DataBase Per Service

*In this pattern each microservice will have its own private database enabling loose coupling of services.*

*Database per service pattern is easy to use and is ideal for new applications. Also, the microservices pattern offers high scalability, loose coupling between databases, and simple impact analysis.*

***Benefits of database per microservice pattern -:***

* *Isolation and Independence*
* *Autonomous Development and Deployment*
* *Technology Flexibility - Different microservices can use database technologies that best suit their specific needs.*
* *Scalability - each microservice can scale its database independently based on its unique requirements.*
* *Reduced Coupling - By avoiding direct database access between microservices, you reduce coupling between them. Microservices interact with each other through well-defined APIs, which allows for loose coupling and greater flexibility in making changes to services.*

***Cons –***

*However, it also comes with challenges, such as ensuring data consistency and implementing effective communication patterns between microservices, which require careful consideration and implementation.*

*bottleneck due to application level microservices – if we divide monolith into microservices only at microservices*