| Estimating Cloud Resources |
| --- |

Date: June 1st 2024

Revision: 1.0

| **SOW Reference** | TEAM # 1 /2024-06/001 | **Professor** | **Yang Sun**  [**Yangsun.com@gmail.com**](mailto:Yangsun.com@gmail.com) |
| --- | --- | --- | --- |
| **Project Title** | **TEAM # 1**  **Part # 3**  **Estimating Cloud Resources** |
| **Team Members Email ID** | **Yilun Jin**  **Maheswar Barrenkala**  **Akshat Jangam**  **Atilla Karaali**  **Shital Hinge**  **Jean Qiong Zhang**  **Caihui Yang**  **Dhruv Basnotra**  **Xu Liu** | **Team Members Email ID** | [**karen.jin.job@gmail.com**](mailto:karen.jin.job@gmail.com)  [**maheswar.barrenkala@gmail.com**](mailto:maheswar.barrenkala@gmail.com)  [**akshat.ja@gmail.com**](mailto:akshat.ja@gmail.com)  [**atilla.karaali@gmail.com**](mailto:atilla.karaali@gmail.com)  [**Shitalkarleg@gmail.com**](mailto:Shitalkarleg@gmail.com)  [**jzhang95014@gmail.com**](mailto:jzhang95014@gmail.com)  [**ashleyyang924@gmail.com**](mailto:ashleyyang924@gmail.com)  [**dhruvbasnotra@gmail.com**](mailto:dhruvbasnotra@gmail.com)  **liuxubupt16@gmail.com** |

| **Part 3: Estimating Cloud Resources** This web application is mainly designed for registering courses. Assuming 5000 Daily Active Users (DAU), the peak time usually takes place after the registration opens, we may assume that there are 4000 concurrent visits, and each request will consume 500ms CPU time and 50MB RAM. After calculation, the 4000 concurrent visits will consume about 200GB RAM, so we may need to choose instances with more memory.We choose the AWS EC2 r5.large to handle our microservices. The cloud resources are estimated as the following: **System Design Components****Compute Resources**   **1.** **Web Servers:**  · Instances: 3 instances  · Type: AWS EC2 r5.large  · Role: Handle frontend and API gateway    **Reasoning:**  · DAU and Concurrency: With 5000 daily active users and a peak of 4000 concurrent visits, distributing the load across 3 instances helps manage high traffic and ensures availability.  · Instance Type: AWS EC2 r5.large instances are chosen due to their high memory (16GB) and sufficient CPU power (2 vCPUs), which are suitable for handling the frontend and API gateway processing needs.  · Performance: Each instance can efficiently handle multiple concurrent connections, ensuring the frontend remains responsive under peak load conditions.    **Cost Calculation:**  · On-Demand Price: $0.126 per Hour (varies by region)  · Monthly Cost: 3 instances \* $0.126 \* 24 hours \* 30 days ≈ $272.16    **2.** **Microservices:**  · Instances: 13 instances  · Type: AWS EC2 r5.large  · Role: Run various microservices    **Reasoning:**  · Workload Distribution: Given the estimation that each request consumes 500ms of CPU time and 50MB RAM, spreading the microservices across 13 instances ensures that the system can handle up to 4000 concurrent requests without performance degradation.  · Instance Type: Similar to the web servers, r5.large instances provide the necessary memory and CPU resources to handle the processing and communication needs of various microservices.  · Scalability: Having multiple instances allows for better scaling and fault tolerance, ensuring that the microservices can handle variable loads and remain available.    **Cost Calculation:**  · On-Demand Price: $0.126 per Hour (varies by region)  · Monthly Cost: 13 instances \* $0.126 \* 24 hours \* 30 days ≈ $1,179.12 **Storage Resources**   **1.** **Auth Service Database (RDS):**  · Storage: 25 GB  · Type: Amazon RDS (PostgreSQL or MySQL)  · Role: Store student data including studentId, email, courses taken, and profile data    **Reasoning:**  · Data Size Estimation: Auth service primarily stores user information, which includes student IDs, emails, courses taken, and other profile data. 25 GB is a conservative estimate to ensure adequate space for user data growth over time.    **Cost Calculation:**  · Storage Cost: $0.115 per GB/month  · Monthly Cost: 25 GB \* $0.115 ≈ $2.88    **2.** **Upload Course Service Database (MongoDB):**  · Storage: 10 GB  · Type: MongoDB Atlas  · Role: Store metadata related to courses such as professor, courseId, course code, and course name    **Reasoning:**  · Metadata Storage: Course metadata (professor info, course ID, course code, course name) typically requires less storage compared to actual course content. 10 GB provides ample space for storing this metadata.    **Cost Calculation:**  · Storage Cost: $0.23 per GB/month (varies by provider)  · Monthly Cost: 10 GB \* $0.23 ≈ $2.30    **3.**  **Course Content Storage (Amazon S3):**  · Storage: 500 GB  · Role: Store course syllabus, assignments, exams, and other large files    **Reasoning:**  · Large File Storage: Course syllabi, assignments, exams, and other related documents can be large. 500 GB ensures there is enough space to store these files, especially as new courses are added and more content is generated.    **Cost Calculation:**  · Storage Cost: $0.023 per GB/month  · Monthly Cost: 500 GB \* $0.023 ≈ $11.50    **4.** **Course Registration Service Database (RDS):**  · Storage: 50 GB  · Type: Amazon RDS (PostgreSQL or MySQL)  · Role: Update student profiles and store courses registered by students    **Reasoning:**  · Data Volume: This database will store detailed registration records, including the courses registered by each student. 50 GB ensures sufficient storage for a large number of records and accommodates data growth.    **Cost Calculation:**  · Storage Cost: $0.115 per GB/month  · Monthly Cost: 50 GB \* $0.115 ≈ $5.75    **5.** **Course Registration Service Database (Cassandra):**  · Storage: 50 GB  · Type: Amazon Keyspaces (Cassandra)  · Role: Store all students registered in courses    **Reasoning:**  · Scalability and Performance: Using Cassandra (via Amazon Keyspaces) for storing students registered in courses provides high availability and scalability. 50 GB is a reasonable estimate to handle large datasets efficiently.    **Cost Calculation:**  · Storage Cost: $0.30 per GB/month  · Monthly Cost: 50 GB \* $0.30 ≈ $15.00   **Cache**   **ElastiCache**  · Instances: 1 instance  · Type: AWS ElastiCache (cache.r5.large)  · Role: Cache frequently accessed data like course details to improve performance    **Reasoning:**  · Frequent Access Data: Caching frequently accessed data such as course details improves performance, especially during peak registration times. An r5.large instance with 16GB RAM ensures that the cache can hold a significant amount of data.    **Cost Calculation:**  · On-Demand Price: $0.192 per Hour  · Monthly Cost: 1 instance \* $0.192 \* 24 hours \* 30 days ≈ $138.24 **Network Resources**   **Load Balancers:**  · Instances: 2  · Type: AWS Network Load Balancers  · Role: Distribute traffic across servers    **Reasoning:**  · Traffic Distribution: To handle high traffic and ensure high availability, using two load balancers distributes incoming traffic evenly across multiple web servers and microservices, reducing the risk of overloading any single instance.    **Cost Calculation:**  · Monthly Cost per LB: $0.0225 per Hour + $0.006 per LCU/hour (assuming 10 LCU)  · Monthly Cost per LB: $0.0225 \* 24 \* 30 + 10 \* $0.006 \* 24 \* 30 ≈ $32.40  · Total Monthly Cost for 2 LBs: 2 \* $32.40 ≈ $64.80 **Additional Resources**   **Message Queue:**  · Type: Apache Kafka on AWS MSK  · Role: Facilitate communication among microservices    Reasoning:  · Service Communication: A message queue like Kafka facilitates efficient and reliable communication among microservices, allowing them to exchange messages asynchronously and handle high throughput.    **Cost Calculation:** *(Assume basic cost)*  · Monthly Cost: ≈ $300 (varies significantly by usage and configuration)    **Notification Service:**  · Type: AWS SNS  · Role: Send notifications to students and staff    Reasoning:  · User Notifications: Using a notification service like AWS SNS ensures that students and staff receive timely updates and notifications regarding course registrations and other important events.    **Cost Calculation:** *(Assume basic cost)*  · Monthly Cost: ≈ $10 (varies by usage) **Total Monthly Cost Estimation**  | **Resource** | **Usage** | **Cost per Unit** | **Total Monthly Cost** | | --- | --- | --- | --- | | Web Servers (EC2 r5.large) | 3 instances | $0.126 per Hour | $272.16 | | Microservices (EC2 r5.large) | 13 instances | $0.126 per Hour | $1,179.12 | | Auth Service Database (RDS) | 25 GB | $0.115 per GB/month | $2.88 | | Upload Course Service Database (MongoDB Atlas) | 10 GB | $0.23 per GB/month | $2.30 | | Course Content Storage (Amazon S3) | 500 GB | $0.023 per GB/month | $11.50 | | Course Registration Service Database (RDS) | 50 GB | $0.115 per GB/month | $5.75 | | Course Registration Service Database (Cassandra) | 50 GB | $0.30 per GB/month | $15.00 | | Cache (ElastiCache r5.large) | 1 instance | $0.192 per Hour | $138.24 | | Load Balancers (Network Load Balancers) | 2 instances | 0.0225 per Hour + 0.006 per LCU/hour | $64.80 | | Message Queue (Kafka on AWS MSK) | Basic usage | ≈ $300 per month | $300.00 | | Notification Service (AWS SNS) | Basic usage | ≈ $10 per month | $10.00 |    **Grand Total Monthly Cost** Total:  $1,451.28 + $37.43 + $138.24 + $64.80 + $310 = $2,001.75   **Accuracy of Estimations** Usage Patterns and Load:  · The estimations are based on a realistic understanding of daily active users (5000 DAU) and peak concurrency (4000 concurrent users), which are typical metrics for high-traffic applications.  · The choice of instance types (r5.large) and the number of instances are calculated to ensure the system can handle the expected load without performance degradation.    Resource Allocation:  · Memory and CPU estimations are based on the assumed resource consumption per request (500ms CPU time and 50MB RAM). This allows for accurate sizing of instances to ensure efficient processing and responsiveness.  · Storage requirements are conservatively estimated to accommodate current needs and future growth, ensuring the system remains scalable.    Scalability and Redundancy:  · Using multiple instances for web servers and microservices ensures the system can scale horizontally to meet increasing demand and provides redundancy to improve availability and fault tolerance.  · Load balancers distribute traffic effectively, preventing any single instance from becoming a bottleneck.    Performance Optimization:  · Caching frequently accessed data with ElastiCache reduces database load and improves response times, especially during peak usage periods.  · Using a combination of RDS, MongoDB, Cassandra, and S3 ensures that different types of data are stored optimally, leveraging the strengths of each storage solution.    By following these estimations and resource allocations, the system is designed to handle high concurrency and provide a reliable, scalable, and efficient course registration web application.        END |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |