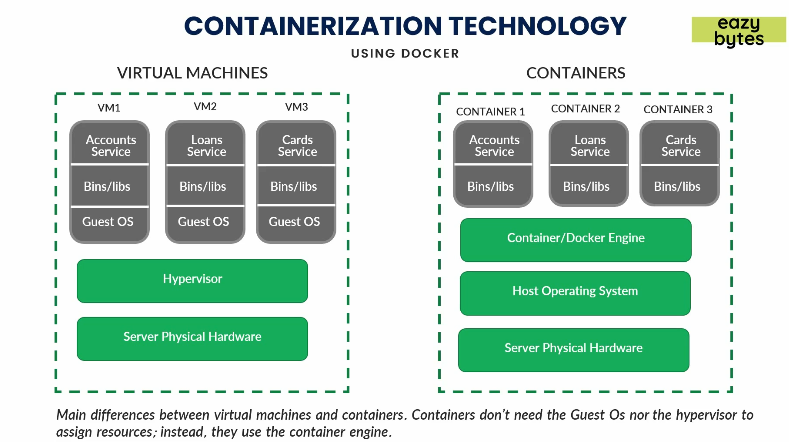
* 1. What is Containerization technology used heavily in Docker Eco-System.

1. Before discussing container, let’s talk about virtual machine.
2. Cloud providers such as AWS provide us a virtual machine which is a virtual server.

AWS calls it EC2 (Elastic Compute Cloud) Instance.

1. So, you buy virtual machine with capacity say 600GB RAM, 1TB Hard-Disk and on this virtual machine you can install whatever you like such as  
   Tomcat Server, JDK, DBMS etc and you deploy your app or DB there.
2. In traditional virtual machine,
   1. We have to take 3 servers VM1, VM2, VM3 etc from a Cloud Provider.
   2. Then we can deploy all three services 🡺 Account, Card, Loan services one on a separate VM.
   3. Problem with this Approach:
      1. You have to pay unnecessarily for 3 different servers for microservices which are not that heavy.
      2. When scaling up or down, it takes good amount of time like you have to onboard another virtual machine and you have to start it, install all dependencies only then you can deploy your service which is time consuming and so not suited for microservices.  
         In microservice world, we can kill and restart a microservice instance within a few seconds if it doesn’t perform well.  
         In VM environment, Hypervisor is installed on top of Host OS.   
         Hypervisor creates VM based on the user’s configuration where Guest OS is installed.  
         So, now after installing all the required dependencies, you can deploy your app.  
         If you want to redeploy your app, you need to restart Guest OS which may take good amount of time.  
         Now, if you have 100 microservices and each needs to be redeployed, it will take very long time.  
         So, VM is not suited for microservices.
3. **Solution**: Container
   1. Container doesn’t have Guest OS so stopping and starting a container is a task of a few seconds.