BITS Pilani
Pilani Campus

Microservices demo

Akanksha Bharadwaj
Asst. Professor, CSIS Department

BITS Pilani
Pilani Campus

SE ZG583, Scalable Services
Lecture No. 8

Design and build Microservice

- Demo of making a sample service in Django
- Demo of making a sample service in Flask

Connecting Database to service

- Connect database to Django service
- Connect Database to Flask service

Using Shared DB

- Demo on how can two services communicate with each other by using a shared database

Revision

When to Use Hadoop

- For Processing Really BIG Data
- For Storing a Diverse Set of Data
- For Parallel Data Processing

Kafka Vs Rabbit MQ

- RabbitMQ and Apache Kafka allow producers to send messages to consumers.
- Producers and consumers interact differently in RabbitMQ and Kafka.
  - In RabbitMQ, the producer sends and monitors if the message reaches the intended consumer.
  - On the other hand, Kafka producers publish messages to the queue regardless of whether consumers have retrieved them.
- Kafka is suitable for applications that need to reanalyze the received data. You can process streaming data multiple times within the retention period or collect log files for analysis. Log aggregation with RabbitMQ is more challenging, as messages are deleted once consumed.
- Rabbit MQ suits applications that must adhere to specific sequences and delivery guarantees when exchanging and analyzing data.

When to use CDN

- If your platform serves a global audience, and you aim to provide a seamless viewing experience, then a CDN is a must-have.
- CDN for large high-load websites.

When to use Load balancer

- Load Balancing for Scale
- High availability

Edge computing scenarios

- Enterprise edge
- Operations edge
- Provider edge

When to use CQRS pattern

- Collaborative domains where many users access the same data in parallel.
- You can use the CQRS pattern to separate updates and queries if they have different requirements for throughput, latency, or consistency.

When serverless architecture is not the right choice

- Entirely Serverless application is not suitable for real-time applications that use WebSockets because FaaS functions have limited lifetime
- After some time of being idle, function will require to go through a cold start which can take up to a few seconds.
- Different FaaS providers may differ in some particularities of using their services which will make the switch to another provider troublesome.