

Microservices Interview Questions

Posted in Microservices (<https://hackr.io/blog/category/microservices>)



 **Barnali Chanda**
(<https://hackr.io/blog/author/barnali.sen>)
Last Updated 14 Aug, 2020







Share:   (<https://twitter.com/intent/tweet?text=Microservices+Interview+Questions+https%3A%2F%2Fhackr.io%2Fblog%2Fmicroservices-interview-questions>)  (<https://www.linkedin.com/shareArticle?mini=true&url=https://hackr.io/blog/microservices-interview-questions>)  (<http://www.reddit.com/submit?url=https://hackr.io/blog/microservices-interview-questions>)  (<https://news.ycombinator.com/submitlink?u=https://hackr.io/blog/microservices-interview-questions>)  (<https://api.whatsapp.com/send?text=https%3A%2F%2Fhackr.io%2Fblog%2Fmicroservices-interview-questions>)

Table of Contents

Microservices architecture is a method of developing a large-scale application in the form of discrete modules that work independently of each other but as a whole account for the entire business application under consideration. Organizations are increasingly adopting this service for developing enterprise solutions. As this operates on simple, modular services, there is now an increasing demand for cloud solutions.

Microservices deployment can be Cloud-Based or On-Premise. It's market spans across Healthcare, Banking, Government, Manufacturing, Retail, IT & Telecommunication, Media & Entertainment, Manufacturing, etc. Its intended audience includes Investors and Consultants, System Integrators, Government Organizations, IT & Technology Solution Providers. To name a few, prominent players in this field of expertise are Cognizant, IBM, Microsoft, Salesforce, and Software AG. The microservices architecture market is expected to take a massive leap by the year 2023.

If you are looking into a career in this field and are getting yourself interview ready, here are a few quick microservices interview questions and answers you can expect.

Microservices Interview Questions

The questions and answers listed here have been compiled from many resources off the internet. This should not be considered as the ultimate guide for microservices interview. Please acquire in-depth knowledge in this field and consult more books and resource guides for connecting questions in this regard.

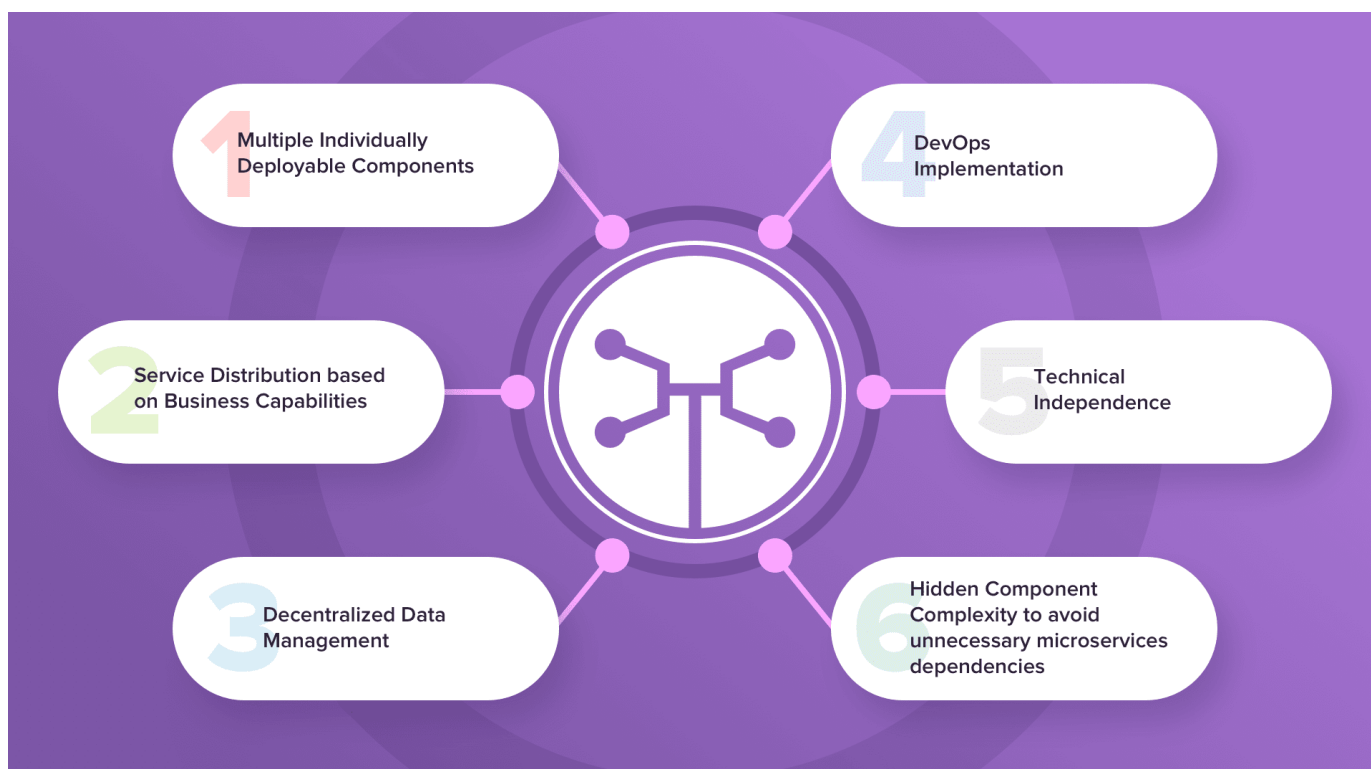
Question: What do you understand by Microservices?

Answer: Microservices or more appropriately Microservices Architecture is an SDLC approach based on which large applications are built as a collection of small functional modules. These functional modules are independently deployable, scalable, target specific business goals, and communicate with each other over standard protocols. Such modules can also be implemented using different programming languages, have their databases, and deployed on different software environments. Each module here is minimal and complete.

Question: What are the main features of Microservices?

Answer: Microservices have the following main features:

- Multiple Individually Deployable Components.
- Service Distribution based on Business Capabilities.
- Decentralized Data Management.
- DevOps Implementation.
- Technical Independence.
- Hidden Component Complexity to avoid unnecessary microservices dependencies.



Question: What are the main components of Microservices?

Answer: The main components for a Microservice Architecture are:

- Containers, Clustering, and Orchestration
- IaC [Infrastructure as Code Conception]
- Cloud Infrastructure
- API Gateway
- Enterprise Service Bus
- Service Delivery

Question: How does a Microservice Architecture work?

Answer: Under a microservice architecture, an application is simplified into multiple modules that independently perform the single precise standalone task:

- Are fragmented into loosely coupled various modules, each of which performs a distinct function.
- It can be or are distributed across clouds and data centers.
- Implement each module as an independent service/process which can be replaced, updated, or deleted without disrupting the rest of the application.

Under microservice architecture, an application can grow along with its requirements.

Question: What are the fundamental characteristics of a Microservices Design?

Answer:

- Services split up and organized around business functionality.
- Separate modules handled and owned by different development teams.
- Decentralized Framework.
- Maintenance of respective modules by respective development teams.
- Separate modules may be maintained by different databases.
- Modules in a Microservice Architecture are separately deployable. They can be updated, enhanced, or deleted without disrupting the entire architecture.
- Real-time monitoring of the application.

Question: What are the main challenges in Microservice Deployment?

Answer: The challenges in Microservice can be both technical as well as functional.

From the point of business, the main challenges are:

- Require heavy investment
- Heavy Infrastructure Setup
- Excessive Planning for managing operations overhead
- Staff Selection and maintenance.

From a technical standpoint –

- Communication between different microservices in the application.
- Component automation
- Application maintenance
- Configuration Management
- Heavy Operations Overhead
- Deployment Challenges
- Testing and Debugging Challenges

Question: What are the advantages and disadvantages of Microservices?

Answer:

Advantages:

- Improved Scalability
- Fault Isolation
- Localized Complexity
- Increased Agility
- Simplified Debugging & Maintenance
- Better correspondence of developers with business users.
- Smaller development teams
- Better scope for technology upgradation.

Disadvantages:

- Complicated as a whole.
- Requires accurate pre-planning

- Modular dependencies are hard to calculate.
- Less control over third party applications
- Modular Interdependencies are challenging to track.
- More opportunities for malicious intrusions.
- Complete end-to-end testing is difficult.
- Deployment Challenges.

Question: What are the different strategies of Microservices Deployment?

Answer:

- **Multiple Service Instance per Host:** Run single or multiple service instances of the application on single/multiple physical/virtual hosts.
- **Service Instance per Host:** Run a service instance per host.
- **Service Instance per Container:** Run each service instance in its respective container.
- **Serverless Deployment:** Package the service as a ZIP file and upload it to the Lambda function. The Lambda function is a stateless service that automatically runs enough micro-services to handle all requests.

Question: List the differences between Monolithic, SOA, and Microservices Architecture with an example for each.

Answer:

In Monolithic Architecture, all software components of the application are assembled and packed tightly.	SOA [Service Oriented Architecture] is a collection of services that communicate with each other through simple data passing or activity coordination.	Microservices Architecture is a collection of small functional modules. These functional modules are independently deployable, scalable, target specific business goals, and communicate with each other over standard protocols.
--	--	---

Question: What is Domain Driven Design?

Answer: Domain-Driven Design is an architectural style based on Object-Oriented Analysis Design concepts and principles. It helps in developing a complex system by connecting the related components of the software system into a continuously evolving system. Domain-Driven Design is based on three core principles:

- Focus on the core domain and domain logic.
- Base complex designs on models of the domain.
- Regularly collaborate with the domain experts to improve the application model and resolve any emerging domain-related issues.

Question: What is the Spring Boot?

Answer: Spring Boot is an open-sourced, Java-based framework (<https://hackr.io/blog/java-frameworks>) that provides its developers with an excellent platform for developing a stand-alone and production-grade spring application. It is easy to understand, increases productivity, and reduces development time. It automatically configures a claim based on the added dependencies of an application.

Question: How do you override a Spring Boot Project's Default Properties?

Answer: Specify the properties in application.properties.

Spring MVC applications need the suffix and the prefix to be specified. This can be done by:

- **For suffix** – spring.mvc.view.suffix: .jsp
- **For prefix** – spring.mvc.view.prefix: /WEB-INF/

Question: How do you implement a Spring Security in a Spring Boot Application?

Answer:

- Add spring-boot-starter-security in the file pom.xml.
- Create a Spring config class that will override the required method while extending the WebSecurityConfigurerAdapter to achieve security in the application.

Question: How will you Configure Spring Boot Application Login?

Answer: Spring Boot Application Login can be configured by specifying the logging.level in the application. Properties file. It is usually pre-configured as console output.

Question: What is Spring Cloud?

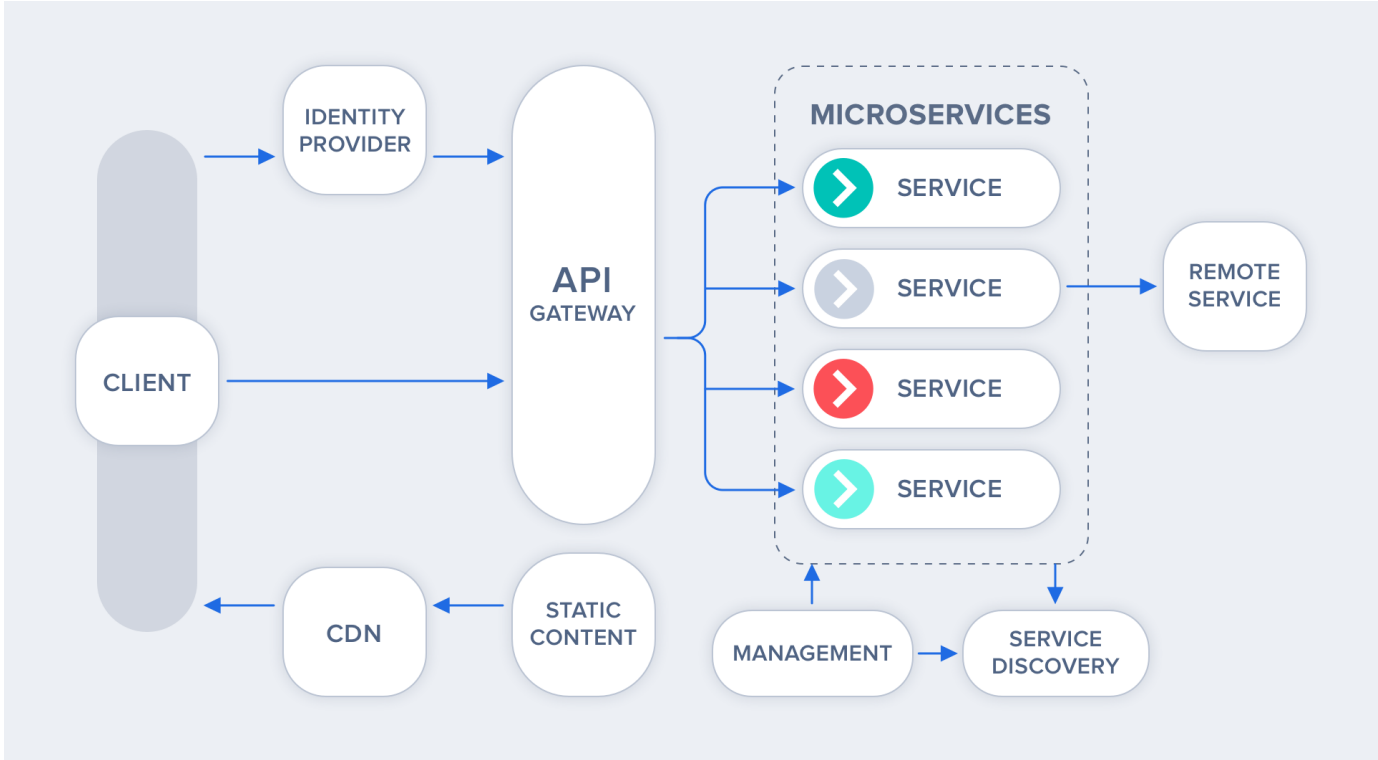
Answer: While building a distributed system, there are a few problems that are encountered. They are Configuration Management, Service Discovery, Circuit breakers, and distributed sessions. Spring Boot is a collection of tools that provides solutions to such commonly encountered problems.

Question: What is an Actuator?

Answer: The actuator brings in production-ready features into an application. It is mainly used to expose operational information about the running application's health, metrics, info, dump, env, etc. It uses HTTP endpoints or JMX beans to interact with it.

Question: What is a Container?

Answer: Containers are isolated workload environments in a virtualized operating system. It consists of an application and all the features and files required to run it. Each box is an independent environment and is not tied to the software on a physical environment, thus providing a solution for application portability.



Question: What are Coupling and Cohesion?

Answer: The strength of dependencies between services in a microservice architecture is said to be coupling. Cohesion refers to the related logic between two or more services. The entire concept of microservices is based on the ability to deploy and update service while keeping other services intact. Hence, loose coupling and high cohesion is the key to a microservice design.

Question: What is PACT in Microservices Architecture?

Answer: A contract between a consumer application and a provider application is called a PACT. Each pact is a collection of interactions. It is an open-source tool that can be used to implement the Consumer-Driven Contract in Microservices.

Question: What is Contract Testing?

Answer: Contract Testing ensures that the explicit and implicit contracts of a microservice architecture work as expected. There are two perspectives to contract to test – Consumer and Provider. The consumer is the [application] entity using the microservice, and the provider is the [application] entity providing the service. Such services work under predefined specifications, and contract testing ensures so.

Question: What is OAuth?

Answer: OAuth stands for Open-standard Authorization Protocol or framework that describes how unrelated servers and services can safely allow authenticated access to their assets without sharing the initial related, single logon credential. This is also known as secure, third-party, user-agent, delegated authorization.

Question: What can you derive/understand from Conway's Law?

Answer: Melvin Conway stated this idea in the late 1960s. This law implies that ‘organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.’ In simpler words, the team of people who design a piece of software will eventually make the design as per their perspective.

Question: What are the steps in an End-to-End Microservices Testing?

Answer: End-To-End testing of a microservice application ensures that every process in the form is running properly. This validates that the system as a whole is working properly. As the microservices application is built with multiple modules orchestrated dynamically, an end to end testing covers all the gaps between the services.

The steps to an end-to-end microservices testing are:

- Define what you expect from an e2e testing.
- Define the scope of the system to be tested.
- Perform authentication in a test environment.
- Choose a testing framework that addresses most of the issues.
- Test Asynchronous flows
- Automate Testing

Question: What is the difference between Mock & Stub?

Answer: A Mock is generally a dummy object where certain features are set into it initially. Its behavior mainly depends on these features, which are then tested.

A Stub is an object that helps in running the test. It functions in a fixed manner under certain conditions. This hard-coded behavior helps the stub to run the test.

Question: What can we derive from Mike Cohn's Test Pyramid?

Answer: Mike Cohn's Test Pyramid describes the type of automated tests required for software development. The Test Pyramid is only a metaphor that implies a grouping of tests based on their granularity. This Pyramid tells which kind of tests should be applied to different levels of the pyramid.

Mike Cohn's test pyramid consists of three layers that a test suite should consist:

- Unit Testing
- Service Testing
- User Interface Testing.

Two points to be derived from Cohn's pyramid is that:

- Define tests with different granularity
- The higher in the level you get, the fewer tests you should have.

Question: How does Docker help in Microservices?

Answer: Microservices, as we know, are self-contained, individual units that perform only one business function, so much so that each unit can be considered an application on its own. The application development environment and application deployment environment are bound to vary in many aspects. This gives rise to deployment issues. Docker provides a static background for the application to run, thus avoiding deployment issues. It is, in fact, a containerization tool. It reduces overhead and deploys thousands of microservices on the same server. Docker ensures that an application microservices will run on their own environments and are entirely separate from their operating system.

Question: What is Canary Releasing?

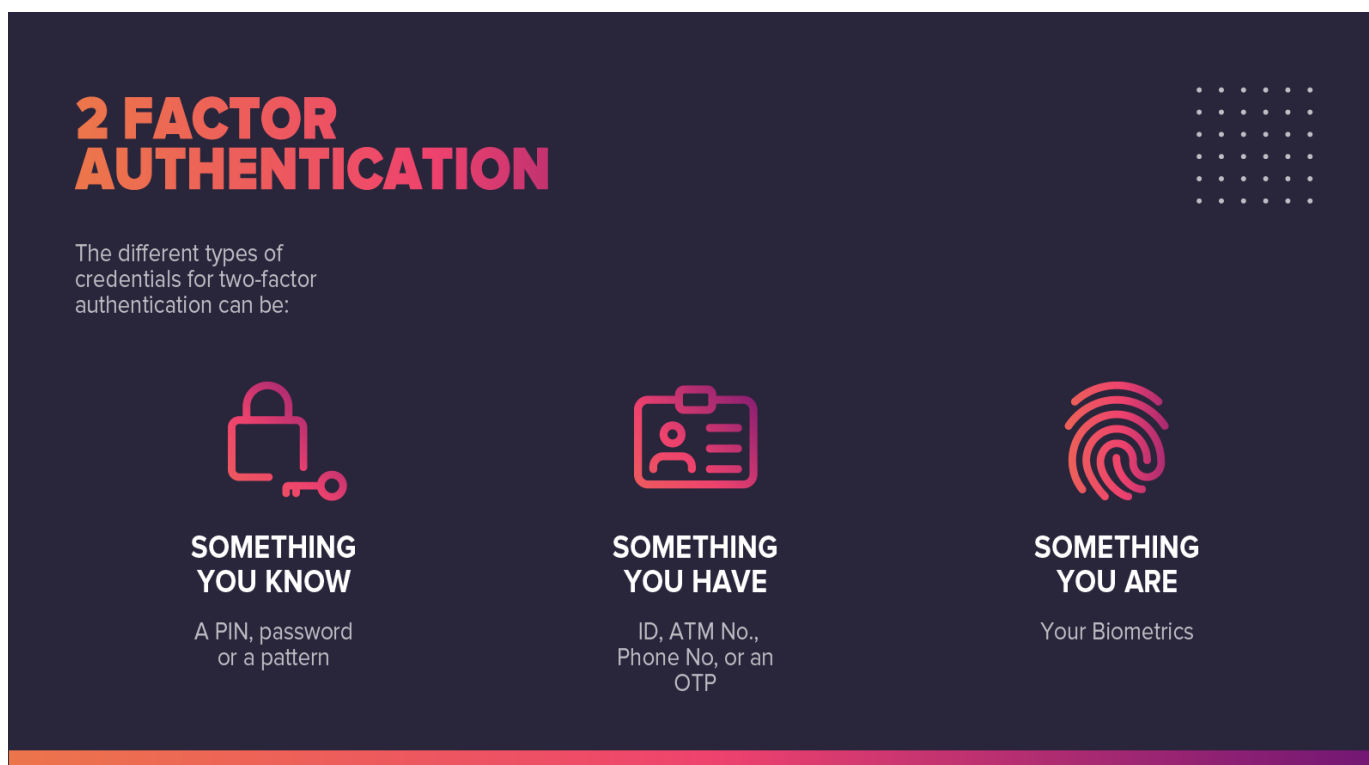
Answer: Canary releasing is a technique by which new software versions are introduced by rolling out the updated version to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody. This technique is so-called because it is based on canary releases in coal mines to alert miners when the toxic gases reach dangerous levels.

Question: What are the different types of credentials of a two-factor Authentication?

Answer: Two-factor authentication calls for a second round of authentication to an account login process. Entering a username – password is a one-factor authentication. Username – Password authentication and then say mobile number or a secret key authentication can be considered as a two - factor authentication.

The different types of credentials for two-factor authentication can be:

- Something you know - A PIN, password or a pattern
- Something you have – ID, ATM No., Phone No, or an OTP
- Something you are – Your Biometrics.



Question: What is a Client Certificate?

Answer: A Client Certificate is a digital certificate that is used by client systems to make authenticated requests to a remote server. It plays a key role in many mutual authentication designs, providing strong assurances of a requester's identity.

Question: What is a CDC [Consumer Driven Contract]?

Answer: Consumer-Driven Contracts are patterns for evolving services. Here, each consumer captures their provider in a separate contract. All these contracts are then shared with the provider, which helps them to gain an insight into the obligations they must fulfill for each individual client.

Question: What are non-deterministic tests, and how will you eliminate them?

Answer: NDT or Non-Deterministic are unreliable tests. Such tests sometimes pass and sometimes fail. When these tests fail, they are re-run to give. Non-Determinism from tests can be eliminated in the following ways:

- Quarantine
- Asynchronous
- Remote Services
- Isolation
- Time
- Resource Leaks

Question: What are Reactive Extensions in Microservices?

Answer: Reactive Extensions is a design approach through which results are collected by calling multiple services in order to compile a combined response. Also known as Rx, these calls can be synchronous or asynchronous.

Question: What is the role of RESTful APIs in Microservices?

Answer: A microservice is based on the concept where all its component services require to interact with one another to complete the business functionality. This requires each microservice to have an interface. RESTful APIs provide a logical model for building these interfaces. It is based on the open networking principles of the Web. Thus, it serves as the most critical enabler of microservices.

Question: What is Eureka in Microservices?

Answer: Eureka is alternatively known as the Netflix Service Discovery Server. It uses Spring Cloud and is not heavy on the application development process.

Question: How will you balance the server-side load by utilizing Spring Cloud?

Answer: The server-side load balancing can be done by using Netflix Zuul. It is also known as a JVM based router.

Question: When will you see fit to use the Netflix Hystrix?

Answer: Hystrix is an error tolerance and latency library. Hystrix mainly isolates the access points. It also makes sure that all 3rd Party libraries and services are restricted. Thus, we can use Hystrix to ensure that an application runs efficiently and avoids the kind of failures that occur in distributed systems.

Question: What is Spring Batch Framework?

Answer: Spring Batch is an open-source framework (<https://hackr.io/blog/top-10-web-development-frameworks-in-2020>) for batch processing – execution of a series of jobs. Spring Batch provides classes and APIs to read/write resources, transaction management, job processing statistics, job restart, and partitioning techniques to process high volume data.

Question: What is Tasklet, and what is a Chunk?

Answer: The Tasklet is a simple interface with one method to execute. A tasklet can be used to perform single tasks like running queries, deleting files, etc. In Spring Batch, the tasklet is an interface that can be used to perform unique tasks like clean or set up resources before or after any step execution.

Spring Batch uses a 'Chunk Oriented' processing style within its most common implementation. Chunk Oriented Processing refers to reading the data one at a time and creating chunks that will be written out, within a transaction boundary.

Question: How will you deploy Exception Handling in Microservices?

Answer: If an exception occurs while processing an HTTP request, you need to catch the exception in your controller or service and return an appropriate ResponseEntity manually. Here are some thumb rules for exception handling.

- Add @ResponseStatus for exceptions that you write.
- For all other exceptions, implement an @ExceptionHandler method on a @ControllerAdvice class or use an instance of SimpleMappingExceptionHandler.
- For Controller specific exceptions, add @ExceptionHandler methods to your controller.

Point to be noted is that @ExceptionHandler methods on the controller are always selected before those on any @ControllerAdvice instance. It is undefined in what order ControllerAdvices are processed.

Question: How can you access RESTful Microservices?

Answer: Considering the microservice architecture concept, each microservice needs to have an interface. Based on the principles of open networking of the Web, RESTful APIs provide the most logical model for building interfaces between the various components of the microservices architecture. RESTful APIs can be accessed in two ways:

- Using a REST Template that is load balanced.

- I am using multiple microservices.

Question: How do independent Microservices communicate with each other?

Answer: Microservices can communicate with each other through:

- HTTP for traditional Request-Response.
- Websockets for streaming.
- Brokers or Server Programs running Advanced Routing Algorithms.

For message brokers, RabbitMQ (<https://www.rabbitmq.com/>), Nats, Kafka, etc., can be used, each built for a particular message semantic. Another way is to use Backend As A Service like Space Cloud, which automates the entire backend.

Question: What is Semantic Monitoring?

Answer: Semantic Monitoring or Synthetic Monitoring is running a subset of the application's automated tests against the live production system. These results are monitored, and alerts are generated in case of failures. Semantic Monitoring approaches microservice monitoring from the business transaction perspective. Instead of monitoring each component, its semantic monitoring ascertains how well the transaction performs for business and the users. It also detects the faulty service layer and the corresponding microservice instance, all at the same flow. This approach allows for faster triaging and reduces the meantime to repair.

Question: How do you perform security testing of Microservices?

Answer: Microservices Application is a collection of smaller, independent, functional modules that may be developed in different programming languages, have different data sources, and run on different operating systems. This makes testing of the microservices as a whole a very tough task. The different pieces/modules need to be tested independently. There are three common procedures for this.

- Code Scanning: In order to ensure that every line of code is bug-free and can be replicated.
- Flexibility: The security protocols should be flexible as per the requirements of the system.
- Adaptability: The security protocols should be adaptable to the malicious intrusions.

Question: What do you understand by Idempotence, and how is it used?

Answer: Idempotence refers to the repeated performing of a task even though the end result remains the same. It is used mostly as a data source or a remote service in a way that when it receives the instruction more than once, it processes the instruction only once.

Question: What are the uses of Reports and Dashboards in Microservices environments?

Answer: Reports and Dashboards are generally used to monitor a system. For microservices Reports and Dashboards help to:

- Find which microservice support which resource.
- Find out the services that are impacted whenever changes in components are made/occur.
- Provide an easy point of access for documentation purposes.
- Review versions of the deployed components.
- Obtain compliance from the components.

Question: What are the tools that can be considered for managing a microservice architecture?

Answer: The main tools that can be used to build/manage a microservice architecture are:

1. **MongoDB:** It is a document-based open-source distributed database. Here data is stored in JSON format with a different structure for different documents. It also supports a lot of programming languages like C, C++, C#, PERL, PHP, Python, Java, Ruby, Scala, etc.
2. **Elasticsearch:** It is a full-text search engine.
3. **KAFKA:** It is an event queue system. All transactions are processed via the event queue, thus avoiding the web like random interactions between different services. Kafka renders a microservice architecture robust and clean.
4. **JENKINS:** It is an automation tool that enables Continuous Integration and Continuous Development. It supports many plugins and easily integrates with almost every tool.
5. **DOCKER:** The application development environment and application deployment environment are bound to vary in many aspects. This gives rise to deployment issues. Docker provides a static background for the application to run, thus avoiding deployment issues.
6. **KUBERNETES:** With thousands of services running in an application, Kubernetes, as an engine orchestrates the entire process.
7. **JAEGER:** It is an open-source end to end distributed tracing tool. Jaeger monitors distributed transactions, helps in performance optimization, and finds the dependencies between services. It also gives the root cause analysis.
8. **FLUENT:** In a multiservice architecture, where all the different systems are managed via different programming languages, different databases, and run in different operating systems, logging in, and keeping track of it is a significant issue. Fluentd provides a single logging layer and simplifies this issue. Logs can be also be collected and aggregated onto a data source.
9. **PROMETHEUS:** It is a monitoring tool, which helps to check if all services are working fine when the application is deployed. It is a time-series data store. It collects metrics from an application and displays it in a graphical format.
10. **grafana:** Grafana provides analytics and monitoring into different visualization formats like graphs, charts, tables, etc.
11. **NGINX:** It acts as a reverse proxy. It acts as a single point entry through which all the API calls are made.

Question: How do you create State Machines out of Microservices?

Answer: Each microservice owning its database is an independently deployable program. This enables the creation of State Machines by which we can specify different states and events for a particular microservice.

Question: What are the most common mistakes while transitioning to Microservices?

Answer:

- Failure to outline the main challenges.
- I am rewriting already existing programs.
- Vague definitions of responsibilities, timeline, and boundaries.
- Failure to identify and implement automation.

Question: Where is the WebMVC Test Annotation used?

Answer: WebMvcTest Annotation is used for unit testing Spring MVC Applications, where the test objective is to focus on Spring MVC Components.

Ex: @WebMvcTest(value = ToTestController.class, secure = false)

In the above example, the intention is to launch the ToTestController. All other controllers and mappings will not be launched when this unit test is executed.

Question: Give one/a few examples of microservices implementation.

Answer: Netflix, Paypal, Amazon, eBay, Twitter, and many other large-scale websites have started as monolithic and evolved into a microservices architecture.

Udemy offers a range of Microservices courses. The Microservices Interview Questions Preparation (<https://click.linksynergy.com/deeplink?id=jU79Zysihs4&mid=39197&murl=https://www.udemy.com/course/microservices-interview-questions/>), for instance, comprises a comprehensive set of Microservices Interview Questions.

If you are a paperback-lover then you may consider reading this book to learn more about Microservices interview questions: Top 50 Microservices Interview Questions & Answers: Good Collection of Questions Faced in Architect Level Technical Interviews (<https://geni.us/eiixjWg>).