VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI 590018

Report on

“Memo-pad”

By

Umang Goel (1BM20CS176)

Vishnu Kumar (1BM20CS190)

Afrah Mahmud (1BM20CS196)

Under the Guidance of

Sowmya T

Assistant Professor

Department of CSE

BMS College of Engineering

Work carried out at

Department of Computer Science and Engineering BMS College of Engineering
(Autonomous college under VTU)
P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019

2023-2024

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Cloud Computing assignment titled "Memo-pad**"** has been carried out by Umang Goel (1BM20CS176), Vishnu Kumar (1BM20CS190) and Afrah Mahmud (1BM20CS196)) during the academic year 2023-2024.

Signature of the guide
SOWMYA T
Assistant Professor
Department of Computer Science and Engineering BMS College of Engineering, Bangalore

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Umang Goel (1BM20CS176), Vishnu Kumar (1BM20CS190) and Afrah Mahmud (1BM20CS196), students of 7th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this assignment work entitled "Memo-pad" has been carried out by us under the guidance of **SOWMYA T,** Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore, during the academic semester Sept 2023-Jan 2024. We also declare that to the best of our knowledge and belief, the assignment reported here is not from part of any other report by any other students.

Signature of the Candidates

Umang Goel (1BM20CS176)

Vishnu Kumar (1BM20CS190)

Afrah Mahmud (1BM20CS196)

Table of Contents

1. Introduction

1.1 Introduction to Cloud Computing

The U.S. National Institute of Standards and Technology (NIST) defines cloud computing as:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet).

Some of its features include:

- On-demand self service
- Broad Network access
- Resource Pooling
- Rapid elasticity
- Measured service
- Improved Performance
- Reduced Costs
- Outsourced Management
- Reliability
- Multi-tenancy

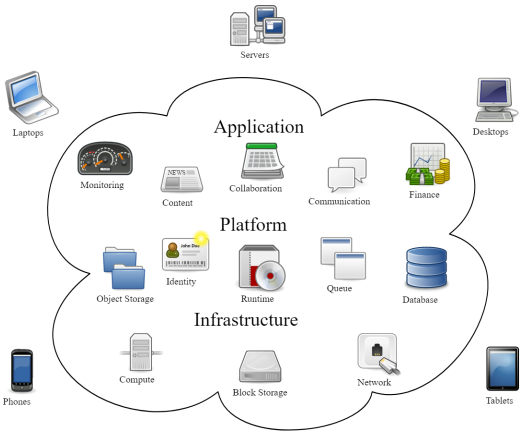


Fig 1: Cloud Computing [1]

1.2 Problem Statement

a. Create a web application with a backend of your choice.
b. Containarise the application and host it on the cloud.

2. Docker and Kubernetes

Docker is an open-source containerization platform. It is a toolkit that makes it easier, safer and faster for developers to build, deploy and manage containers. It uses a client-server architecture with simple commands and automation through a single API. Docker also provides a toolkit that is commonly used to package applications into immutable container images by writing a Dockerfile and then running the appropriate commands to build the image using the Docker server. Developers can create containers without Docker but the Docker platform makes it easier to do so. These container images can then be deployed and run on any platform that supports containers.

Docker architecture

Docker uses a client-server architecture. The Docker *client* talks to the Docker *daemon*, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon *can* run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface. Another Docker client is Docker Compose, which lets you work with applications consisting of a set of containers.

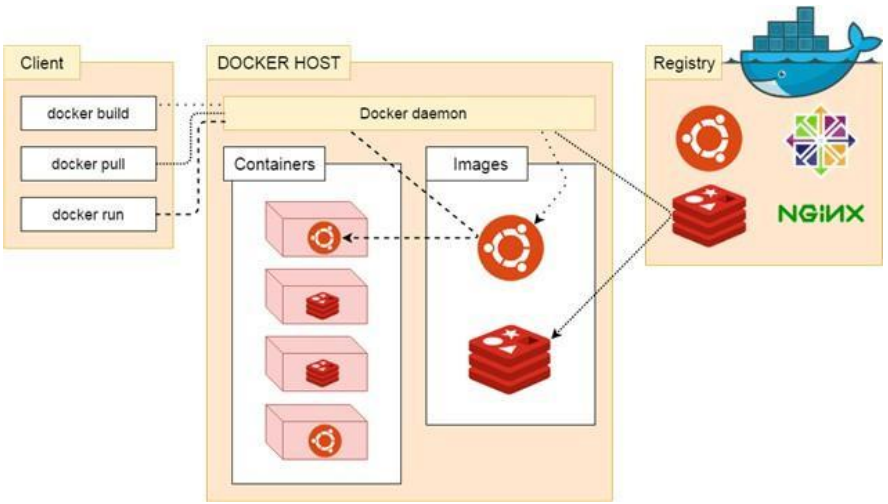


Fig 2. Docker architecture [5]

Docker daemon: The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

Docker client: The Docker client (docker) i uses commands and REST APIs to communicate with the Docker Daemon (Server). When a client runs any docker command on the docker client terminal, the client terminal sends these docker commands to the Docker daemon. Docker daemon receives these commands from the docker client in the form of command and REST API's request.

Docker Host: Docker Host is used to provide an environment to execute and run applications. It contains the docker daemon, images, containers, networks, and storage.

Docker Desktop: Docker Desktop is an easy-to-install application for your Mac, Windows or Linux environment that enables you to build and share containerized applications and microservices.

Docker registries: Docker Registry manages and stores the Docker images. There are 2 types of registries:

Public registry-Also called as DockerHub. Private registry-It is used to share images within the enterprise.

Docker objects: There are the following Docker Objects -

Docker Images: Docker images are the read-only binary templates used to create Docker Containers. It uses a private container registry to share container images within the enterprise and also uses a public container registry to share container images within the whole world. Metadata is also used by docket images to describe the container's abilities.

Docker Containers: Containers are the structural units of Docker, which is used to hold the entire package that is needed to run the application. The advantage of containers is that it requires very less resources.

Benefits of Docker

- Lightweight portability: Containerized applications can move from any environment to another (wherever Docker is operating), and they will operate regardless of the OS.
- Agile application development: Containerization makes it easier to adopt CI/CD processes and take advantage of agile methodologies, such as DevOps. For example, containerized apps can be tested in one environment and deployed to another in response to fast-changing business demands.
- Scalability: Docker containers can be created quickly and multiple containers can be managed efficiently and simultaneously.
- Version control: Like git, Docker has a built version control system. Docker containers work just like GIT repositories, allowing you to commit changes to your Docker images and version control them.
- Build the app only once: An application inside a container can run on a system that has Docker installed. So there is no need to build and configure apps multiple times on different platforms.

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications that orchestrates container runtime systems across a cluster of networked resources. Kubernetes can be used with or without Docker. Kubernetes

was originally developed by Google. It bundles a set of containers into a group that it manages on the same machine to reduce network overhead and increase resource usage efficiency. It is useful for DevOps teams since it offers service discovery, load balancing within the cluster, automated rollouts and rollbacks, self-healing of containers that fail, and configuration management. Plus, Kubernetes is a critical tool for building robust DevOps CI/CD pipelines.

A Kubernetes cluster includes a container designated as a control plane that schedules workloads for the rest of the containers — or worker nodes — in the cluster.The master node determines where to host applications (or Docker containers), decides how to put them together and manages their orchestration. By grouping containers that make up an application into clusters, Kubernetes facilitates service discovery and enables management of high volumes of containers throughout their life cycles.

Kubernetes Architecture

Kubernetes is an architecture that offers a loosely coupled mechanism for service discovery across a cluster. A Kubernetes cluster has one or more control planes, and one or more compute nodes. Overall, the control plane is responsible for managing the overall cluster, exposing the application program interface (API), and for scheduling the initiation and shutdown of compute nodes based on a desired configuration. Each of the compute nodes runs a container runtime like Docker along with an agent, kubelet, which communicates with the control plane. Each node can be bare metal servers, or on-premises or cloud-based virtual machines (VMs).

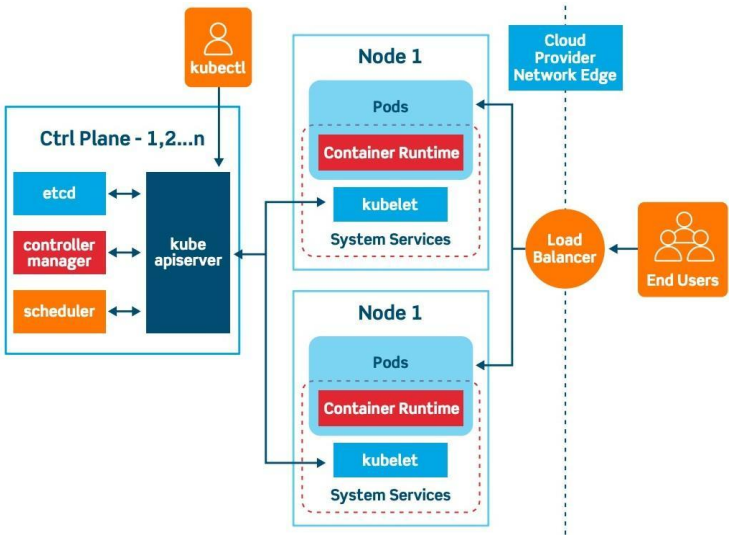


Fig 3. Kubernetes architecture [6]

The main components of a Kubernetes cluster include:

Nodes: Nodes are VMs or physical servers that host containerized applications. Each node in a cluster can run one or more application instance. There can be as few as one node, however, a typical Kubernetes cluster will have several nodes

Image Registry: Container images are kept in the registry and transferred to nodes by the control plane for execution in container pods.

Pods: Pods are where containerized applications run. They can include one or more containers and are the smallest unit of deployment for applications in a Kubernetes cluster.

A Kubernetes control plane is the control plane for a Kubernetes cluster. Its components include:

kube-apiserver. As its name suggests the API server exposes the Kubernetes API, which is communications central. External communications via command line interface (CLI) or other user interfaces (UI) pass to the kube-apiserver, and all control planes to node communications also goes through the API server.

etcd: The key value store where all data relating to the cluster is stored. etcd is highly available and consistent since all access to etcd is through the API server. Information in etcd is generally formatted in human-readable YAML.

kube-scheduler: When a new Pod is created, this component assigns it to a node for execution based on resource requirements, policies, and 'affinity' specifications regarding geolocation and interference with other workloads.

kube-controller-manager: Although a Kubernetes cluster has several controller functions, they are all compiled into a single binary known as kube-controller-manager.

Nodes are the machines, either VMs or physical servers, where Kubernetes place Pods to execute. Node components include:

kubelet: Every node has an agent called kubelet. It ensures that the container described in PodSpecs are up and running properly.

kube-proxy: A network proxy on each node that maintains network nodes which allows for the communication from Pods to network sessions, whether inside or outside the cluster, using operating system (OS) packet filtering if available.

container runtime: Software responsible for running the containerized applications. Although Docker is the most popular, Kubernetes supports any runtime that adheres to the Kubernetes CRI

Other components

Pods: By encapsulating one (or more) application containers, pods are the most basic execution unit of a Kubernetes application. Each Pod contains the code and storage resources required for execution and has its own IP address. Pods include configuration options as well. Typically, a Pod contains a single container or few containers that are coupled into an application or business function and that share a set of resources and data.

Deployments: A method of deploying containerized application Pods. A desired state described in a Deployment will cause controllers to change the actual state of the cluster to achieve that state in an orderly manner. Learn more about Kubernetes Deployments.

ReplicaSet: Ensures that a specified number of identical Pods are running at any given point in time.

Cluster DNS: serves DNS records needed to operate Kubernetes services.

Container Resource Monitoring: Captures and records container metrics in a central database.

Benefits of Kubernetes

- Automated operations: Kubernetes comes with a powerful API and command line tool, called kubectl, which handles a bulk of the heavy lifting that goes into container management by allowing you to automate your operations. The controller pattern in Kubernetes ensures applications/containers run exactly as specified.
- Infrastructure abstraction: Kubernetes manages the resources made available to it on your behalf. This frees developers to focus on writing application code and not the underlying compute, networking, or storage infrastructure.
- Service health monitoring: Kubernetes monitors the running environment and compares it against the desired state. It performs automated health checks on services and restarts containers that have failed or stopped. Kubernetes only makes services available when they are running and ready.
- Automated rollouts and rollbacks: It rolls out application changes and monitors application health for any issues, rolling back changes if something goes wrong.
- Storage orchestration: Automatically mounts a persistent local or cloud storage system of choice as needed to reduce latency — and improve user experience.
- Dynamic volume provisioning: Allows cluster administrators to create storage volumes without having to manually make calls to their storage providers or create objects.

3. Cloud Solutions for Application Hosting

Cloud hosting makes applications and websites accessible using cloud resources. Unlike traditional hosting, solutions are not deployed on a single server. Instead, a network of connected virtual and physical cloud servers hosts the application or website, ensuring greater flexibility and scalability.

Key features

- Applications and solutions are deployed on a cloud network rather than an on-premises, single server.
- Resources scale to user needs.
- Organizations only pay for the resources they use.
- Cloud hosting can support SQL (including MySQL) or NoSQL databases.
- Solutions are automated and controlled using APIs, web portals, and mobile apps.

1) Amazon Web Services

Amazon Web Services offers cloud web hosting solutions that provide businesses, non-profits, and governmental organizations with low-cost ways to deliver their websites and web applications.
Amazon EC2 provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers and allows maximum scalability and availability for websites and web applications. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.
AWS can be used for the following reasons:

- Broad platform support
- Data Centers worldwide
- Scalable from day one
- Flexible pricing models

2) Microsoft Azure

Azure is also commonly used as a platform for hosting databases in the cloud. It provides a broad range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications or run existing applications in the public cloud.. In addition, the platform is frequently used for backup and disaster recovery. It provides for:

- Highly available and massively scalable platform for your applications and APIs and accelerated application deployment.
- Autoscaling of your cloud environment to optimize costs and improve performance.
- Integrated health monitoring and load balancing with dashboards and real-time alerts.
- Excellent development experience using the Azure SDK, which integrates seamlessly with Visual Studio.

3) Google Cloud Platform

Google Cloud is a suite of public cloud computing services offered by Google. The platform includes a range of hosted services for compute, storage and application development that run on Google hardware. Google Cloud services can be accessed by

software developers, cloud administrators and other enterprise IT professionals over the public internet or through a dedicated network connection.
Some of its features include:

- Outstanding Availability and 99.99% Uptime along with Monitoring
- High throughput and Performance optimization with Network Service Tiers
- Leading Global Infrastructure, Easy setup and high-level security

Thanks to the platform's simplicity and click-to-deploy hosting configurations, getting started is easy. Once hosting is up, your website will load at lightning speed and almost always remain available.

4. Tools used

4.1 Tools used for front end design: HTML5 and CSS3
4.2 Tools used for back end: MySQL
4.3 Tools used for hosting the application: Amazon Web Services

5. Detailed design

Application design

Memo-pad App is the app that is used to maintain our day-to-day tasks or list everything that we have to do, with the most important tasks at the top of the list, and the least important tasks at the bottom. It is helpful in planning our daily schedules. Memo-pads are a way to keep track of your tasks in an organized way.

A Memo-pad can be useful for the following:

- Increased Productivity.
- Helps With Motivation.
- Helps Complete Tasks.
- Deal With Multitasking Requirements Effectively.
- Keep Track of Work.
- Time Management.
- Helps You Stay Focused.

In the Memo-pad application, the user can add their tasks and its accompanying description for the tasks which are yet to be completed. The users can specify the date on which task has to be completed and the priority as well ranging from low to high. Once the task has been completed, the status of the task can be changed to 'completed'. The users can view the list of completed tasks and uncompleted tasks. These tasks can be updated and deleted as well. The users can search the tasks using the references - task name, description, date or priority.

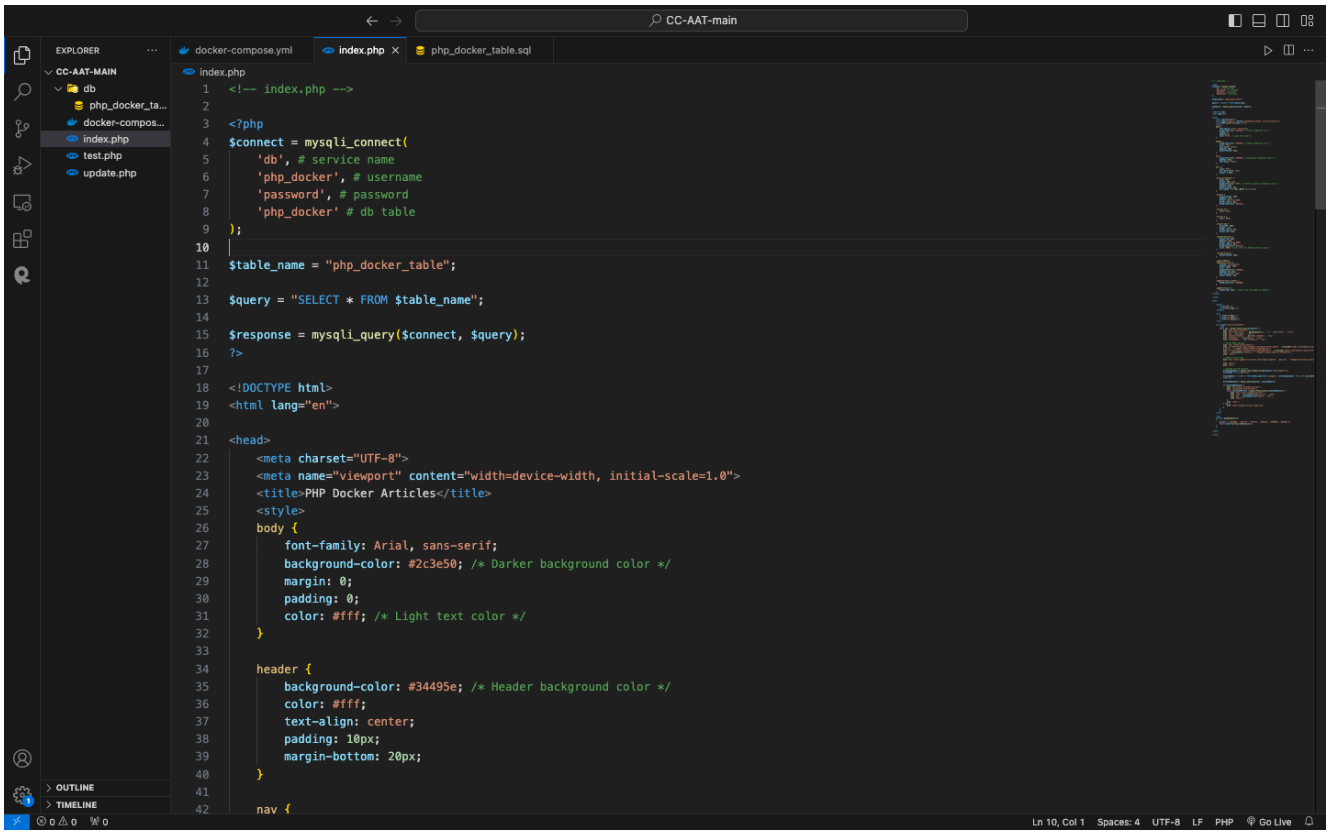
Steps to host the application

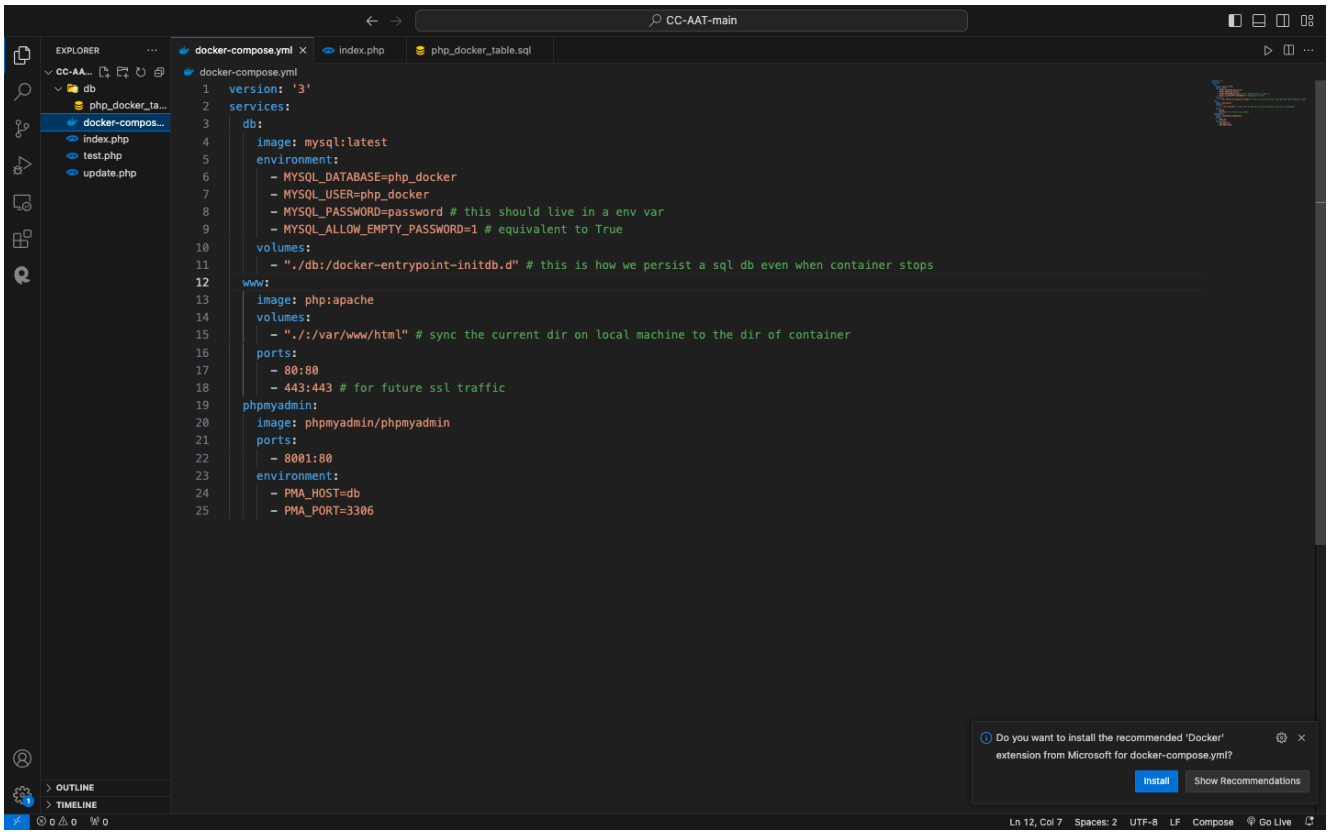
- Create a Flask application and set up Nginx as a reverse-proxy to accept the requests from the user.

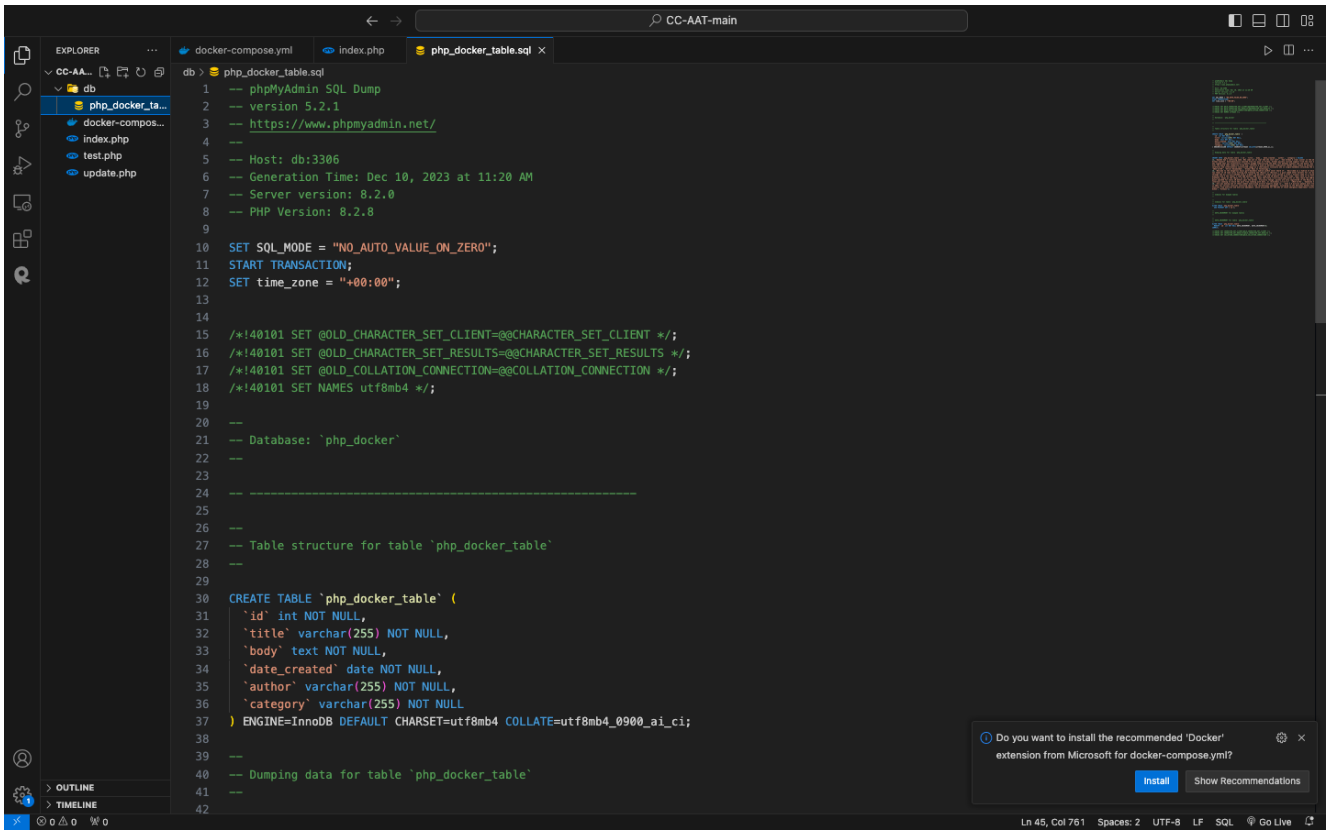
- Write a Dockerfile and a docker-compose.yml file to define and configure the container for your flask app.
- Commit your code to a Git repository, such as GitHub, to version control and manage your code.
- Create an AWS account and sign in to the AWS Management Console.
- Follow the instructions to create an EC2 (Elastic Compute Cloud) instance, selecting the appropriate instance type and configuring any required settings.
- Connect to the EC2 console to manage and monitor your instance.
- Install Docker, Git, and Docker Compose on the EC2 instance using the appropriate commands for your operating system.
- Clone the Git repository containing your code onto the EC2 instance.
- Navigate to the repository directory and use the docker-compose build and docker-compose up command to build and run the Docker container for your flask app

6. Implementation Details

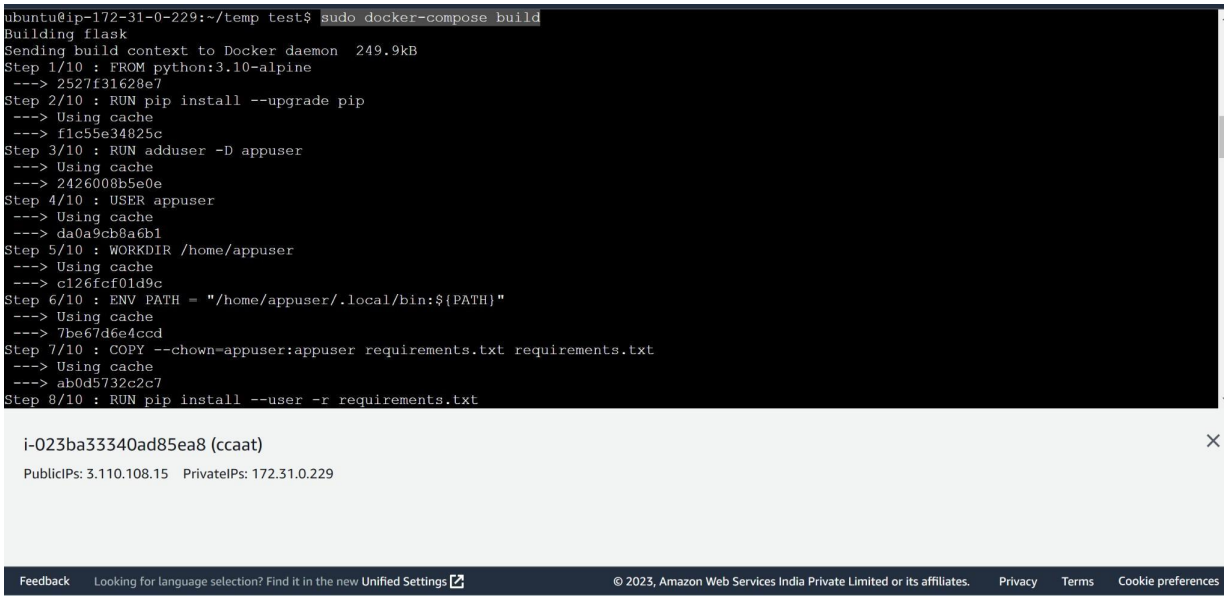
Application code snapshot along with database connectivity

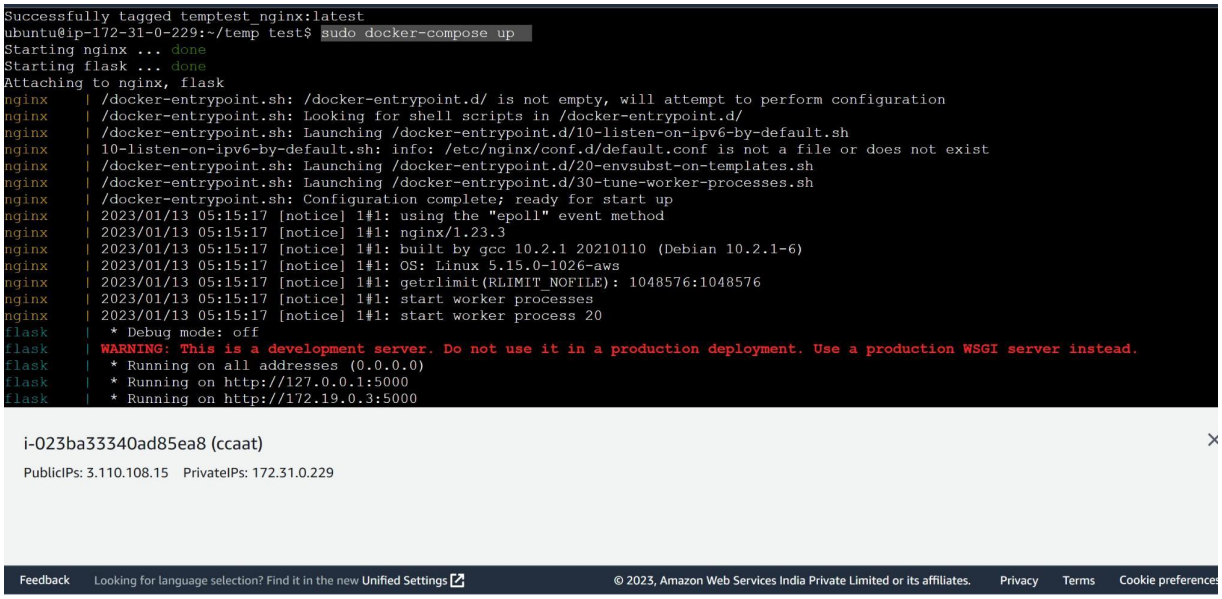



CC-AAT-main
EXPLORER
docker-compose.yml
index.php
php_docker_table.sql
CC-AA...
db
php_docker_ta...
docker-compos...
index.php
test.php
update.php
OUTLINE
TIMELINE
docker-compose.yml
version: '3'
services:
db:
image: mysql:latest
environment:
- MYSQL_DATABASE=php_docker
- MYSQL_USER=php_docker
- MYSQL_PASSWORD=password # this should live in a env var
- MYSQL_ALLOW_EMPTY_PASSWORD=1 # equivalent to True
volumes:
- "./db:/docker-entrypoint-initdb.d" # this is how we persist a sql db even when container stops
www:
image: php:apache
volumes:
- "./:/var/www/html" # sync the current dir on local machine to the dir of container
ports:
- 80:80
- 443:443 # for future ssl traffic
phpmyadmin:
image: phpmyadmin/phpmyadmin
ports:
- 8001:80
environment:
- PMA_HOST=db
- PMA_PORT=3306
Do you want to install the recommended 'Docker' extension from Microsoft for docker-compose.yml?
Install
Show Recommendations
Ln 12, Col 7
Spaces: 2
UTF-8
LF
Compose
Go Live

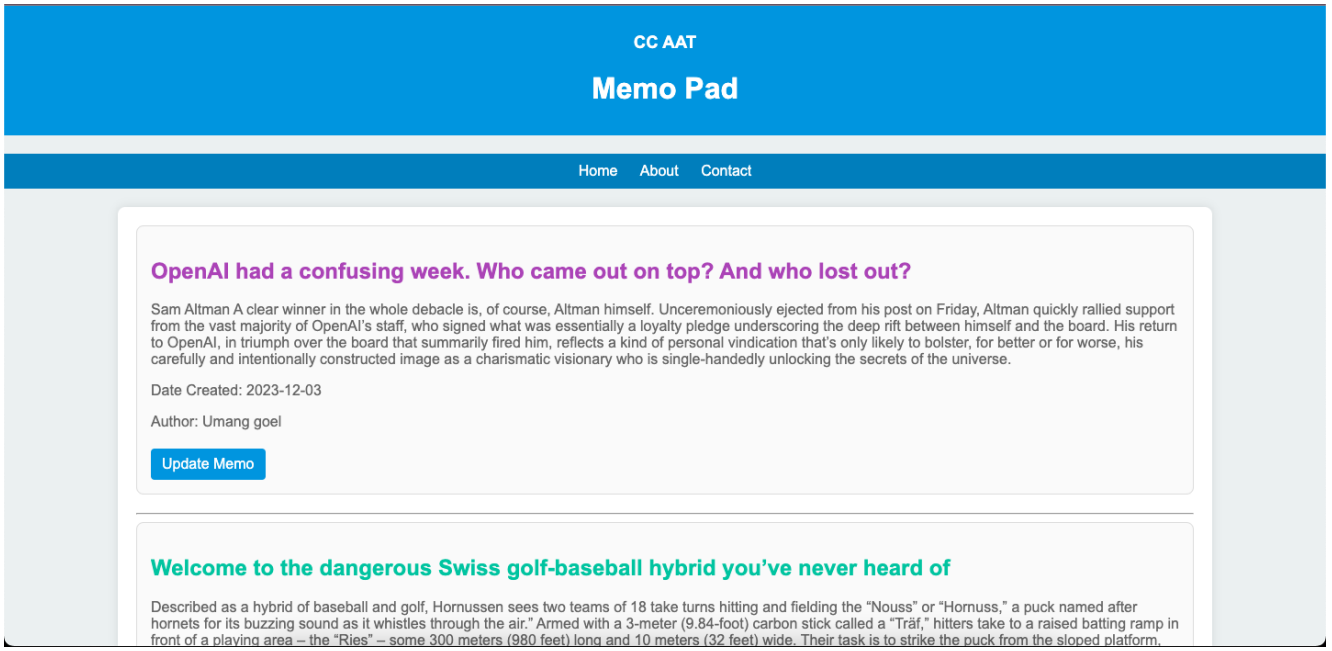

CC-AAT-main
EXPLORER
docker-compose.yml
index.php
php_docker_table.sql
db > php_docker_table.sql
CC-AA...
db
php_docker_ta...
docker-compos...
index.php
test.php
update.php
OUTLINE
TIMELINE
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: db:3306
-- Generation Time: Dec 10, 2023 at 11:20 AM
-- Server version: 8.2.0
-- PHP Version: 8.2.8
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
--
-- Database: `php_docker`
--
-- --
--
-- Table structure for table `php_docker_table`
--
CREATE TABLE `php_docker_table` (
`id` int NOT NULL,
`title` varchar(255) NOT NULL,
`body` text NOT NULL,
`date_created` date NOT NULL,
`author` varchar(255) NOT NULL,
`category` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
--
-- Dumping data for table `php_docker_table`
--
Do you want to install the recommended 'Docker' extension from Microsoft for docker-compose.yml?
Install
Show Recommendations
Ln 45, Col 761
Spaces: 2
UTF-8
LF
SQL
Go Live

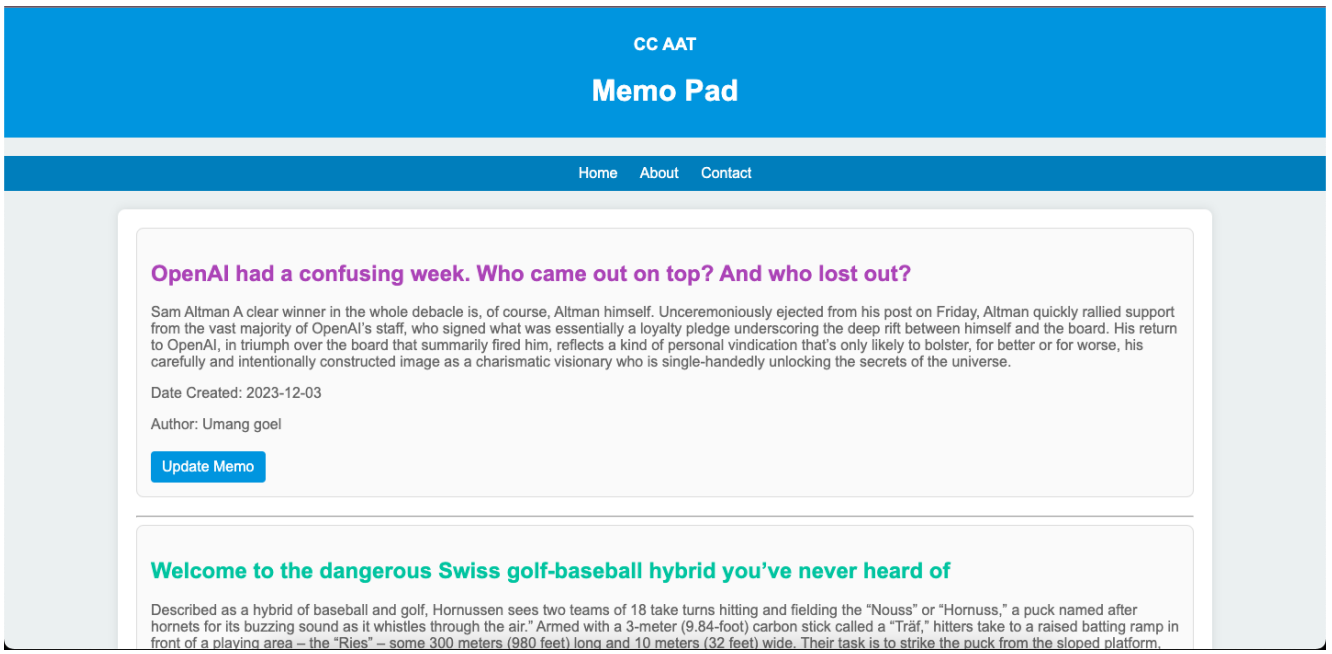
Commands used to host the application with snapshots

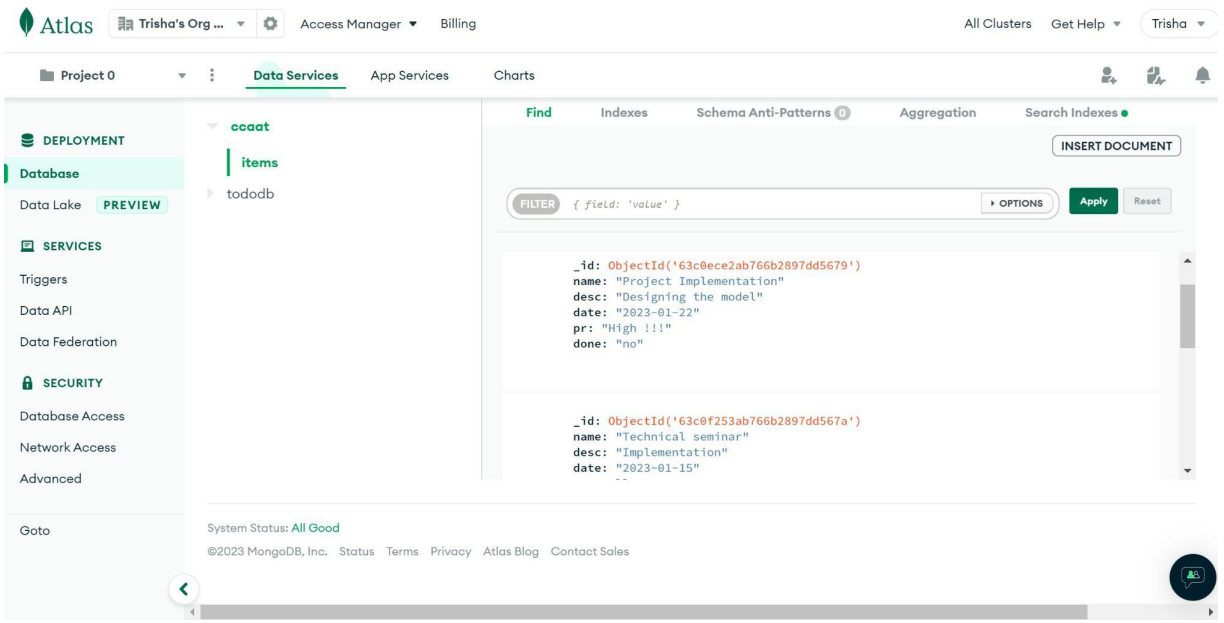




7. Results


CC AAT
Memo Pad
Home
About
Contact
OpenAI had a confusing week. Who came out on top? And who lost out?
Sam Altman A clear winner in the whole debacle is, of course, Altman himself. Unceremoniously ejected from his post on Friday, Altman quickly rallied support from the vast majority of OpenAI's staff, who signed what was essentially a loyalty pledge underscoring the deep rift between himself and the board. His return to OpenAI, in triumph over the board that summarily fired him, reflects a kind of personal vindication that's only likely to bolster, for better or for worse, his carefully and intentionally constructed image as a charismatic visionary who is single-handedly unlocking the secrets of the universe.
Date Created: 2023-12-03
Author: Umang goel
Update Memo
Welcome to the dangerous Swiss golf-baseball hybrid you've never heard of
Described as a hybrid of baseball and golf, Hornussen sees two teams of 18 take turns hitting and fielding the "Nouss" or "Hornuss," a puck named after hornets for its buzzing sound as it whistles through the air." Armed with a 3-meter (9.84-foot) carbon stick called a "Träf," hitters take to a raised batting ramp in front of a playing area – the "Ries" – some 300 meters (980 feet) long and 10 meters (32 feet) wide. Their task is to strike the puck from the sloped platform.


CC AAT
Memo Pad
Home
About
Contact
OpenAI had a confusing week. Who came out on top? And who lost out?
Sam Altman A clear winner in the whole debacle is, of course, Altman himself. Unceremoniously ejected from his post on Friday, Altman quickly rallied support from the vast majority of OpenAI's staff, who signed what was essentially a loyalty pledge underscoring the deep rift between himself and the board. His return to OpenAI, in triumph over the board that summarily fired him, reflects a kind of personal vindication that's only likely to bolster, for better or for worse, his carefully and intentionally constructed image as a charismatic visionary who is single-handedly unlocking the secrets of the universe.
Date Created: 2023-12-03
Author: Umang goel
Update Memo
Welcome to the dangerous Swiss golf-baseball hybrid you've never heard of
Described as a hybrid of baseball and golf, Hornussen sees two teams of 18 take turns hitting and fielding the "Nouss" or "Hornuss," a puck named after hornets for its buzzing sound as it whistles through the air." Armed with a 3-meter (9.84-foot) carbon stick called a "Träf," hitters take to a raised batting ramp in front of a playing area – the "Ries" – some 300 meters (980 feet) long and 10 meters (32 feet) wide. Their task is to strike the puck from the sloped platform.


Atlas
Trisha's Org ...
Access Manager
Billing
All Clusters
Get Help
Trisha
Project 0
Data Services
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
SERVICES
Triggers
Data API
Data Federation
SECURITY
Database Access
Network Access
Advanced
Goto
ccaat
items
tododb
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
_id: ObjectId('63c0ece2ab766b2897dd5679')
name: "Project Implementation"
desc: "Designing the model"
date: "2023-01-22"
pr: "High !!!"
done: "no"
_id: ObjectId('63c0f253ab766b2897dd567a')
name: "Technical seminar"
desc: "Implementation"
date: "2023-01-15"
System Status: All Good
©2023 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

8. Conclusion

The delivery of computing services via the cloud has grown in popularity, and it has several benefits including lower costs, scalability, flexibility, and reliability.

Depending on an institution's demands and goals, there are many cloud service types, deployment methodologies, and service model options. Before choosing a cloud provider, it is essential to carefully evaluate and compare possibilities because cloud providers offer a variety of services and have different capacities, price structures, and service representations. Many businesses are concerned about cloud security, so it's critical to thoroughly consider the safeguards and authorities in place before employing cloud services. The use of cloud computing is expected to continue to grow and mature in the years to come, so it will be important for businesses to stay informed about developments and the most cutting-edge methods in the field.

9. References

[1] "Cloud computing" available at https://en.wikipedia.org/wiki/Cloud_computing, accessed on 09-01-2023

[2] https://docs.aws.amazon.com/codedeploy/latest/userguide/tutorials-windows.html

[3] https://learn.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-deplo y-app

[4] " Kubernetes vs Docker: Why not both?" available at https://www.ibm.com/cloud/blog/kubernetes-vs-docker, accessed on 12-01-2023

[5] "Docker Architecture" available at https://www.javatpoint.com/docker-architecture, accessed on 12-01-2023

[6] "Kubernetes concepts and Architecture" available at https://platform9.com/blog/kubernetes-enterprise-chapter-2-kubernetes-architecture-concepts/, accessed on 12-01-2023

[7] "Web Hosting - Amazon Web Services" available at https://aws.amazon.com/websites/, accessed on 13-01-2023

[8] "Azure Cloud services" available at https://azure.microsoft.com/en-in/products/cloud-services, accessed on 13-01-2023

[9] "Top 10 Advantages of Choosing Google Cloud Hosting (2023)" available at https://kinsta.com/blog/google-cloud-hosting/, accessed on 13-01-2023