Nowadays, we’re using software everywhere, which is application. To simplify it, we’ve different modules. Developers are working in different modules as different teams.

In monolithic, we’re developing software’s (using only one technology) that runs in one machine and deploy it in cloud or any server. Whereas in microservice architecture, we can use different technologies like Java, Python and Node JS. Different services can develop using different technologies, we can scale based on services. If one microservices is down, it’ll not crash the entire application, but in monolithic application will be down if any service is down. In microservices, each service will run independently, so if any of it fails, application will work, only that particular service will be down.

Real time example :   
  
Let’s consider Amazon  
  
In Amazon application, we’ve lots of services such as

--------------Payment, Searching, Catalogue, Seller, User, Shipping---------------

We’ve sales is going on in Amazon, at that time, the number of users who is going to use the application is almost same, searching and payment services will be used by users a lot. We’ve to improve the performance of only searching and payment. But if we’re using monolithic, even though different developers worked on different services, we’re going to create a single war at the end, all services in a single package, we’ve to deploy the entire application instead of particular services. And also team dependency is there, if I am improving the performance of one service, then also we’ve to wait and dependent on other teams for release. Team dependency (One team has to dependent on other team like when they’re going to have next release) and scalability (Vertical scaling by adding more servers or horizontal scaling – same application having multiple instances. Whereas in microservices, instead of scaling the entire application, we can scale particular services) are two disadvantages of using monolithic architecture. Some of the advantages of microservices architecture is, we can increase the number of servers for one particular services, we can use the required number of servers for different services and we can develop different services using different frameworks, For example, I can develop one service using Java, another using Python, and another using Node JS. We can deploy particular services which we improve the performance. Some of the issues are how can we connect these services, and we can connect through http Request. But how do we map the request to particular services. We can use API Gateway to map user request to particular services. Also one more problem is, user might not have access to particular services, these things also, we need to handle in microservices.  
  
(An [**API gateway**](https://www.nginx.com/resources/glossary/api-gateway/) accepts API requests from a client, processes them based on defined policies, directs them to the appropriate services, and combines the responses for a simplified user experience. Typically, it handles a request by invoking multiple microservices and aggregating the results)