

Question - 1**SCORE: 5 points****Transactions****Microservices** **Easy**

A payment gateway company that integrates PayPal, Stripe, and many other vendors uses microservices for their infrastructure. Whenever a user clicks on the payment button, a request to the backend is made and money is instantly deducted. Whenever a user presses the button twice, the charges are deducted twice. How can you fix this problem?

- Disable the button after it is pressed one time.
- Disable the transaction from the backend once it is approved one time.
- Add an idempotence key to the request and add the transaction to the queue.
- A and C
- A and B

Question - 2**SCORE: 5 points****Microservices Security****Microservices** **Easy**

A company deals has implemented the GPT-3 AI Text Generation using microservices infrastructure for handling thousands of users. The operations team receives a notification that three of their services are down. There is evidence of image pullback failure. Upon investigating, they deduced that one of their services was internet facing and was exploited. The exploited service started messaging other services with corrupted data. The team applied security patches to fix the problem.

What is the fix?

- The team added mTLS and added rules for inter-service communications. After that, they pushed the internet-facing server behind a load balancer and applied JWT tokens for authentication.
- The team cut off the internet-facing service from the infrastructure and applied a JWT token for each call to other services.
- A and B

Question - 3**SCORE: 5 points****Semantic Monitoring Drawback****Easy** **Microservices**

Semantic monitoring (a.k.a. synthetic monitoring) runs a subset of an application's automated tests against the live production system on a regular basis. What are the side effects of semantic monitoring?

- extra load on the services
- There is a probability that tests will inject test data into production.
- There is a possibility that semantic monitoring provides false positives.
- All of the above

Question - 4 Tests in Production

SCORE: 5 points

Microservices Easy

Semantic monitoring (a.k.a. synthetic monitoring) runs a subset of an application's automated tests against the live production system on a regular basis. Which of the following options is the most appropriate and actionable benefit of semantic monitoring?

- decrease in average downtime as we can pinpoint which service is at fault
- detailed monitoring about production environment
- consistent health checks about different services

Question - 5 Conway's Law

SCORE: 5 points

Microservices Easy

A food retail company has an infrastructure based on microservices. While in development the various departments were isolated from each other and were provided with tight deadlines. The department leads informed their teams that if anything went wrong, they could do it over again.

According to Conway's law (the organization system design structure replicates their communication structure), the scenario should be remedied. Which of the following options correctly remedies the problems that arose in the scenario?

- Give up features while adding more communication channels to the team and create independent teams.
- Restructure the team to report to multiple people at the same time.
- Start micro-managing a set of teams by allocating tasks more closely to make delivery effective.
- Restructure and also create independent teams that will dictate their communication channels.

Question - 6 Client Side Load Balancing

SCORE: 5 points

Microservices Easy

A streaming service provider uses a microservices architecture to support its infrastructure. To ensure a seamless user experience, the service uses client-side load balancing, where the client automatically connects to different servers based on availability. However, a security flaw has been identified: the client stores sensitive data, such as server locations and authentication tokens, on the client side, making them vulnerable to exposure and misuse.

What is the most effective way to mitigate this security vulnerability?

- A:** Each service will need to authenticate the requests individually and cannot trust other services.
- B:** Add more layers of authentication and authorization in between the servers/clients
- C:** Add mTLS certificates that expire after 10 minutes.
- D:** B and C

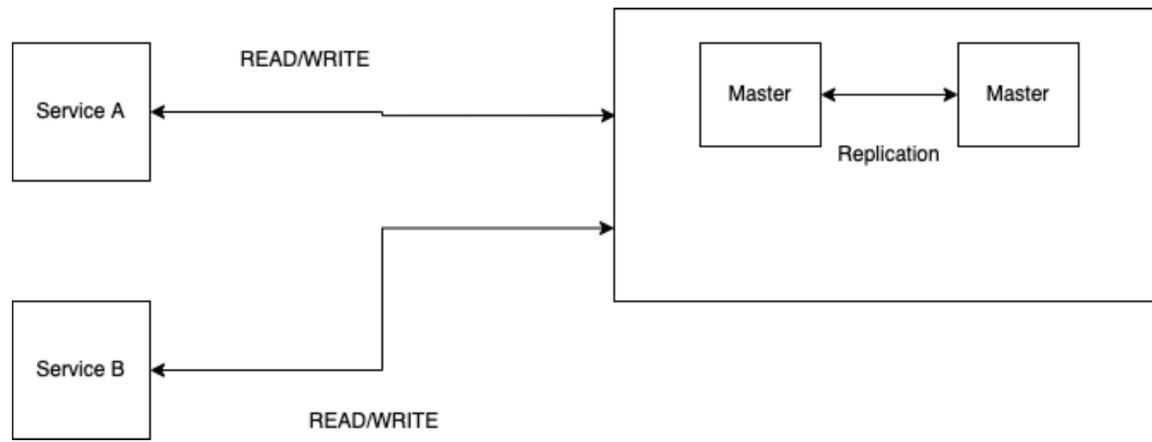
Question - 7

SCORE: 5 points

Master Master Replication 3

Microservices Medium

Consider the diagram below.



A team of developers has successfully refactored the system to resolve problems in consistency, latency, and lag. They would like to test the environment before going to production.

How should the following steps be organized to conduct the tests?

1. Perform E2E (end-to-end) tests first and identify the key differences.
2. Divert some production traffic to the test environment to see how it works i.e.
A/B Testing.
3. Perform unit testing.

- 1 -> 2 -> 3
- 2 -> 3 -> 1
- 3 -> 2 -> 1

Question - 8

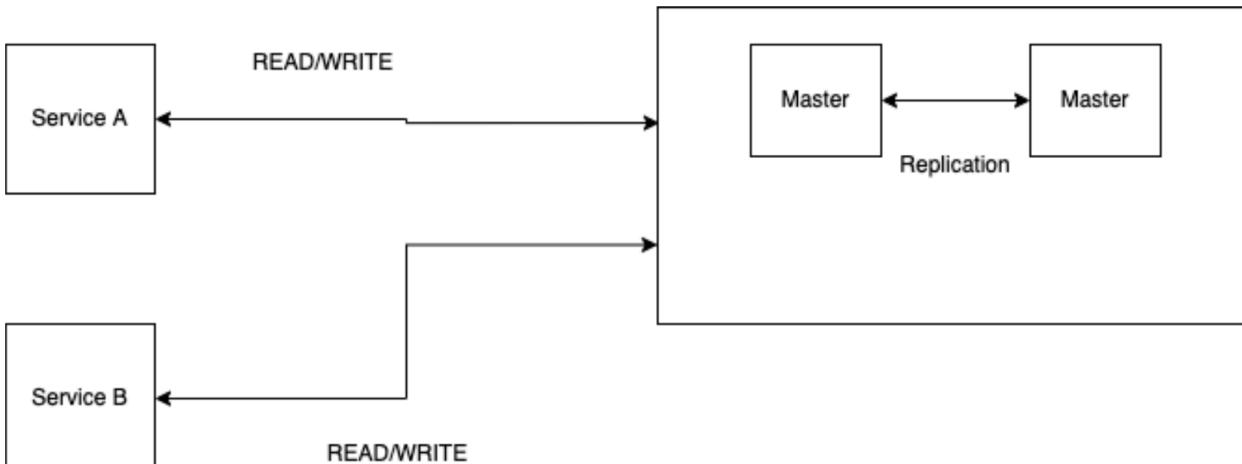
SCORE: 5 points

Master Master Replication 2

Microservices Medium

The team has identified the issues of increased latency, volatile consistency, greater replication lag, and potential for data loss of data in the architecture diagram below. How should the steps be organized to mitigate these issues?

1. Create a separate test environment.
2. Start adding queues and caches.
3. Start refactoring the database.



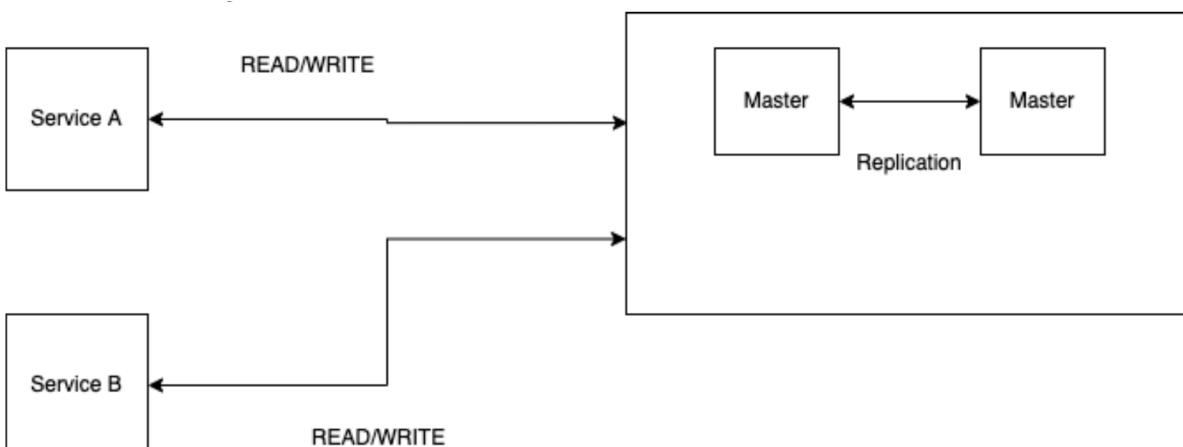
- 3 -> 2 -> 1
- 1 -> 3 -> 2
- 1 -> 2 -> 3
- 2 -> 1 -> 3

Question - 9 Master Master Replication

SCORE: 5 points

Microservices | Medium

Consider the diagram below:



A team of

developers manages Service A and Service B. The services read and write data to the Master-Master replication stateful set. What is a potential flaw in the system?

- increased latency due to synchronization
- volatile consistency due to lag in write
- The more read slaves are added to the system, the more you have to replicate, which leads to greater replication lag.
- potential loss of data if one of the masters fails

SCORE: 5 points

Question - 10

Microservices Reliability

A chat-based app like Whatsapp is using microservices to connect thousands of real-time users together. They have a flaw in the design of the microservices that all the reads and writes go through one service, "Service A". Service A provides them a great deal of volatility and unpredictability. The company would like to refactor the codebase to address the unpredictability of the system.

Which of the following decisions should they go with?

- Separate the read and write APIs into separate services.
- Add a queue to the write API and a cache on the read API so the system can efficiently handle messages.
- Add redundancy to Service A.

Question - 11

Disaster Recovery

SCORE: 5 points

A common cause of failure in microservices is downstream services. Downstream services are the services that are impure. That is, they have dependencies such as database connections, queues, etc. How would you mitigate the failure of these services?

- Add redundancy to the downstream services.
- Add master/worker nodes to the downstream services.
- Add more services to downstream services for better logging and error handling.
- None of the above

Question - 12

Service Discovery

SCORE: 5 points

On March 8, 2022, a music provider detected a global outage in its services. Their system is based on microservices. For microservices to be able to find each other, they utilize service discovery technologies like Google Traffic Detector. The incident happened due to a bug within Google Traffic Detector.

Since this issue was from a third-party provider, what is the likely solution to this problem?

- Migrate service discovery to other vendors, e.g. AWS, and Azure.
- Migrate microservices back to monolith.
- Add multiple service discovery options in the infrastructure.
- Ask the vendor to fix the problem and wait for the fix.

Question - 13 Microservices Usage

SCORE: 5 points

Microservices Easy

A services-based company is building an AI tooling system that can help their client streamline their training and inference data set using AWS Fargate. However Fargate does not provide GPU access to users, and it is expensive for the client. The team decided to implement a custom solution that can scale and train custom models of users. Given the short time span, they have two options:

1. Build the custom Solution using Microservices, LinkerD, MicroK8s.
2. Use an existing system like KubeFlow and build custom features on top of it.

Which of the following is the best way forward?

- Try 2 then 1.
- Try and test 1.
- Try 1 and then 2.

Question - 14 Distributed System Characteristics

SCORE: 5 points

Microservices Medium Distributed Systems

Which of the following are advantages of a distributed system?

- If one component fails in a distributed system, the remaining components may be able to continue operating.
- It is less difficult to implement a distributed database system because of its low cost of installation.
- The amount of processing overhead is less than with a monolithic architecture.
- It overcomes bottlenecks of the processing pipeline easier than with a centralized system.

Question - 15 Understanding Microservice

SCORE: 5 points

Medium Distributed Systems Microservices

Which of the following statements describe the properties of a microservice?

- Focuses on a single responsibility principle
- Too large and complex to fully understand and make changes fast and correctly.
- Responsible for its own task and communicates with other services through simple APIs to solve a larger complex business problem
- Should implement a fully testable end-to-end business use-case

Question - 16

SCORE: 5 points

Which of these is a method of communication between microservices in a larger application?

- HTTP Protocol
- Event-driven or Message-driven
- SOAP
- All of these