Back To Module Home

Architecture of Scalable Applications

0% completed

Different Tiers in Software Architecture

Monolith and Microservices

- What is Monolithic Architecture?
- When should you pick a Monolithic Architecture?
- What is Microservice Architecture?
- When should you pick Microservices Architecture?
- Monolith and Microservices– Understanding the Trade-Offs – Part 1
- Monolith and Microservices– Understanding the Trade-Offs – Part 2

Monolith and Microservices Quiz

Conclusion

Mark Module as Completed

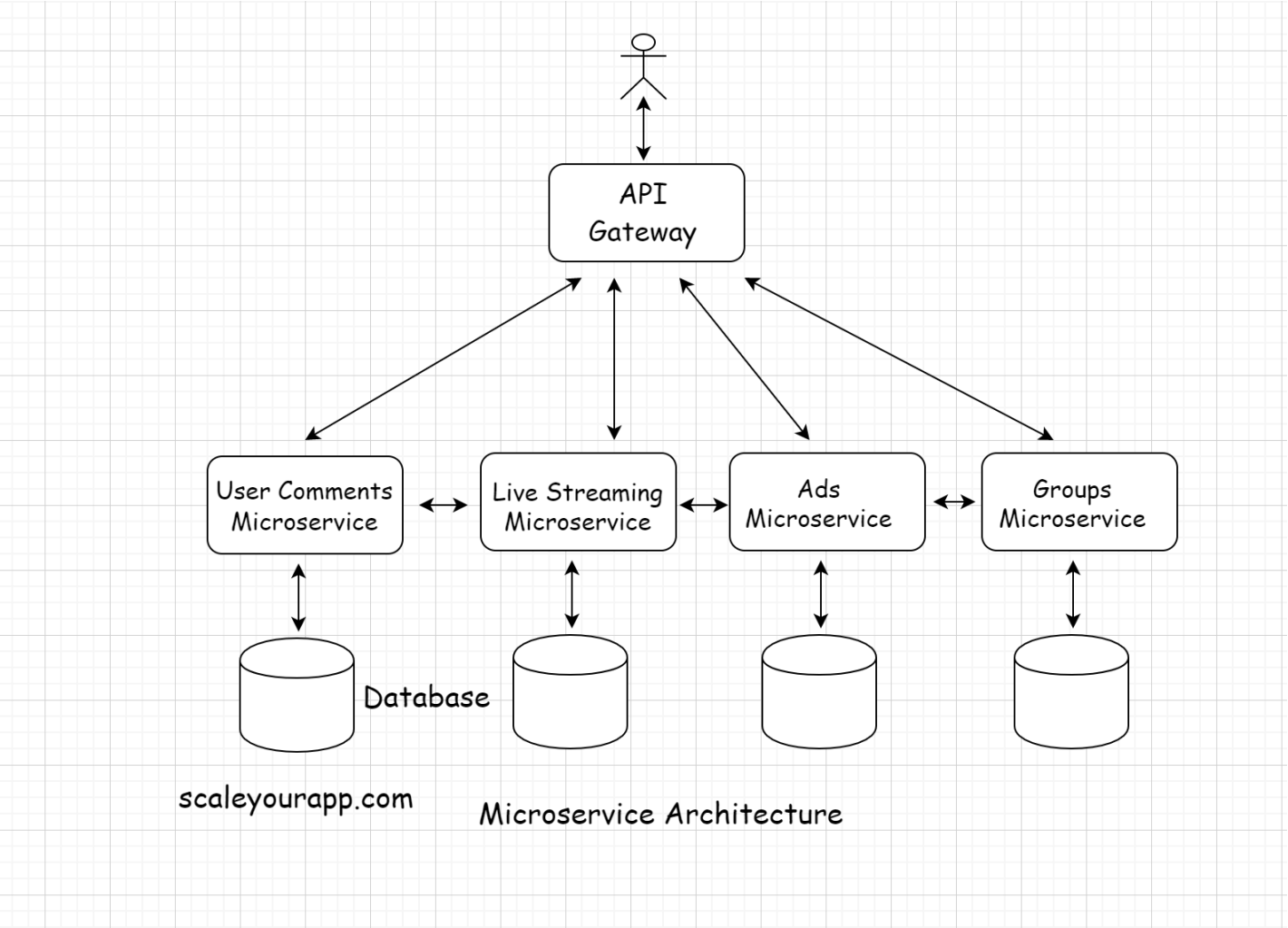
What is Microservice Architecture?

In this lesson, you will learn about microservice architecture.

We'll cover the following

What is microservices architecture?#

> In a *microservices architecture*, different features of an extensive service like *Facebook* are deployed separately as smaller loosely coupled services called *microservices*. These microservices work in conjunction to form a large distributed online service as a whole.



Remember the *single responsibility* and the *separation of concerns* principles? Both principles come into effect in a microservices architecture.

Every service has a single responsibility of running a specific feature and is separated from other services facilitating a loosely coupled architecture.

This particular architecture facilitates easier, cleaner app maintenance, feature development, testing, and deployment of individual modules in contrast to a monolithic architecture.

Imagine accommodating every feature in a single repository. How complex would things get? It would be a maintenance nightmare.

Also, when the project is large, it is managed by several different teams. When application modules/features are separate, they can be assigned to dedicated teams with minimal fuss, smoothing out the development process.

With microservices, scalability becomes easy too. The architecture is inherently designed to scale. Services that need scaling can be scaled independently without affecting other services.

Also, every microservice ideally has a separate database. This eliminates single points of failure and system bottlenecks.

In the next lesson, let's go through some of the pros and cons of using a microservices architecture.

When should you pick a Monolithic Ar…

Next

When should you pick Microservices …

Mark as Completed

Report an Issue