input text, which can be beneficial for downstream tasks such as machine translation and question answering.

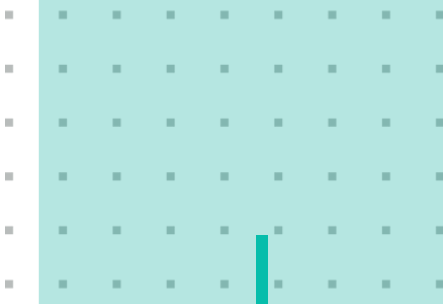
# Group Members

**Devapangu Abhishek (21BDS016)**

**Hosur Sai Kartik (21BDS021)**

**Nischay Kondai (21BDS045)**

**Pratik Raj (21BDS047)**



תודה רבה

Hebrew

Ευχαριστώ

Greek

Спасибо

Russian

Danke

German

Merci

French

धन्यवादः

Sanskrit

நன்றி

Tamil

شكراً

Arabic

ಧನ್ಯವಾದಗಳು

Kannada

Thank You

English

നന്ദി

Malayalam

Grazie

Italian

ధన్యవాదాలు

Telugu

આભાર

Gujarati

多謝

Traditional Chinese

Gracias

Spanish

ਧੰਨਵਾਦ

Punjabi

धन्यवाद

Hindi & Marathi

多谢

Simplified Chinese

<https://sites.google.com/site/animeshchaturvedi07>

Obrigado

Portuguese

ありがとうございました

Japanese

ขอบคุณ

Thai

감사합니다

Korean