Confidentiality

Reliability

Availability

Scalability (Horizontal / Vertical)

Iaas

Paas

Saas

Serverless computing

1. Transaction management in Microservices
2. Communication in microservices
3. Scalable microservices
4. Design and architecture in microservices
5. Java 8 Streams
6. Load Balancer
7. Fault tolerance
8. Archivement in real time project – How to overcome
9. Performance optimization in Project
10. Security feature implementation in project
11. 12 factor methodology in spring boot service
12. Traffic management , request filtering, routing,caching,authentication,authorization
13. Deploy application and update the version without downtime
14. Blue -green deployment
15. Third party tool for Authentication and Authorization

AWS Lambda

AWS Fargate

Docker 🡪 portability , productivity and efficiency

Amazon Elastic Container Service (ECS)

Amazon Elastic Kubernetes Service (EKS)

Amazon Elastic Block Store ( EBS)

IAM Roles

AWS Fargate

Compute Power

* AWS EC2 and Elastic Container Service
* EKS
* AWS Labda computing

Storage :

* S3 , Amazon Elastic cache

Networking :

* Amazon Service Discovery
* AWS App Mesh
* AWS Elastic Load Balancing
* AWS API Gateway
* AWS Route 53 for DNS

Messaging :

* Amazon SQS for message queueing
* SNS for publishing and notify

Monitoring :

* AWS cloud trail for API Monitroing
* AWS CloudWatch for inforstructure monitoring

DevOps – CI/CD

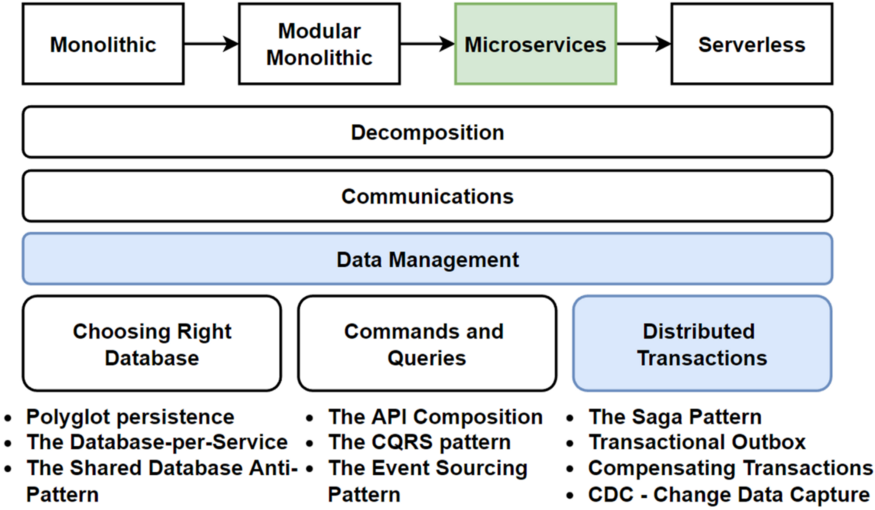
* Amazon ECR

User Instance run on cloud front CDN static content is stored in Amazon S3

* Incoming traffic 🡪 ELB 🡪 Kubernetes cluster
* Data is cached by Elastic Cache and stored in DB

**Example of Microservices Data Management — Main Topics**

Here you can see that, we can divide by 3 for Microservices Data Management Topics:



You can see the parts and related patterns that we will cover during the Practical Test for Microservices Data Managements.

**Sample Questions of this Practical Test**

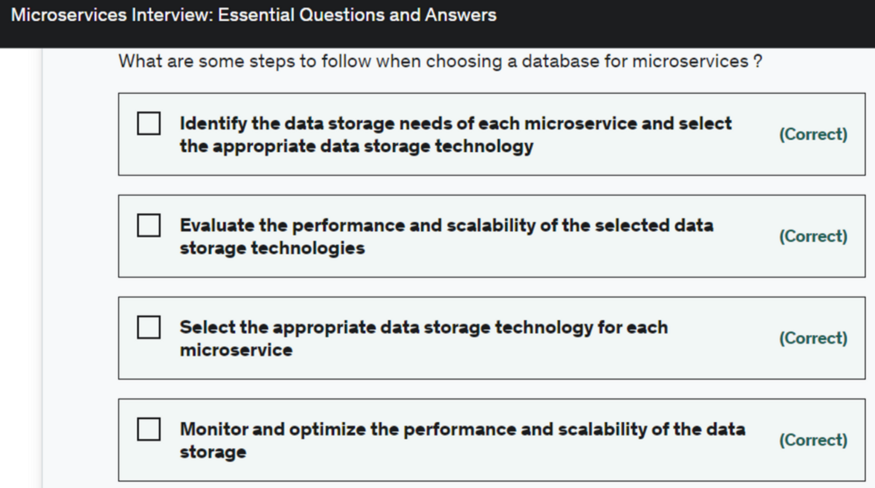
Here you can find the Sample Questions of this Practical Test that you can face these question during the course and you will get very detailed extensive answers for every questions.

* What does “design for failure” mean in a Microservices Architecture ?
* If you design e-commerce application with microservices, Which services could be exist in your architecture ?
* What are some patterns and best practices that are commonly used with asynchronous communication in microservices ?
* What are some advantages of using HTTP and WebSocket for push and real-time communication in microservices ?
* What are some advantages of using the publish/subscribe model and AMQP protocols for event-driven communication in microservices ?
* How does using GraphQL in a microservices architecture compare to using traditional RESTful APIs ?
* Which of the following use case best fit for gRPC usage in e-commerce microservices ?
* What are the use cases of WebSocket API Usage in real-world scenarios ?
* Which of the following is NOT typically a responsibility of an API Gateway ?
* How does the API Gateway apply rate limiting in an microservices-based application ?
* How Service Registry and Service Discovery is used together in microservices ?
* If you design e-commerce microservice application, how you apply Publish/Subscribe Messaging Pattern ?
* If you decide e-commerce databases, What is the data storage technology used by the inventory microservice that processes high-velocity streams of unstructured data ?
* Why is the Shared Database principle considered an anti-pattern for microservices ?
* Which type of database is best suited for a microservice that handles orders and transactions in an e-commerce system ?
* How can the CAP theorem be applied in the design and implementation of microservices ?
* How is database sharding pattern used to improve the performance and scalability of databases in microservices ?
* When should the materialized view pattern be used in microservices ?
* How can read and write databases be kept in sync in a CQRS architecture ?
* Which of the following can be real-world use cases of microservices that use CQRS and Event Sourcing ?
* How does distributed caching improve the performance of a system that uses microservices ?
* What is the main disadvantage of cache miss in a distributed caching system for microservices ?
* What is the advantage of using the saga pattern for implementing distributed transactions in microservices architecture ?
* Which is the example of a real-world scenario where the saga pattern could be used in a microservices architecture?
* In a choreography-based saga, who is responsible for coordinating the updates to the databases ?
* What is the key responsibility of the central orchestrator service in an orchestration-based saga implementation ?
* How can CDC and the transactional outbox pattern be used together in a microservice architecture ?
* What is the role of service meshes in the communication and coordination of microservices ?
* Why use Helm charts in Kubernetes for microservices deployments ?
* When is the sidecar pattern typically used in microservices deployments ?
* How can Infrastructure as code (IaC) be used with Kubernetes for microservices deployments ?
* What is the built-in mechanism in Kubernetes for microservices health monitoring ?
* When using the retry pattern, what is typically done after a temporary failure occurs ?
* What is the Circuit Breaker Pattern ?
* Which of the following is a microservices resilience pattern that helps prevent cascading failures ?
* What are some advantages of using the Strangler Fig Pattern for migrating a monolithic application to microservices ?
* What is the main advantage of using contract testing in a microservices architecture ?

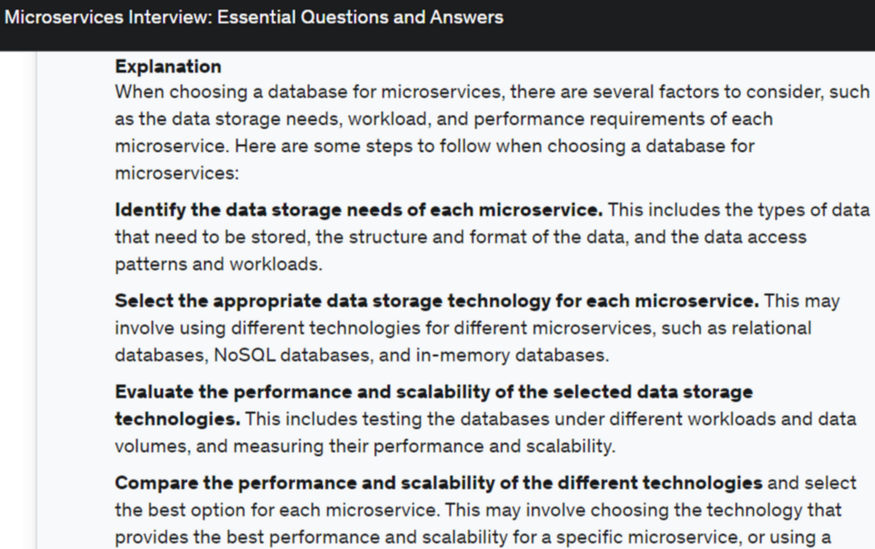
These are kind of questions that need to experience that situation and deeply understand the condition of why we use that particular microservices architecture and patterns.

**How to maximize benefit of this practical test ?**

In order to maximize benefit of this course, After taking this practical test, you should read all answers of every questions even you replied correctly. Because every question has extensive detailed explanations that you can learn all aspects of that topics. Here you can see example question:



Here you can find Example Answer for above question:



As you can see that answers are includes extensive explanation of topics and related topics for microservices. So to understand every topics, I high recommended to check all detailed answer after taking all practical tests.

**Conclusion**

By the end of this course, you’ll have a deep understanding of microservices and the ability to articulate their benefits, drawbacks. Also, you’ll gain the confidence and skills to hack your next software architecture microservices interview and take your career to the next level.

By practicing with this test, You will learn how to demonstrate your knowledge and expertise in microservices with real-world experience and projects, and create strategy for how to effectively communicate and collaborate with the interviewers.

I hope you’ll join me on this journey.