**Microservices**

Microservices, also known as microservice architecture is an architectural style that structures an application as collection of smaller services.

* Highly maintainable and testable
* When the services are small, the code base is much easier to maintain and test.
* Loosely coupled
* The loosely coupled this is an important point services connect with each other on a service end point. So no two services know any implementation details about the other or have any direct references.
* Organized around business capabilities
* What that means is instead of splitting a large application into more minus services based on size. You decouple them based on the business problems. Often patterns like the domain-driven design are used to identify the business problem and separate them in their own domain logic, they’re owned by small team.
* Owned by a small team
* This is why people often talk about implementing Microservices need a cultural change in the organization to every Microservices is usually managed by independent teams that give the autonomy to decide when to release a feature or fix a bug with a weighting on the entire teams released.
* Independently deployable
* Having a service model based on domain business, domain, autonomous teams, managing them and loosely coupled their package and deployed independently.