Name : Jadeja Akshayrajsinh N.

**Microservices Assignment 1**

**What is microservices?**

The term microservices portrays a software development style that has grown from contemporary trends to set up practices that are meant to increase the speed and efficiency of developing and managing software solutions at scale. Microservices is more about applying a certain number of principles and architectural patterns as architecture. Each microservice lives independently, but on the other hand, also all rely on each other. All microservices in a project get deployed in production at their own pace, on premise on the cloud, independently, living side by side.

**Challenges with monolithic oriented architecture**

* This simple approach has a limitation in size and complexity.
* Continuous deployment is difficult.
* Monolithic applications can also be difficult to scale when different modules have conflicting resource requirements.
* Another problem with monolithic applications is reliability. Bug in any module (e.g. memory leak) can potentially bring down the entire process. Moreover, since all instances of the application are identical, that bug will impact the availability of the entire application.
* Monolithic applications has a barrier to adopting new technologies. Since changes in frameworks or languages will affect an entire application it is extremely expensive in both time and cost.

**Advantages of microservices**

* It enables each service to be developed independently by a team that is focused on that service.
* Microservice architecture enables each microservice to be deployed independently. As a result, it makes continuous deployment possible for complex applications.
* Microservice architecture enables each service to be scaled independently.

**Disadvantages of microservices**

* Microservices architecture adding a complexity to the project just by the fact that a microservices application is a distributed system. You need to choose and implement an inter-process communication mechanism based on either messaging or RPC and write code to handle partial failure and take into account other [fallacies of distributed computing](http://www.antonkharenko.com/2015/06/notes-on-fallacies-of-distributed.html).
* Microservices has the partitioned database architecture. Business transactions that update multiple business entities in a microservices-based application need to update multiple databases owned by different services. Using distributed transactions is usually not an option and you end up having to use an eventual consistency based approach, which is more challenging for developers.
* [Testing a microservices](http://martinfowler.com/articles/microservice-testing/) application is also much more complex then in case of monolithic web application. For a similar test for a service you would need to launch that service and any services that it depends upon (or at least configure stubs for those services).