Name: Meghna Mehta, 2021 Trainee, meghna.mehta@thegatewaycorp.co.in

MICROSERVICES ASSIGNMENT1

1) What is Microservice?

Ans) Microservices are an architectural approach/Service Oriented Architecture to building applications. As an architectural framework, microservices are distributed and loosely coupled, they have bounded contexts.

2) Challenges with monolithic oriented architecture.

Ans) The following are the challenges faced in the Monolithic Architecture:

1) Difficult to Scale.
2) Long Build/Test/Release Cycles.(Who's changes affected the application badly?)
3) If any separate operation is failing than it's a nightmare.
4) It takes long time to add new features.
5) Architecture is hard to maintain and evolve.
6) Lack of Innovation.
7) Lack of Agility.
8) New releases takes weeks/months.
9) Frustrated customers.

3) Any 3 advantages and disadvantages of Microservices.

Ans) The following table shows the advantages and disadvantages of Microservices:

Advantages	Disadvantages
1) Increased resilience With microservices, your entire application is decentralized and decoupled into services that act as separate entities. Unlike the monolithic architecture wherein a failure in the code affects more than one service or function, there is minimal impact of a failure using microservices.	1) With a microservices architecture, service discovery, networking, testing and monitoring all become more complex and difficult

2) Improved scalability Scalability is the key aspect of microservices. Because each service is a separate component, you can scale up a single function or service without having to scale the entire application. Business-critical services can be deployed on multiple servers for increased availability and performance without impacting the performance of other services.	2) As the number of services increases, the challenge using Microservices gets amplified.
3) The ability to use the right tool for the right task Each service can use its own language, framework, or ancillary services while still being able to communicate easily with the other services in your application.	3) Tools that once seemed essential, such as logging, are now racking up huge bills that can send a microservices migration into negative ROI territory.
4) Faster time to market Because microservices works with loosely coupled services, you don't need to rewrite your entire codebase to add or modify a feature. You make changes only to a specific service. By developing applications in smaller increments that are independently testable and deployable, you can get your application and services to market quicker.	4) **Extra complexity.** Since a microservices architecture is a distributed system, you have to choose and set up the connections between all the modules and databases. Also, as long as such an application includes independent services, all of them have to be deployed independently.
5)Easier debugging and maintenance: Microservices also makes it easy to debug and test applications. With	5) Debugging is difficult as the control flows over many microservices and to point out why and where exactly the error occurred is a difficult task.

smaller modules going through a continuous delivery and testing process, your ability to deliver error-free applications is vastly improved.	