

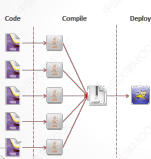
什么是微服务?

- ① 微服务是一种架构风格, 将应用程序组织为小的自治服务集合, 围绕业务领域建模。
- ② 在微服务架构中, 每个服务都是独立的, 实现单一业务功能。
- ③ 微服务架构, 简称微服务, 是专注于构建具有良好定义的接口和操作的单一功能模块。

单体架构



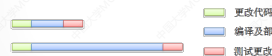
单体架构



编译时, 这些模块被打包到jar包中并最终合并在一起形成war包。接下来, 需要将war包上传到Web容器, 解压war包, 然后重新启动服务器。

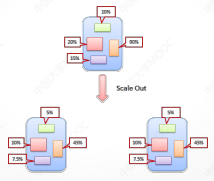
单体架构

如果应用程序部署繁琐, 软件开发人员还需要在部署之前进行大量环境设置, 以便测试其更改, 使软件开发人员的工作变得复杂和枯燥

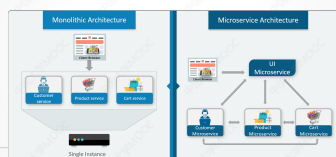


如图所示, 在应用程序变大之后, 软件开发人员花费了更多的时间进行编译和部署, 超过了其对代码进行更改和测试的时间, 效率也变得非常低。

单体架构



单体架构与微服务之间的差异



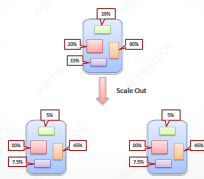
微服务

微服务是一种架构风格，其中大型复杂软件应用程序由一个或多个服务组成。微服务可以彼此独立地部署并且耦合松散。每个微服务都专注于完成一项任务。

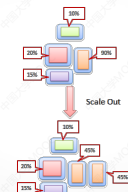
对比

Category	Monolithic architecture	Microservices architecture
Code	A single code base for the entire application.	Multiple code bases. Each microservice has its own code base.
Understandability	Often confusing and hard to maintain.	Much better readability and much easier to maintain.
Deployment	Complex deployments with maintenance windows and scheduled downtimes.	Simple deployment as each microservice can be deployed individually, with minimal if not zero downtime.
Language	Typically entirely developed in one programming language.	Each microservice can be developed in a different programming language.
Scaling	Requires you to scale the entire application even though bottlenecks are localized.	Enables you to scale bottle-necked services without scaling the entire application.

单体架构



微服务



微服务的好处

Simpler To Deploy	Deploy in literal pieces without affecting other services.
Simpler To Understand	Follow code easier since the function is isolated and less dependent.
Reusability Across Business	Share small services like payment or login systems across the business.
Faster Defect Isolation	When a test fails or service goes down, isolate it quickly with microservices.
Minimized Risk Of Change	Avoid locking in technologies or languages change on the fly without risk.

微服务架构

