



Spring Framework 6

Beginner to Guru

Introduction to Microservices



What Are Microservices?

- Microservices are small targeted services
- Each service has its own repository
- Microservices are isolated from other services
 - Should not be bundled with other services when deployed
- Microservices are loosely coupled
 - When interacting with other services, should be done in a technology agnostic manner
 - ie - Restful web services - HTTP / JSON
 - Messaging w/JSON payloads





Microservice Architecture

- With a Microservice Architecture:
 - Applications are composed using individual microservices
 - Each service will typically have its own database
 - Each microservice is independently deployable
 - Scaling of individual services is now possible
 - CI/CD becomes easier since services are smaller and less complex to deploy





Benefits of Microservices

- Easy to understand & develop - Services are smaller and more targeted
- Software Quality - Since services are more targeted and have a limited scope
- Scalability - Independent services can be scaled up and down to the application's demands.
- Reliability - Software bugs are isolated
- Technology flexibility - Services can be developed using any language or technology stack.





Cons of Microservices

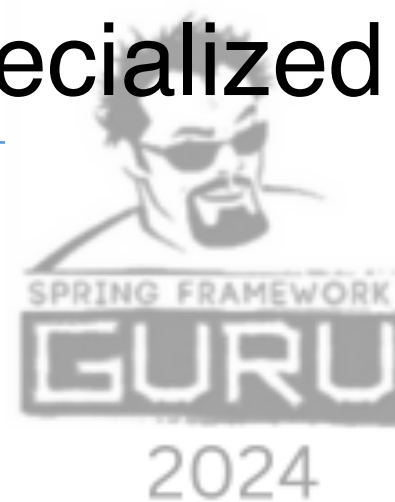
- Integration testing can be difficult
- Deployments are more complex. Rather than one application to deploy, you now have many.
- Operational cost with each service - Each service is a small application
 - Needs own repo, own deployment process, own database, etc
- Additional hardware resources - Additional services need additional hardware to run on
- Error Handling can become complex to implement





How 'Big' Should a Microservice Be?

- A microservice can be as small as a single API endpoint
 - ie - 'Get Orders'
- A microservice can be several or even dozens of API endpoints
- Answer is a topic of much debate
- Guideline - Amazon's Two Pizza Team - A microservice should be able to be supported by a team you can feed with two pizzas. (~12 people)
- Scalability - This can also be a consideration in the size of a microservice
 - The higher the scalability, the more specialized the service should be





How 'Big' Should a Microservice Be?

