1. There is great saying that with great power comes great responsibility and this is true with microservice architecture.
2. This is because when enjoying the benefits from microservices, you have to face the challenges in building the micro-services.
3. We already discussed the 1st challenge in microservice about how to decide about the size of a microservice.
4. Let’s move on to the 2nd challenge.
5. Challenge: How do you do the following with so many microservices in your organization.
   1. Deployment,
   2. Portability,
   3. Scalability.
6. So, to know more about this challenge, let me ask you a few questions.
7. 1st Question:
   1. In monolithic app, we have only one WAR which can easily be deployed.
   2. **Time**: Now, you have 100 microservices.   
      If you go to deploy them, it may take days and days of time.   
      Still, there may be human error.
   3. **Infrastructure**: For monolithic app, you may take 64GB RAM, 1TB Hard-Disk on a server.   
      But for 100 microservices, you can’t follow this way to deploy them on 100 different servers.  
      As it would be maintain-nightmare and cost a lot to buy such huge resources on cloud providers 🡺 AWS, GCP, Azure etc.
   4. **Portability**: In monolithic app, only one war to be deployed so easy to move from one environment to another environment as you don’t have to do much configuration.  
      Take for example: DB Configuration, e-mail configuration like SMPT Configuration.  
      Such configuration modifications are difficult in microservices.
   5. **Scalability:** What you do with monolithic app to scale it is you increase RAM, Hard-Disk (Vertical Scaling),   
      or you create a cluster with a few servers using load balancer (Horizontal Scaling).  
      But in case of 100 microservices, this approach is not suitable. You don’t buy virtual machine on cloud as it costs a lot.  
      So what is the solution?  
      **Answer**: Containerization.  
      Docker Vendor makes our life very easy in building, deploying, scaling micro services very easily and with less cost.
8. Let’s see in next lecture, what is this containerization and how it is different from virtual machine.