Date:08/03/2023
Day:01

1. Use the reference of YouTube[YT] :

S1 : A new user account is created in Gmail/YT
    Publisher → Create_account service ,
    Subscriber
        - Create_new_account service
        - ID_Auth_service
        - New_Account_ login service
        - New_ID_login service
        - Database
S2 : The user logs in to YOUTUBE using the account
    Publisher →New_user_account service ,
    Subscriber
        - ID_Auth_service
        - New_ID_login service
        - Location services
        - Timeline services
        - List services/network service
        - Notification services
        - Tag/Hash service
        - Advertisement service
        - Database
        - Logging
S3 : Create a subscription service in YT
    Publisher →Search_channel service,
    Subscriber
        - Subscription service
        - Related_content service
        - Database
        - Notification services
S4 : User :: Create a channel
    Publisher →New_channel service,
    Subscriber
        - Location services
        - Timeline services
        - Database
S5 :: user --> channel --> uploads a video
    Publisher →Upload_video service,
    Subscriber
        - Tag/Hash service
        - Advertisement service
        - Database
        - Recomended_video service

Date:08/06/2023
Day:02

6.What is an endpoint [NOT API endpoints --> Queue endpoint + topic endpoint]

An endpoint refers to a unique address or location within the messaging infrastructure where messages can be sent, received, or consumed.

For Queue Endpoints:
In the case of message queues, a queue endpoint represents a destination where messages are placed by producers and retrieved by consumers. Each queue has a distinct endpoint associated with it. Consumers can connect to this endpoint to receive and process messages in the order they were added to the queue.

For Topic Endpoints:
In publish-subscribe systems, a topic endpoint is a virtual channel or category to which messages can be published by producers. Consumers (subscribers) interested in specific topics can connect to the corresponding topic endpoint to receive messages related to that topic. Unlike queues, where each message is consumed by a single consumer, topics allow multiple consumers to subscribe to the same topic and receive copies of the published messages.

7.What is High Availability Architecture?

High Availability (HA) architecture is an approach to designing and implementing systems that ensure a high level of operational continuity, even in the face of hardware failures, software glitches, network issues, or other types of disruptions. The primary goal of high availability architecture is to minimize downtime and provide reliable access to services and applications, thus ensuring a seamless user experience.

what happens in High Availability Architecture

- Rapid Detection and Response: High availability systems continuously monitor the health and performance of components. When an issue is detected, the system responds quickly by initiating failover procedures or other corrective actions.

- Seamless Failover: In the event of a failure, the system automatically switches to a redundant or standby component. This can involve redirecting traffic, rerouting requests, and maintaining service continuity.

- Load Distribution: Load balancers distribute incoming requests across multiple resources to prevent any single component from becoming overwhelmed. This ensures that workloads are evenly distributed and that no component is a single point of failure.

- Data Integrity: Data replication and backup strategies ensure that data remains consistent and available. In case of a failure, data can be retrieved from replicas or backups to minimize data loss.

- Continuous Availability: High availability systems aim to provide continuous access to services and applications, even during maintenance, upgrades, or unexpected failures.