Department of Computer Science and Engineering

VII Sem CS19741 Cloud Computing

Study Group Creator System for students

Team ID: 6
HARINI A 220701084
RAKHUL PRAKASH S B 220701216
SANJAY S 220701248
SHANTHOSH S 220701263

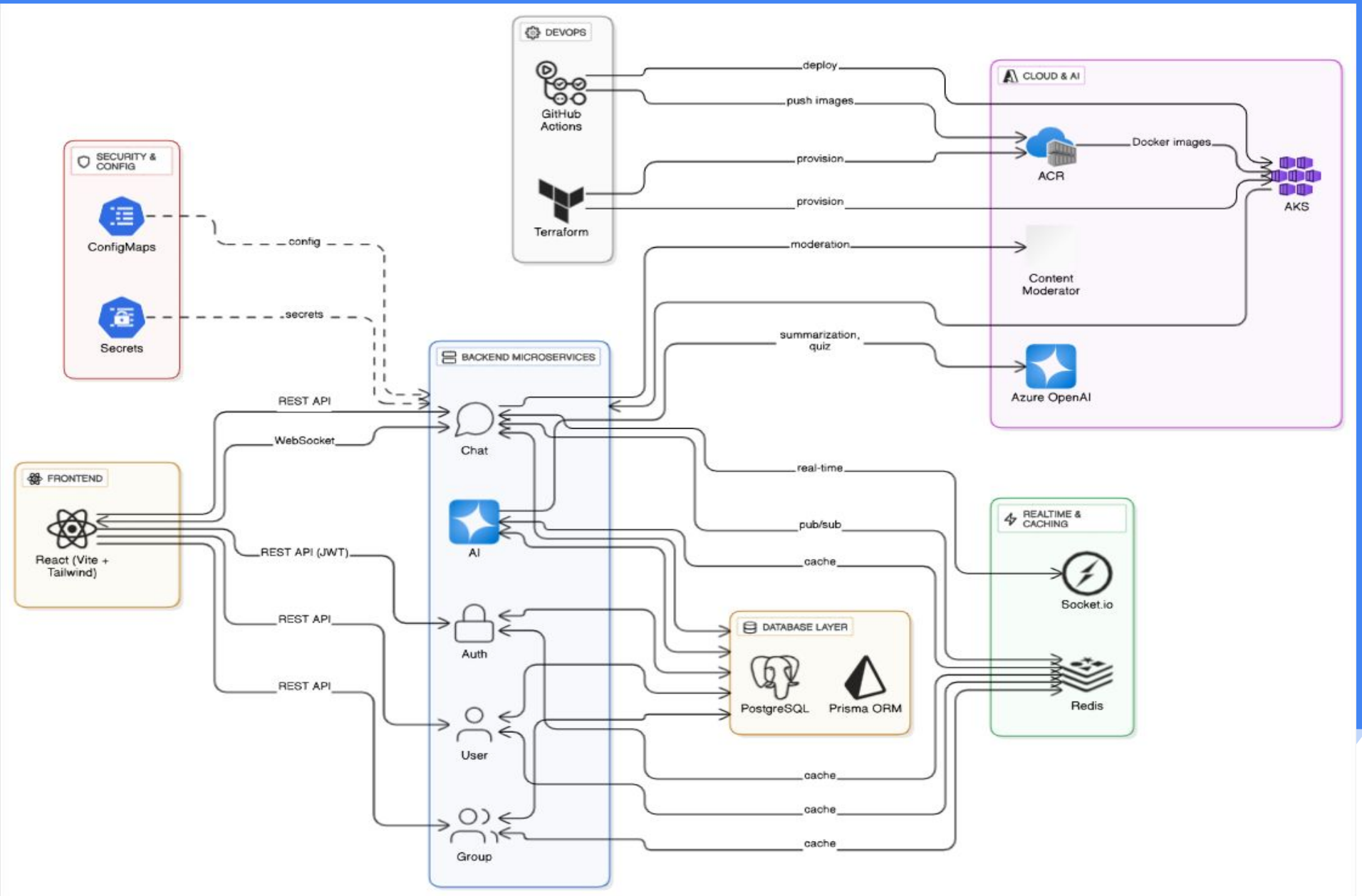
Problem Statement:

Students often struggle to **collaborate effectively, share resources, and manage study activities** online. Existing platforms lack a unified solution for **study group formation, real-time communication, event scheduling, and AI-powered assistance**. This leads to difficulty in finding peers, sharing learning materials efficiently, and organizing study schedules.

Objectives:

- Enable **collaborative learning** through study groups.
- Provide **real-time chat** with moderation for safe communication.
- Simplify **resource sharing and AI-powered summarization** of study materials.
- Manage **events, schedules, and online sessions** efficiently.
- Ensure **scalable and reliable architecture** using Docker & Kubernetes.
- Enhance the **learning experience** with AI-driven insights and quiz classification.

SYSTEM ARCHITECTURE DIAGRAM



Azure Services Integration Azure Kubernetes Service

StudySphere leverages Azure cloud services to ensure a robust, scalable, and intelligent platform for collaborative learning. These services provide container orchestration, AI capabilities, secure storage, and content moderation.

- **Azure Kubernetes Service (AKS)**
  AKS manages the deployment and orchestration of all microservices. Each service runs in its own container, and AKS handles scaling, self-healing, and load balancing, ensuring high availability and performance.

- **Azure Container Registry (ACR)**
  Docker images for each microservice are stored in ACR. This enables seamless deployment and version control of images directly into the Kubernetes cluster.

- **Azure OpenAI Service**
  Powers AI-driven features:

  - Quiz Classification: Categorizes user quiz responses to capture interests.

  - Resource Summarization: Generates concise bullet-point summaries of uploaded study materials for easy learning.

- **Azure Content Moderator**
  Ensures safe communication by monitoring chat messages and shared resources, flagging harmful or inappropriate content in real-time.

DevOps Automation & Infrastructure Terraform

StudySphere employs modern DevOps practices to automate infrastructure setup, deployment, and continuous integration, ensuring reliable and reproducible environments.

- Infrastructure as Code (Terraform)
  Terraform provisions cloud resources such as the Azure Kubernetes Service (AKS) cluster, networking components, and storage. This approach allows consistent and automated infrastructure setup, reducing manual errors.

- CI/CD Automation (GitHub Actions)
  GitHub Actions orchestrates the build, test, and deployment pipeline for all microservices:

  - Build: Docker images are created for each microservice.
  - Test: Unit and integration tests ensure code quality.
  - Deploy: Images are pushed to Azure Container Registry (ACR) and deployed to AKS.

- Secrets & Configuration Management
  Environment variables, API keys, and sensitive information are securely handled within the pipeline using GitHub secrets and Kubernetes ConfigMaps/Secrets. This ensures secure and seamless deployment across environments.

CI/CD WORKFLOW

The CI/CD workflow in StudySphere ensures continuous integration, automated testing, and seamless deployment of all microservices, providing a reliable and reproducible development process.

- **Code Commit & Trigger**
  Every code push or pull request triggers the CI/CD pipeline in GitHub Actions.
- **Build Stage**
  - **Docker images are built for each microservice using their respective Dockerfiles.**
  - **Dependencies are installed and the code is packaged for deployment.**
- **Test Stage**
  - **Unit and integration tests are executed to ensure code quality and service reliability.**
- **Deploy Stage**
  - **Docker images are pushed to Azure Container Registry (ACR).**
  - **Images are then deployed to Azure Kubernetes Service (AKS).**
- **Approval & Security**
  - **Approval gates ensure manual review when necessary.**
  - **Secrets and environment variables are securely managed using GitHub secrets and Kubernetes ConfigMaps/Secrets.**

CONTAINERIZATION

StudySphere uses Docker to package each microservice into isolated, portable containers, ensuring consistency across development, testing, and production environments.

- Dockerfile Setup
  - Each microservice uses `node:18-alpine` as the base image.
  - Dependencies are installed, and the source code is copied into the container.
  - Required ports are exposed, and the service start command is configured.
- Image Creation & Storage
  - Docker images are built for every microservice.
  - Images are pushed to Azure Container Registry (ACR) for version control and deployment.
- Deployment Readiness
  - Containers are ready to be deployed to Azure Kubernetes Service (AKS), enabling scalable and reliable orchestration.

Docker Implementation in StudySphere

Docker plays a crucial role in containerizing StudySphere's microservices, ensuring a consistent runtime environment and smooth deployment across all stages — from development to production.

- Purpose of Dockerization
  - Eliminates the "it works on my machine" problem by standardizing environments.
  - Simplifies deployment and scaling by packaging applications with their dependencies.
- Implementation in StudySphere
  - Each microservice (Auth, Group, Chat, AI, User) has its own Dockerfile based on the lightweight `node:18-alpine` image.
  - The Dockerfile defines steps to install dependencies, copy the source code, expose required ports, and specify the startup command.
  - Built images are tested locally before being pushed to the Azure Container Registry (ACR).
- Integration with CI/CD
  - Docker image creation is automated within the GitHub Actions pipeline.
  - After testing, images are deployed to Azure Kubernetes Service (AKS) for orchestration.

Through Docker, StudySphere achieves portability, scalability, and streamlined deployments, making the system easier to manage, replicate, and scale across cloud environments.

Kubernetes Orchestration

StudySphere leverages Kubernetes (AKS) to orchestrate its containerized microservices, ensuring scalability, resilience, and efficient resource management.

- Microservice Deployment
  - Each microservice (Auth, User, Group, Chat, AI) is deployed as a separate pod.
  - Kubernetes manages replicas, service discovery, and load balancing for consistent performance.
- Configuration & Secrets Management
  - ConfigMaps store non-sensitive configuration data.
  - Secrets securely store API keys, database credentials, and environment variables.
- Scaling & Resilience
  - Horizontal Pod Autoscaling (HPA) adjusts the number of pods dynamically based on load.
  - Failed pods are automatically restarted, ensuring high availability and fault tolerance.

By using Kubernetes, StudySphere achieves a robust, self-healing, and scalable system, capable of handling real-time collaboration for multiple study groups without downtime.

GenAI Integration & Cloud Mapping AI features

StudySphere integrates AI capabilities to enhance learning, resource management, and personalized study experiences, while ensuring safe and moderated communication.

- AI-Powered Features
  - Quiz Classification: Categorizes user quiz responses to capture academic interests and personalize study recommendations.
  - Resource Summarization: Generates concise summaries of uploaded study materials for quick comprehension.
- Backend Integration
  - Backend services communicate with the Azure OpenAI API to process AI requests asynchronously.
  - AI responses are returned to the frontend for display in study groups, dashboards, and resource modules.
- Content Moderation
  - Azure Content Moderator scans chat messages and shared resources to flag inappropriate content, ensuring a safe learning environment.
- System Flow Mapping
  - AI requests flow from user interactions → backend microservices → OpenAI API → processed results returned → frontend.
  - Moderation checks occur in parallel to maintain safe collaboration.