Microservices Basics Tutorial Part 1

- What are Monolith Application?
  In Monolith Architecture the application is build as a single unit.
  Such Applications comprises of client side interface, server-side application and a database.
  Normally a monolith application have one large code base and it lack modularity

- Disadvantages of Monolith Application
  The code base get larger in size with time and hence it's very difficult to manage.
  It is very difficult to introduce new technology as it affects the whole application.
  A single bug in any module can bring down the whole application
  It is very difficult to scale a single module. One has to scale the whole application
  Continuous deployment is extremely difficult. Large monolithic applications are actually an obstacle to frequent deployments. In order to update one component, we have to redeploy the entire application.

- What are Microservices?
  While Monolith Application works a single component, a Microservice Architecture breaks it down to independent standalone small applications, each serving one particular requirement. Eg, 1 Micro service for Handling all vaccination center

operations , 1 for handling all the user base
Within this microservice architecture, the entire functionality is split in independent deployable module which communicate with each other through API's(RESTful web services)

- Advantages of Microservices
  All the services are independent of each other. Therefore testing and deployment is easy as compare to Monolith application
  If there is bug in one microservice it has an impact only on a particular service and does not affect the entire application. Even if ur Vaccination center service is down, u still have Ur users register to your application
  With microservice architecture, it's easy to build complex applications.
  It will give flexibility to choose technologies and framework for each microservices independently

- How to start with Micro services?
  Steps
  If u have a monolith application, Identify all possible standalone functionalities.
  Once u have identify them, you need to create standalone projects, we are taking spring boot to create these microservices.
  You need them to interact with each other through some ways , It can be Rest Api or Messaging. We are going to use restful architecture for the same
  But just doing this does not make sure the you have implemented microservices architecture. These are till now just 2 Restful web services. You need load balancer , eureka for service discovery (useful during load balancing and cloud deployments), API gateways and many more stuff.