**What are Micro services**

1. An architectural style in which large complex software applications are composed of one or more services
2. Each of these micro services focuses on completing one task only and does that task really well
3. Communicate with each other using language-neutral APIs e.g. REST
4. Have a bounded context i.e. one micro service does not need to know anything about another micro service
5. Individual micro services can be developed using different technologies that best suit to their need
6. Can be deployed and scaled independent of each other
7. Better isolation for error detection and damage control

**Problems with micro services:**

1. Complexity
2. Security
3. Network latency
4. Connectivity issue
5. Debugging is harder while dealing with loosely coupled services

**Role of containers:**

* The best foundation for running a micro services application is application containers
* Containers encapsulate a lightweight runtime environment for the application, presenting an isolated, consistent software environment
* Containers themselves do not make sense without an orchestrator service such as Kontena, Docker Swarm or Kubernetes that delivers the desired application functionality
* Docker is used to create images and containers

**API Gateway**

**Problems that comes into picture if a client directly calls a micro service:**

* Number of calls made to different micro services for a page to render with all information can be a limiting factor
* So many requests over a public network is not at all a prudent way. This would also make the client much more complex
* For clients directly calling micro services, protocols might not be web friendly. Applications should use only HTTP/web sockets outside the firewall
* This approach is also problematic for refactoring micro services

**Need for an API gateway:**

1. Services may be updated and the address of the service could change
2. Dealing with different kinds of clients that have different kinds of requirements
3. Systems might have centralized authentication and security

**An API gateway provides a single, unified entry-point across one or more internal APIs:**

1. Enables support for mixing communication protocols
2. Decreases micro services complexity by providing authorization using API tokens
3. Provides request routing. All requests from clients at first must go thru API gateway. Then those are routed to appropriate micro services

**API Gateway**