1. Patterns
   1. User Feed/Timeline (real-time)
      1. Redis
   2. User Feed real-time update
      1. Pub-Sub architecture
   3. Store Tweets
      1. Cassandra
   4. Messaging apps (Real time message)
      1. WebSockets
   5. Notification System (Async)
      1. Kafka, RabbitMQ
   6. Search systems (Read)
      1. ElasticSearch
2. Problems & Solutions
   1. Parallel process related data across multiple servers
      1. MapReduce algorithm
3. Metrics
   1. Availability
      1. 99.9% or more
   2. Load
      1. 100 Million DAU
4. Process
   1. High availability
      1. E.g. Social Media apps, Google search
      2. Replication
      3. Failover mechanism
   2. Eventual Consistency
      1. E.g. Likes on post
      2. Cassandra, DynamoDB (Highly districted systems)
      3. Achieved through Kafka??!
   3. Strong consistency
      1. E.g. Banking apps
      2. ACID compliant database (SQL)
   4. Low Latency
      1. E.g .Real time systems (chat apps, online gaming)
      2. WebSocket, UDP, Edge Caching (like CDNs for static content)
   5. Asychronous / Event Driven systems
      1. E.g. : Notification system
      2. Message Queue
5. Learnings
   1. Error codes :
      1. 401 : Unauthorized
   2. Authentications
      1. Token based : JWT, OAuth2