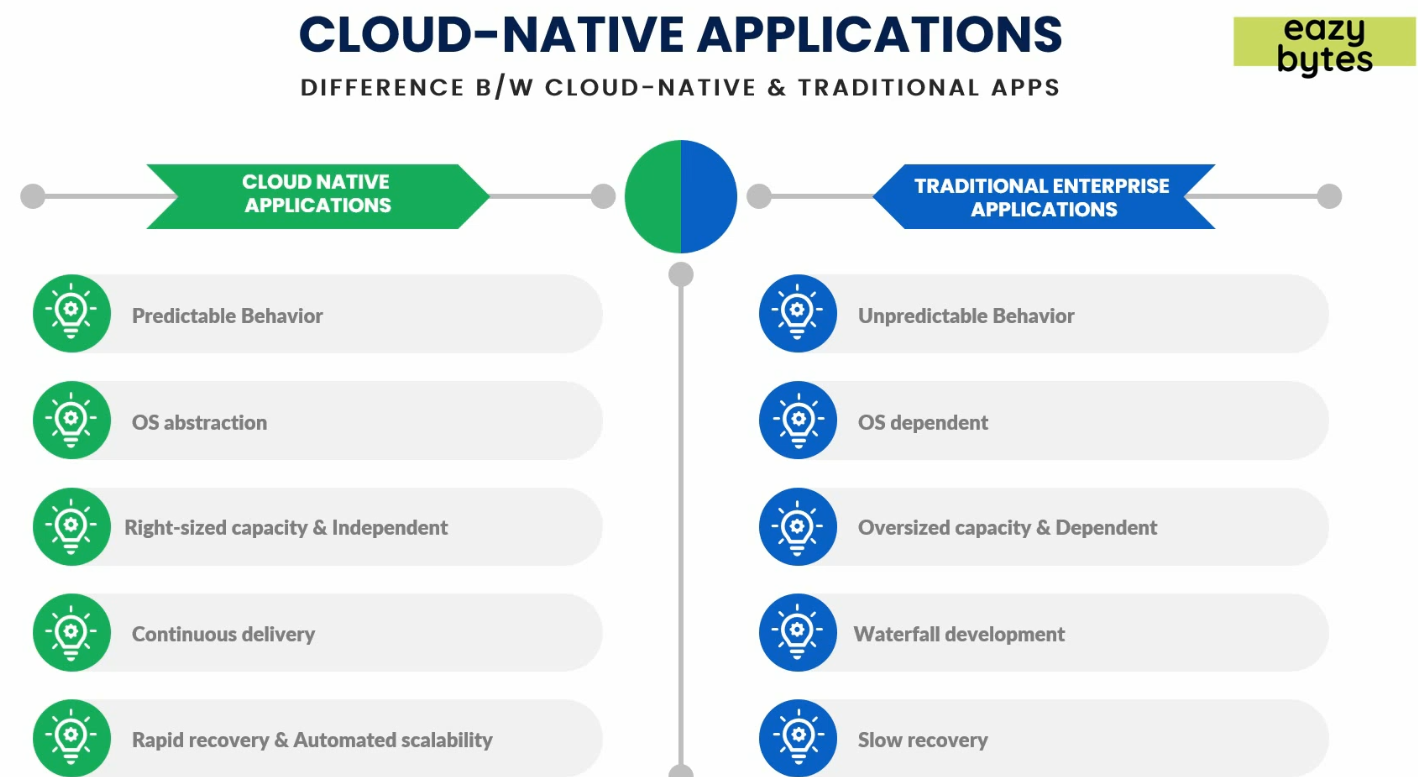
1. 
2. **Predictable Behaviour:** 
   1. If your one microservice instance is down due to any reason, with Kubernetes Orchestration and Docker, you can replace that within a matter of seconds and thus making sure that your app is not getting affected.
   2. If one of our microservices is not working, we have enough ways to have resilience/self-healing inside our app.
   3. But in case of monolithic app, if it goes down, all kinds of users get affected.
   4. So, if one of the microservices is not working, we have enough ways to have that resilience inside our app. Actually, self healing inside our apps, which is one of the advantages that we have with cloud native app.  
        
      **NOTE**: Don’t worry how to add this feature to our microservice, we will discuss in future lecture.
3. **OS Abstraction**:
   1. No layer of OS in case of Docker Image. Docker will take care of the underlying OS.
   2. Monolithic app is dependent on the OS in VM. Because Inside VM, an OS is required to be installed.
   3. Cloud native uses container which is light weight whereas traditional apps (monolithic) uses VM which is heavy weight as it requires Guest OS in VM.
4. **Right-Sized Capacity & Independent**:
   1. Microservices are small in size so easy to predict the size of infrastructure or for container.
   2. But in monolithic app, as the size is big, so you take bigger size infrastructure.
5. **Continuous Delivery**:
   1. You can roll in a new microservice or a new feature into already running microservice without affecting other microservices.
   2. Whereas in monolithic app, you have to stop the complete app along with DBs.
6. **Rapid Recovery & Automated Scalability:**
   1. If any microservice is not working, we have self-healing capacity using resilience framework.
   2. Automatically scaling up or down based on different parameters.
      1. When using more than x% CPU Capacity.
      2. When more load.
      3. On weekend.