Introduction and what is the problem that microservices trying to solve:

* The problem that microservices try to solve:

We will look at the main design principles that make up this style of architecture. We will look at exactly what these design principles are and what they imply when it comes to designing your software. For years, people have developed software using a style which is the opposite of microservices and this has unfortunately resulted in large, rigid applications and services, and over time, these large, rigid applications and services and APIs become so complex that they are very difficult to change and deploy without introducing risk. The phrase **monolith architecture** is sometimes used to describe this large and unwieldy software architecture style and microservices architecture is the solution to avoid a monolith-type style.

* Microservices benefits:

[image]

Microservices architecture and its design principles enable us to develop software that is easy and risk-free to change and deploy, and it lets us make softer architecture that is made up of small components that are each individually scalable, allowing us to scan out specific components that require improved performance. This also reduces risk during deployment, because we only have to deploy components that have changed instead of deploying the entire software architecture each time. There are many other design benefits that microservices architecture style brings to software.