20 Microservice Interview Questions with Answers for Java Programmers

Here is a list of some of the frequently asked Microservice questions from Java and Spring Boot interviews. You can use these questions to quickly review important Microservice concepts before interviews.

The list includes questions on Microservice architecture, the pros, and cons of Microservice architecture, Microservices vs Monolith architecture, and essential frameworks and tools required for Microservice development in Java, like Spring Boot, Spring Cloud, Docker, and Kubernetes.

1. What does "Microservices" mean? Explain the term Microservices?
Answer: Microservices is a Systems Development Life Cycle (SDLC) approach in which huge applications are built as a collection of small functional modules. These modules are deployed independently. They are scalable and can communicate with each other over standard protocols.

2. What are the advantages of using Microservices?
Answer: If you have been doing software development then you know that Microservices is the future, the biggest advantage of Microservices is that it fits nicely in cloud infrastructure. By using containers like Docker it's easy to deploy and scale and it also makes the development easier.

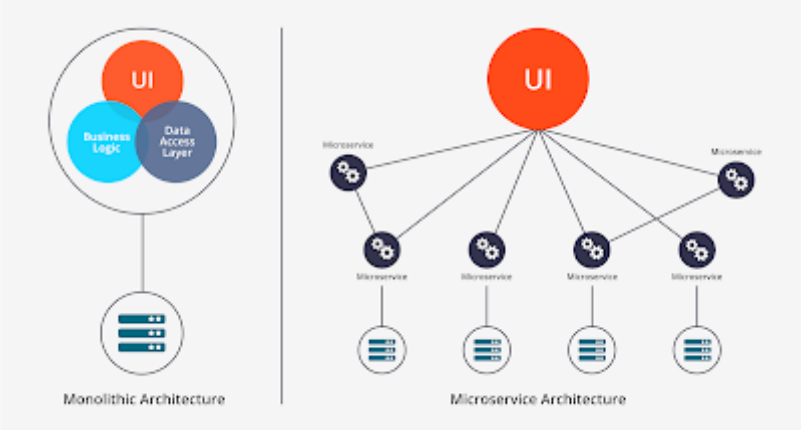
Here are some key advantages of using Microservice architecture:

- They can be deployed independently.
- They are fault isolated.
- They are technologically diverse.
- Deployment time is less.

3. Name the main components of Microservices?.
Answer: Containers, cloud infrastructure, API gateway, Service delivery, IaC, and Service bus.

4. How is Microservice architecture different from Monolithic architecture?
Answer: Monolithic architecture is tightly coupled (mostly) while microservice architecture is loosely coupled. Microservices focus on products while monolithic architecture focuses on the whole project. Moreover, service startups are faster in the microservice architecture.

If you want to learn more about Microservice architecture, I highly recommend **the An Introduction to Microservice Principles and Concepts** course on Educative. This is a text-based interactive course that allows you to run code on a browser.



5. What is the meaning of RESTful?
Answer: RESTful means Representational State Transfer web services. It's based on HTTP protocol and has been the backbone of modern web development which is highly based upon APIs.

6. What is the meaning of OAuth? And why is it used?
Answer: OAuth means open authorization protocol. OAuth is used to access the client applications on HTTP for third-party providers Facebook, GitHub, etc. On Java world, Spring Security supports OAuth 2.0 which you can use to secure your application. If you want to learn more about OAuth and Spring Security I suggest you join **OAuth 2.0 in the Spring Boot Applications** course on Udemy. It's a great course to learn OAuth 2 in depth.

7. What are some challenges faced while using Microservices?
Answer: Here are some challenges faced on building applications on Microservices architecture:
a) Being a distributed system, Microservice architecture is a heavily involved model.
b) Microservices always need to communicate with each other because they always rely on each other.
c) There are always operation overheads.

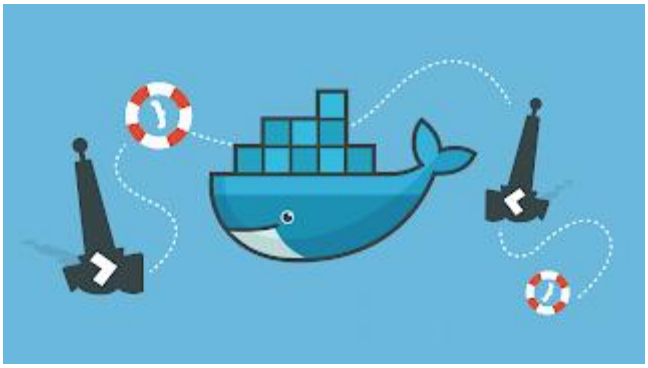
8. What is the use of containers in Microservices?
Answer: Containers are used to manage microservice-based applications. They are easy and effective. Containers also help effectively in deploying and developing individually. The biggest advantages of containers are that they are easy to scale.

You can easily scale your Microservices using tools like Kubernetes which can manage containers at scale. Containers also make deployment uniform, for example, you can deploy a Microservice written in Java or any other programming language in the same way.

9. Explain end-to-end Microservices testing?
Answer: End-to-end microservices testing is a technique in which the entire flow of the application is tested using a business transaction. Such kind of testing covers the gaps left during other testing techniques such as unit and integration testing.

10. In what kind of application we should use microservices?
Answer: Microservices should be used in applications like web, desktop, mobile devices, Smart TVs, etc.

11. What is Docker used for?
Answer: Docker provides a container environment that is used to host applications. It provides a static background for the app to run. Thus, preventing deployment issues. If you want to learn more about Docker and Kubernetes then I highly recommend you to join **Docker & Kubernetes: The Practical Guide** course by Maximillian Schwarzmuller on Udemy.



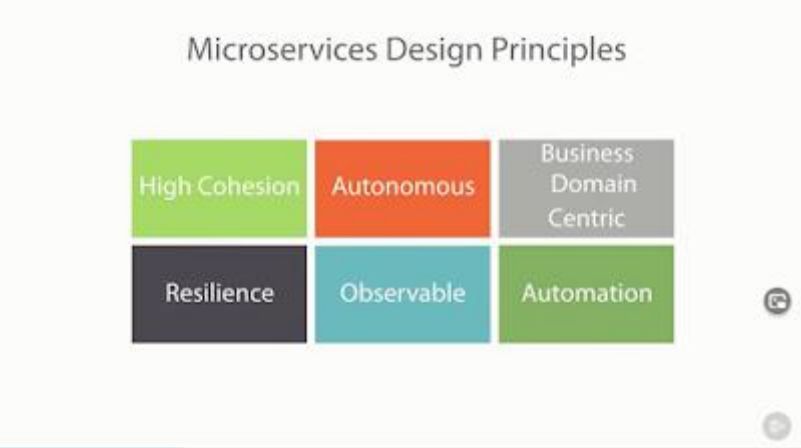
12. What is a "Client certificate"?
Answer: It is a digital certificate. It is used by client systems to make authenticated requests to any remote server. It is highly useful in mutual authentication designs as it provides strong assurances of a requester's identity.

13. How do independent Microservices communicate with each other?
Answer: Microservices can communicate with others through WebSockets for streaming, HTTP for request-response, or brokers.

14. What are some common Microservices design principles?
Microservices architecture is a better way of implementing Service-oriented architecture and following design principles are key for implementing Microservices applications

- High Cohesion
- Autonomous
- Business Domain Centric
- Resilience
- Observable
- Automation

If you want to learn more, I highly recommend you to check out the **Microservices Architecture** course by Rag Dhiman on Pluralsight.



15. Give the major difference between Cohesion and Coupling?
Answer: Coupling is the relationship between two modules while cohesion is the relationship between two or more parts within a module.

16. What are the disadvantages of using microservices?
Answer: While Microservices go hand-in-hand with modern Cloud infrastructure and they are easy to scale, it all comes with a cost of complexity.

Here are some of the notable disadvantages of Microservice architecture

- As a whole, microservices architecture is complicated.
- There is less control over third-party apps.
- Overall end-to-end testing is tough.
- Challenges while deployment.
- Accurate pre-planning is required.

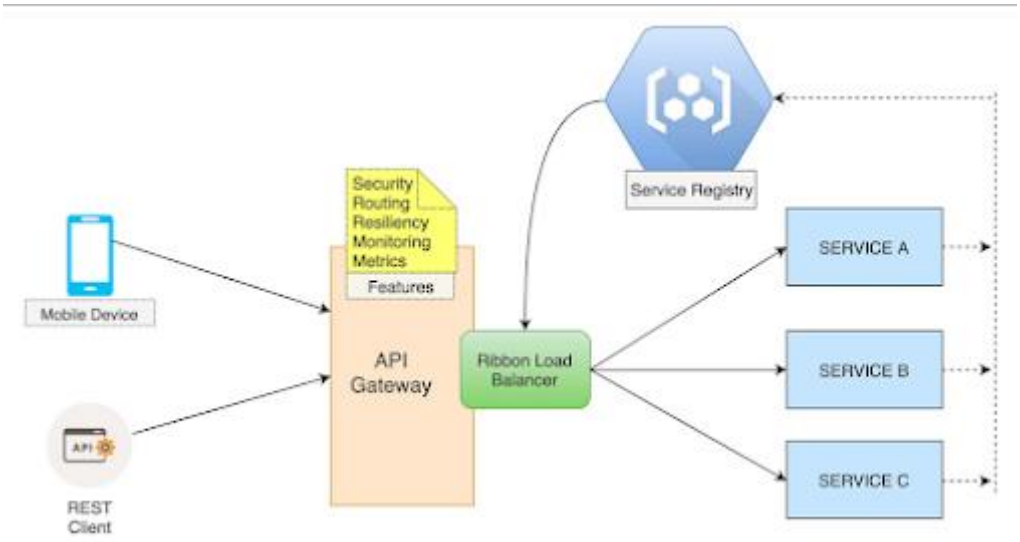
Though the advantages offered by Microservices in a Cloud Computing environment outweigh these disadvantages, it's now becoming a standard way to develop a cloud-native software application.

17. What is difference between API Gateway and Load Balancer in Microservices? (answer)
Load Balancer is an old concept which is used to distribute load or traffic across multiple instances. It can be implemented as a Hardware solution or as a Software Solution but distributing load is its main function and it can be used not just on Microservices architecture but also on Monolith and Service oriented architecture.

On the other hand, API Gateway is a Microservice pattern which not only does load balancing but also can be used for lookup and can also simplify client code. Instead of remember 10s and 100s of Microservices, client can only remember API gateway host and port details.

You can also implement authentication, authorization, security, and other cross-cutting concern at API Gateway level instead of implementing them on each Microservice.

So, in short, API Gateway does a lot more than load balancer.



18. What are few ways to achieve synchronous communication between Microservices?
There are many ways to implement synchronous communication in Microservices architecture like you can use REST API calls, gRPC or GraphQL API to retrieve data from server in synchronous manner and also to upload data to server.

19. How will you achieve asynchronous communication between Microservices?
Well, like synchronous communication there are multiple ways to achieve asynchronous communication in Microservices architecture like you can use Apache Kafka, RabbitMQ, and ActiveMQ. Message brokers allows senders to send and forget and receiver then can process when they are free giving both sender and receiver freedom to work on their speed.

20. What is CQRS Pattern in Microservices? What problem does it solve? (answer)
This one is an interesting question as CQRS is an essential design pattern for Microservice architecture. CQRS stands for Command and Query System in which data operations are separated into two, command which writes data into database and Query system which reads data from database or data store. By splitting your application into two component you can better optimize your application depending upon whether its write heavy or read heavy application. You can read more about CQRS Pattern here.

24 microservices interview questions and answers

1. What is the microservices architectural style?

The microservices architectural style is commonly used to build enterprise applications and distributed systems. It's a software development approach that develops a web application through the collective efforts of several independent standalone applications, or microservices, that can be run on different machines. These individual microservices are highly specialized, and upon execution, form a single application.

2. What are some advantages of microservice architecture?

Advantages of microservice architecture include:

- Supports system scalability, testability, and maintainability
- Developer teams can focus and specialize on specific microservices
- Enables the setup of continuous delivery pipelines
- Microservices often communicate using HTTP, so software developer teams can use their preferred tech stacks or combination of languages on the backend of their respective microservice applications
- As it prevents single points of failure, the entire application won't crash if one component crashes
- Possibility of containerization
- Microservices are separate applications, and as such they can be deployed and scaled separately

3. What are some disadvantages of microservices?

Disadvantages of microservice architecture include:

- **New complexities**: While microservices help reduce complexity in the codebase, the communication and networking between components in the application becomes more complex (e.g. more difficult to track paths of communication because of interaction with all the separate components).
- **Infrastructural overhead**: Many resources are needed to set up microservice architecture. While upfront costs may be high, long-term savings are a benefit.

4. What is loose coupling and high cohesion?

Loose coupling and high cohesion are core principles in microservices design. Loose coupling advocates for the separation of unrelated components, while high cohesion advocates for the grouping together of related components.

5. What is the single responsibility principle?

In microservice design, the single responsibility principle requires each microservice application to have just one focused responsibility.

6. What's the relation between microservice architecture and DevOps?

DevOps is a methodology that combines practices from both software development and IT operations. It strives to reduce the system development life cycle. Both DevOps and microservices architectural style emphasize the importance of continuous delivery and agile development. As such, **the methodology and software development approach are complementary and often used together**.

7. What are the advantages of microservices over monolithic architecture?

Compared to monolithic architecture, the advantages of microservice architecture include:

- **Easier to manage than one large complex codebase**: Breaking the entire application into individual microservice applications makes it easier to test, debug, and scale the entire system.
- **Easier for teams to work on**: Development teams can focus on one application at time, instead of one large complex codebase
- **Better flexibility**: When traffic increases, only a small portion of the app needs to be scaled in times of increased traffic (eg. server).
- **Shorter startup time**: Microservices have a faster service startup than monolithic applications
- **Less dependencies**: Loosely coupled architecture reduces dependencies and supports system scalability, maintainability, and flexibility

8. What are the differences between microservice architecture and service-oriented architecture (SOA)?

While microservice architectural style is a variation of SOA, there are various differences between SOA and microservices architecture.

Microservices:

- More scalable
- No dependencies between business units
- Prioritizes decoupling
- Lightweight communication protocols such as HTTP and REST

SOA:

- Less scalable
- Dependencies between business units
- Prioritizes reusability
- Supports various communication protocols

9. What is a client certificate?

A client certificate is a digital certificate that allows a client system to make authenticated requests, by verifying the requester's identity. Without a client certificate, client systems can't send requests to remote servers.

10. What is Kubernetes?

Kubernetes is a container management platform and open-source tool. It's used to deploy and manage containerized workloads.

11. What is Eureka?

Eureka is the Netflix Service Discovery Server.

12. What is service discovery?

Service discovery is the automated detection of services and devices on a network. Service discovery helps keep download distributed appropriately between microservices during deployment, where the number of active microservice instances are in constant flux.

13. What is a distributed transaction?

A distributed transaction is a database transaction that involves **more than one physical server** or network host.

14. What is domain-driven design?

Domain-driven design is a software development framework that models software in accordance with the model of the underlying business domain. **Because rules, definitions, and protocols vary across a large business domain model, domain-driven design suggests that we identify individual bounded contexts within our domain**, wherein rules, entities, and services are consistently applied. Bounded contexts can interact with each other, but they are separated by their own respective domain models that define their functionality.

15. What is end-to-end microservices testing?

End-to-end testing evaluates a microservice application from start to finish through an entire business transaction. End-to-end microservices testing can become very complex with microservices.

16. What is a consumer-driven contract (CDC)?

CDC is a **testing method** that ensures the compatibility of services based on requirements defined by the consumers. The contract refers to an agreement between consumer and provider about the format of data transfer. CDC tests then are performed by both consumer and provider to ensure the contract is continually honored. PACT is an open-source tool that provides a CDC testing framework.

17. What is idempotence in microservice architecture?

Idempotence in a microservice is achieved if the service provides a consistent output in the case of duplicate messages. Idempotence is critical in ensuring consistent behavior from services, with no unintended side effects in the case that it receives duplicate requests.

18. What communication styles are used in microservices architecture?

Synchronous request-based communication such as HTTP and REST are great for deploying services outside your microservice cluster or containerization platform. For internal communication between services, binary format communication mechanisms such as WCF using TCP and binary format are a good choice, or asynchronous message-based comm mechanisms like AMQP. Message formats can also be JSON, XML, and binary formats.

19. What is Docker?

Docker is a containerization platform. It's an open source tool developers use to deploy microservice applications.

20. What is Hystrix?

Hystrix is a latency and fault tolerance library. Hystrix helps manage latency and failures by isolating the points of access between microservices.

21. What are RESTful web services?

Restful web services are web services that follow the REST architectural style. RESTful web services communicate with each other using CRUD operations. The REST architectural style emphasizes scalability and flexibility, which makes it compatible and frequently preferred for microservice architecture.

22. What is OAuth?

OAuth means open authorization protocol. It's a security standard that's used to secure endpoints.

23. What is Spring Boot?

Spring Boot is a Java open-source tool that enables microservice application development. You can use the Spring Boot application to streamline microservice development with the Spring Framework.

24. What are the benefits of using the Spring MVC framework?

The Spring MVC framework has many useful benefits, including:

- Data binding
- Security and testing capabilities
- Annotation system
- Enables separation of concerns
- Helpful for developing REST APIs and web services

1. **What do you understand by Monolithic Architecture?**
    - Monolithic architecture is like a big container that contains all the software components of an application.
    - This applications are clubbed inside a single package within the application.
2. **What are the main components of Microservices?**
    - Containers, Clustering, and Orchestration
    - IaC (Infrastructure as Code Conception)
    - Cloud Infrastructure
    - API Gateway
    - Enterprise Service Bus
    - Service Delivery
3. **Which are some famous companies that are using Microservice architecture?**
    - Most large-scale software companies and websites such as Twitter, Netflix, Amazon are using microservices architecture instead of monolithic architecture.
4. **What are the biggest challenges in Microservice deployment?**
    - It requires a heavy infrastructure setup also.
    - Microservices require a heavy investment.
    - We need excessive planning for managing operations overhead.
    - It cost a lot in staff selection and maintenance.
5. **What are the three commonly used tools for Microservices?**
    - Wiremock
    - Docker
    - Hysrix
6. **What do you understand by Spring Cloud?**
    - Spring cloud is an Integration software used to integrate with external systems.
    - It allows a microservices framework to build applications that perform restricted amounts of data processing.
7. **What are the main advantages of using Microservices?**
    - Microservices provide great technology diversity. You can mix it easily with other frameworks, libraries, and databases.
    - It provides excellent support for the minor and parallel team.
    - It reduces the deployment time significantly.
    - Independent deployment
8. **How does a Microservice architecture work?**
    - An application is fragmented into loosely coupled various modules, each of which performs a distinct function.
    - It is distributed across clouds and data centers.
    - Under microservice architecture, an application can grow along with its requirements.
9. **What are the three types of tests used in Microservices?**
    - Bottom Level Test
    - Middle-Level Tests

- Top Level Tests

10. What is the main difference between SOA and the Microservices Architecture?

- SOA stands for Service Oriented Architecture.
- It is a collection of services used to communicate with each other through simple data passing or activity coordination.

11. What are the most significant disadvantages of using Microservices?

- It requires accurate pre-planning before use.
- It uses modular dependencies that are hard to calculate.
- The third-party applications are hard to control.
- More opportunities for malicious intrusions.
- Complete end-to-end testing is complex.
- Deployment Challenges.

12. What is a Client certificate? What is its usage?

- A client certificate is a digital certificate used to make authenticated requests to a remote server.
- A certificate is generated for each microservice.

13. What are the different strategies used in Microservices deployment?

- Multiple Service Instance per Host
- Service Instance per Host
- Service Instance per Container
- Serverless Deployment

14. What are the most significant benefits of using microservices?

- The most significant benefit of using microservices is that it builds an application to collect small autonomous services developed for a business domain.
- Business needs to change constantly, the development teams can rapidly build new apps components to meet the requirement.

15. What do you understand by RESTful?

- REST or RESTful stands for Representational State Transfer.
- The RESTful web service is an architectural style that helps computer systems to communicate over the internet.
- This web services make microservices easier to understand and implement.

16. What are the most significant advantages of using Microservices?

- Provide improved scalability
- Increased Agility
- Localized Complexity
- Provide fault isolation
- Smaller development teams
- You can easily upgrade the technology etc

17. In which cases microservice architecture is best suited?

- The microservice architecture is best suited for all tech devices such as desktop, web, mobile devices, Smart TVs, Wearable devices, etc.

18. What are the principles Domain-Driven Design?

- Focus on the core domain and domain logic.

- Base complex designs on models of the domain.
- Collaborate with the domain experts to improve the application model and resolve any emerging domain-related issues regularly.

19. What do you understand by Domain-Driven Design?

- Domain-Driven Design is an architectural style based on Object-Oriented Analysis Design concepts and principles.
- It is used to develop a complex system by connecting the related components of the software system into a continuously evolving system.

20. What do you understand by semantic monitoring in Microservices architecture?

- Semantic monitoring is used to combine the automated tests by monitoring the application.
- It is used to find out the reasons why your business is not getting more profits.

21. What are the main differences between Microservices and Monolithic Architecture?

Microservices	Monolithic Architecture
It is a loosely coupled architecture.	It is primarily a tightly coupled architec
The service startup is fast in Microservices.	The service startup takes time as it is slo Monolithic Architecture.

22. What do you understand by OAuth?

- OAuth stands for Open Authorization protocol.
- This protocol allows you to access the client applications on HTTP for third-party providers GitHub, Facebook, etc.
- It also facilitates us to share resources stored on one site with another site without their credentials.

23. What is Spring Cloud?

- Spring Cloud is a collection of tools used by developers to quickly build some of the common patterns in distributed systems such as configuration management, circuit breakers, service discovery, intelligent routing, micro-proxy, control bus, one-time tokens, global locks, leadership election, distributed sessions, cluster state and more.

24. What are Microservices?

- Microservices are an architectural approach or style that is used to build applications.
- The microservice architecture provides a rapid, frequent and reliable delivery of large and complex applications.

25. What is the use of containers in Microservices?

- Containers are the easiest and effective method to manage microservice-based applications.
- They are like a software development platform.
- They also help us to develop and deploy individually.

26. What is Spring Boot? Why is it used?

- Spring Boot is an open-source, Java-based framework that provides developers an excellent platform for developing a stand-alone and production-grade spring application.
- It is easy to understand, reduces development time, and increases productivity. It automatically configures a claim based on the added dependencies of an application.

27. What is the difference between Coupling and Cohesion?

Coupling	Cohesion
Coupling is specified as a relationship between module A and another module B. There are mainly three types of coupling	Cohesion is the relationship between 2 or parts within a module.
Any module can be highly coupled (highly dependent), loosely coupled, and uncoupled with other modules.	The high cohesion within a module specifi the module can perform a specific task maximum efficiency on its own, without the communicate with other modules.

28. What do you understand by end-to-end Microservices testing?

- End-to-end testing is used to validate that every process in the workflow is functioning correctly.
- It also ensures that the system works together as a whole and fulfills all the requirements.

29. What is the method to override a Spring Boot project's default properties?

- We can do it by specifying the properties in application properties.
- The Spring MVC applications need the suffix and the prefix to be specified i.e For suffix:mvc.view.suffix: .jsp For prefix**:**mvc.view.prefix: /WEB-INF/

30. What is the use of PACT in Microservices architecture?

- PACT is an open-source tool used for testing interactions between service providers and consumers.
- It increases the reliability of the Microservices applications.