**Top Microservices Interview Questions for 2019, Part 1**

**1. What Is Spring Cloud?**

 Spring Cloud, in microservices, is a system that provides integration with external systems. It is a short-lived framework that builds an application, fast. Being associated with the finite amount of data processing, it plays a very important role in microservice architectures.

For typical use cases, Spring Cloud provides the out of the box experiences and a sets of extensive features mentioned below:

* Versioned and distributed configuration.
* Discovery of service registration.
* Service to service calls.
* Routing.
* Circuit breakers and load balancing.
* Cluster state and leadership election.
* Global locks and distributed messaging.

**2. What Is Spring Boot?**

Spring boot is a major topic under the umbrella of microservices interview questions.  
With the new functionalities that have been added, Spring keeps getting more complex. Whenever you are starting a new project, it is mandatory to add a new build path or Maven dependencies. In short, you will need to do everything from scratch. Spring Boot is the solution that will help you to avoid all the code configurations.

**3. How Do You Override a Spring Boot Project’s Default Properties?**

This can be done by specifying the properties in the application.properties file.  
For example, in Spring MVC applications, you have to specify the suffix and prefix. This can be done by entering the properties mentioned below in the application.properties file.

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

**4. Role of Actuator in Spring Boot**

It is one of the most important features, which helps you to access the current state of an application that is running in a production environment. There are multiple metrics which can be used to check the current state. They also provide endpoints for RESTful web services which can be simply used to check the different metrics.

**5. How Is Spring Security Implemented In a Spring Boot Application?**

Minimal configuration is needed for implementation. All you need to do is add thespring-boot-starter-securitystarter in the pom.xml file. You will also need to create a Spring config class that will override the required method while extending the WebSecurityConfigurerAdapter to achieve security in the application. Here is some example code:

package com.gkatzioura.security.securityendpoints.config;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@Configuration

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/welcome").permitAll()

.anyRequest().authenticated()

.and()

.formLogin()

.permitAll()

.and()

.logout()

.permitAll();

}

}

**6. Which Embedded Containers Are Supported by Spring Boot?**

Whenever you are creating a Java application, deployment can be done via two methods:

* Using an application container that is external.
* Embedding the container inside your jar file.

Spring Boot contains Jetty, Tomcat, and Undertow servers, all of which are embedded.

* **Jetty** – Used in a wide number of projects, Eclipse Jetty can be embedded in framework, application servers, tools, and clusters.
* **Tomcat** – Apache Tomcat is an open source JavaServer Pages implementation which works well with embedded systems.
* **Undertow** – A flexible and prominent web server that uses small single handlers to develop a web server.

**7. What Do You Mean by End-To-End Testing of Microservices?**

[End-to-end testing](https://dzone.com/articles/all-you-need-to-know-about-end-to-end-testing) validates all the processes in the workflow to check if everything is working as expected. It also ensures that the system works in a unified manner, thereby satisfying the business requirement.

**8. What Is Semantic Monitoring?**

It combines monitoring of the entire application along with automated tests. The primary benefit of Semantic Monitoring is to find out the factors which are more profitable to your business.

Semantic monitoring along with service layer monitoring approaches monitoring of microservices from a business point of view. Once an issue is detected, they allow faster isolation and [bug triaging](https://dzone.com/articles/all-about-triaging-bugs), thereby reducing the main time required to repair. It triages the service layer and transaction layer to figure out the transactions affected by availability or poor performance.

**9. How Can You Set Up Service Discovery?**

There are multiple ways to set up service discovery. I’ll choose the one that I think to be most efficient, Eureka by Netflix. It is a hassle free procedure that does not weigh much on the application. Plus, it supports numerous types of web applications.

Eureka configuration involves two steps – client configuration and server configuration.  
Client configuration can be done easily by using the property files. In the clas spath, Eureka searches for a eureka-client.properties file. It also searches for overrides caused by the environment in property files which are environment specific.

For server configuration, you have to configure the client first. Once that is done, the server fires up a client which is used to find other servers. The Eureka server, by default, uses the Client configuration to find the peer server.

**10. Why Would You Opt for Microservices Architecture?**

This is a very common microservices interview question which you should be ready for! There are plenty of pros that are offered by a microservices architecture. Here are a few of them:

* Microservices can adapt easily to other frameworks or technologies.
* Failure of a single process does not affect the entire system.
* Provides support to big enterprises as well as small teams.
* Can be deployed independently and in relatively less time.

**11. Why Would You Need Reports and Dashboards in Microservices?**

Reports and dashboards are mainly used to monitor and upkeep microservices. There are multiple tools that help to serve this purpose. [Reports](https://www.lambdatest.com/blog/advanced-guide-on-writing-a-bug-report/) and dashboards can be used to:

* Find out which microservices expose what resources.
* Find out the services which are impacted whenever changes in a component occur.
* Provide an easy point which can be accessed whenever documentation is required.
* Versions of the components which are deployed.
* To obtain a sense of maturity and compliance from the components.

**12. Why Do People Hesitate to Use Microservices?**

I have seen many devs fumble over this question. After all, they're getting asked this question when interviewing for a microservices architect role, so acknowledging its cons can be a little tricky. Here are some good answers:

* **They require heavy investment** – Microservices demand a great deal of collaboration. Since your teams are working independently, they should be able to synchronize well at all times.
* **They need heavy architecture set up** – The system is distributed, the architecture is heavily involved.
* **They need excessive planning for handling operations overhead** – You need to be ready for operations overhead if you are planning to use a microservices architecture.
* **They have autonomous staff selection** – Skilled professionals are needed who can support microservices that are distributed heterogeneously.

**13. How Does PACT Work?**

PACT is an open source tool. It helps in testing the interactions between consumers and service providers. However, it is not included in the contract, increasing the reliability of the application. The consumer service developer starts by writing a test which defines a mode of interaction with the service provider. The test includes the provider’s state, the request body, and the response that is expected. Based on this, PACT creates a stub against which the test is executed. The output is stored in a JSON file.

**14. Define Domain Driven Design**

The main focus is on the core domain logic. Complex designs are detected based on the domain’s model. This involves regular collaboration with domain experts to resolve issues related to the domain and improve the model of the application. While answering this microservices interview question, you will also need to mention the core fundamentals of DDD. They are:

* DDD focuses mostly on domain logic and the domain itself.
* Complex designs are completely based on the domain’s model.
* To improve the design of the model and fix any emerging issues, DDD constantly works in collaboration with domain experts.

That's all for Part 1. [In Part 2](https://dzone.com/articles/top-microservices-interview-questions-for-2019-par), we'll cover 15 more great microservice interview questions to know!

## 15. What Are Coupling and Cohesion?

Coupling can be considered to be the measurement of strength between the dependencies of a component. A good microservices application design always consists of low coupling and high cohesion.

Interviewers will often ask about cohesion. It is also another measurement unit. More like a degree to which the elements inside a module remain bonded together.

It is imperative to keep in mind that an important key to designing microservices is a composition of low coupling along with high cohesion. When loosely coupled, a service knows very little about other services. This keeps the services intact. In high cohesion, it becomes possible to keep all the related logic in a service. Otherwise, the services will try to communicate with each other, impacting the overall performance.

## 16. What Is OAuth?

Open Authorization Protocol, otherwise known as OAuth, helps to access client applications using third-party protocols like Facebook, GitHub, etc., via HTTP. You can also share resources between different sites without the requirement of credentials.

OAuth allows the account information of the end user to be used by a third-party like Facebook while keeping it secure (without using or exposing the user’s password). It acts more like an intermediary on the user’s behalf while providing a token to the server for accessing the required information.

## 17. Why Do We Need Containers for Microservices?

To manage a microservice-based application, containers are the easiest alternative. It helps the user to individually deploy and develop. You can also use Docker to encapsulate microservices in the image of a container. Without any additional dependencies or effort, microservices can use these elements.

## 18. What Are the Ways to Access RESTful Microservices?

Another one of the frequently asked microservices interview questions is how to access RESTful microservices? You can do that via two methods:

* Using a REST template that is load balanced.
* Using multiple microservices.

## 19. What Are Some Major Roadblocks for Microservices Testing?

Talking about the cons, here is another one of the microservices interview questions you may be ready for, will be around the [challenges faced while testing microservices](https://www.lambdatest.com/blog/testing-challenges-related-to-microservice-architecture/).

* Testers should have a thorough understanding of all the inbound and outbound processes before they start writing the test cases for integration testing.
* When independent teams are working on different functionalities, collaboration can prove to be quite a struggling task. It can be tough to find an idle time-window to perform a complete round of [regression testing](https://www.lambdatest.com/blog/regression-testing-strategies-of-mobile-web-pages/).
* With an increasing number of microservices, the complexity of the system also increases.
* During the transition from monolithic architecture, testers must ensure that there is no disruption between the internal communication among the components.

## 20. Common Mistakes Made While Transitioning to Microservices

Not only on development, but mistakes also often occur on the process side. And any experienced interviewer will have this in the queue for microservices interview questions. Some of the common mistakes are:

* Often the developer fails to outline the current challenges.
* Rewriting the programs that are already existing.
* Responsibilities, timeline, and boundaries not clearly defined.
* Failing to implement and figure out the scope of automation from the very beginning.

## 21. What Are the Fundamentals of Microservices Design?

This is probably one of the most frequently asked microservices interview questions. Here is what you need to keep in mind while answering to it:

* Define a scope.
* Combine loose coupling with high cohesion.
* Create a unique service which will act as an identifying source, much like a unique key in a database table.
* Creating the correct API and taking special care during integration.
* Restrict access to data and limit it to the required level.
* Maintain a smooth flow between requests and response.
* Automate most processes to reduce time complexity.
* Keep the number of tables to a minimum level to reduce space complexity.
* Monitor the architecture constantly and fix any flaw when detected.
* Data stores should be separated for each microservice.
* For each microservice, there should be an isolated build.
* Deploy microservices into containers.
* Servers should be treated as stateless.

You can also follow this article explaining [9 Fundamentals to a Successful Microservice Design](https://dzone.com/articles/9-fundamentals-to-a-successful-microservice-design).

## 22. Where Do We Use WebMVC Test Annotation?

WebMvcTest is used for unit testing Spring MVC applications. As the name suggests, it focuses entirely on Spring MVC components. For example,  
@WebMvcTest(value = ToTestController.class, secure = false):  
Here, the objective is to only launch ToTestController. Until the unit test has been executed, other mappings and controllers will not be launched.

## 23. What Do You Mean by Bounded Context?

A central pattern is usually seen in domain driven design. Bounded context is the main focus of the strategic design section of DDD. It is all about dealing with large teams and models. DDD works with large models by disintegrating them into multiple bounded contexts. While it does that, it also explains the relationship between them explicitly.

## 24. What Are the Different Types of Two-Factor Authentication?

There are three types of credentials required for performing two-factor authentication.

1. A thing that you know – like password or pin or screen lock pattern.
2. A physical credential that you have – like OTP or phone or an ATM card, in other words, any kind of credential that you have in an external or third-party device.
3. Your physical identity – like voice authentication or biometric security, like a fingerprint or eye scanner.

## 25. What Is a Client Certificate?

This is a type of digital certificate usually used by client systems for making a request that is authenticated by a remote server. It plays an important role in authentication designs that are mutual and provides strong assurance of the identity of a requester. However, you should have a fully configured backend service for authenticating your client certificate.

## 26. What Is Conway's Law?

[Conway’s Law states](https://en.wikipedia.org/wiki/Conway%27s_law), “organizations which design systems … are constrained to produce designs which are copies of the communication structures of these organizations.”

The interviewer may ask a counter microservices interview question, like how is Conway's Law related to microservices. Well, some loosely coupled APIs form the architecture of microservices. The structure is well suited to how a small team is implementing components which are autonomous. This architecture makes an organization much more flexible in restructuring its work process.

## 27. How to Configure Spring Boot Application Logging?

Spring Boot comes with added support for Log4J2, Java Util Logging, and Logback. It is usually pre-configured as console output. They can be configured by only specifying logging.level in the application.properties file.

logging.level.spring.framework=Debug

## 28. How Would You Perform Security Testing on Microservices?

Before answering this microservices interview question, explain to the interviewer that microservices cannot be tested as a whole. You will need to test the pieces independently. There are three common procedures:

* **Code scanning** – To ensure that any line of code is bug-free and can be replicated.
* **Flexibility** – The security solution should be flexible so that it can be adjusted as per the requirements of the system.
* **Adaptability** – The security protocols should be flexible and updated to cope up with the new threats by hackers or security breaches.

You can also check out this article explaining the [influence of Microservices architecture on security](https://www.lambdatest.com/blog/does-microservices-architecture-influence-security-testing/).

## 29. What Is Idempotence and How Is it Used?

Idempotence refers to a scenario where you perform a task repetitively but the end result remains constant or similar.

Idempotence is mostly used as a data source or a remote service in a way that when it receives more than one set of instructions, it processes only one set of instructions.

**Q1) What is Spring Cloud in the realm of Microservices?**  
It is that kind of a feature in the realm of Microservices which would provide integration with the outside systems. It is also known as a short-lived Microservices framework that has the ability to build applications in a quick fashion. Moreover, it plays an important function in Microservices as it is associated with finite amounts of processing of data.

**Q2) Shed light on the architecture of Microservices**  
It is that kind of an architecture that facilitates the avoidance of huge application implementation for a large system. It is associated with the providence of loose coupling that takes place between various collaborating procedures. On the other side, it has the ability to run in an independent manner under various types of situations.

**Q3) Describe the way in which you can set up service discovery**  
Although a large number of ways are there in setting up service discovery, you have to use Eureka created by Netflix. This is usually a very hassle-free process that is not heavy on the application. Also, it can suit a variety of applications which is quite good.

**Q4) What do you mean by Eureka in the realm of Microservices?**  
Eureka is also known as the Netflix Service Discovery Server. This uses Spring Cloud and is most often known as the most used setup to start service discovery. It is also not that heavy on the application development process. This is the reason that it is quite popular among the developers of today.

**Q5) Shed light on the ways by which you can access a RESTful Microservices**  
These are the following ways with the help of which you can use a RESTful Microservice.

* With the use of the load balanced rest template
* With the use of multiple Microservices, you can easily use a RESTful template
* If you have been given a large number of RESTful templates, then always make sure that you use the right one

**Q6) Describe the process by which you can balance the server-side load by utilizing Spring Cloud**  
It is interesting to note that the balancing act in case of achieving server-side load can be achieved by the utilization of Netflix Zuul. The Zuul is also known as a JVM based router. It is also regarded as a load balancer by Netflix. This is the reason that it always facilitates a single entity to the system.

**Q7) Can you integrate Zuul with other types of projects?**  
Yes, Zuul can be integrated with other types of Netflix services that are known as Hystrix. It is especially meant for tolerance of various types of faults that are commonly present in Eureka. By tolerating various kinds of faults, service discovery can be made easier within the realm of Microservices. One can also use it to manage routing tables and effective balancing of the load across the system.

**Q8) Shed light on the basic need of Microservices in today’s context of application development**  
Microservices are also known as a new pattern in the realm of software development. It has assumed importance owing to the fact that it has the ability to increase speed and efficiency. It can also manage software solutions in an effective manner. Some also call it as an approach to process and culture which delivers business value in an increased manner. In today’s context of application development, it plays a very crucial role as it can be deployed in the development process of a monolithic application in an efficient manner.

**Q9) Describe the circumstances under which you would use the Netflix Hystrix**  
It is a commonly accepted fact that Hystrix is also known as an error tolerance and latency library. The main purpose of Hystrix is to make sure that it isolates the access points. On the other hand, with the help of these access points, the remote systems can be easily reached. It also makes sure that it restricts the widespread use of 3rd party libraries as well as services. In this manner, it ensures that an application runs in an efficient manner. It is also quite effective in prohibiting the failure that frequently takes place in distributed systems that are quite complex in nature.

**Q10) Define the process by which you can easily deploy Spring Batch Tasks with reference to Microservices**  
The task related to Spring Batch is also known as a simple interface that has just one method to execute. With the help of this feature, you can easily perform singular tasks that relate to deleting and file queries. An example of Spring Batch in the Microservices segment can be in the form of Hello World.

**Q11) Shed light on Tasklet with reference to Spring Batch and Microservices**  
It is important to note that Spring Batch easily provides a Tasklet interface with the help of which the application can perform a single task. It can also clean and delete the various types of resources that are needed before the final execution step.

**Q12) Define the Spring Batch Framework with respect to Microservices**  
It is a batch framework that is quite comprehensive and is light in weight on the application. Its lightweight nature is a great source of positivity in the realm of Microservices. This is so because it is designed in such a manner that it can facilitate the development of robust batch applications. It is that kind of a feature in the realm of Microservices that can easily build upon productivity. It is especially used in the development of enterprise applications that are necessary to make sure that the applications developed are able to meet the set standards of a particular organization.

**Q13) Shed light on the way by which you can disable the endpoint of Actuator in the realm of Spring Boot**  
At the time of answering this question, you should be well aware of the fact that all the insightful HTTP endpoints are protected. Hence, the developer with the role of an actuator can only access them. On the other hand, to make this answer more credible, you should also add that security can be enforced by the utilization of a user-specific request. This request can be made to the batch frameworks also so that the application runs in a smooth manner.

**Q14) What do you know about YAML in the realm of Microservices?**  
You should always note that YAML is also known as an individual legible language. Many experts also call it a language that can result in the sterilization of data. Hence, basically, it denotes that it can be used to cleanse data. This has a great advantage on the Microservices network. As compared to the other files of various properties, the YAML file is said to be more organized in manner. On the other hand, it is also less confusing and hence it provides easy accessibility to a variety of web developers. However, it is also vital to note that YAML possesses the hierarchical form of data that has played an essential role in the swift development of various applications.

**Q15) Shed light on the various aspects of using Spring Profiles**  
In the context of Microservices, the Spring Profiles play an essential role. It is so because it allows the users in the process of registering various beans. However, the registration process of beans is always dependent on various ways by which the application has been executed. Hence, if you are running the application in the DEVELOPMENT mode, you would witness that a certain number of items can be encumbered in an easy manner. On the other hand, during the time of production, the other items can also be loaded. With the widespread use of Spring Boot, it has become relatively quite easy to make sure that all the profiles can be easily booted. This makes sure that the developed application is running free of errors and is light on the interface.

**Q16) Define caching with respect to the environment in which Microservices operate**  
 It is a commonly observed fact that the cache is that kind of an area in the local memory that has the ability to hold the copy of frequently researched data. In other words, if you have cache accumulated in the application, the application speeds up in a phenomenal manner. On the other side, you can also use the cast function to control the amount of cache you want.

**Q17) Do you frequently use the framework related to integration with that of the Spring Boot?**  
While answering this question, you can always say that you have used [Apache Camel](https://mindmajix.com/apache-camel-training) and have experience in integrating with the Spring Boot function. You can also say that you have used the Apache Camel Boot starter in the environment of Microservices.

**Q18) Shed light on AOP and also describe the way in which you can use it with Microservices**  
At the time of answering this question, you should always focus on the concerns of cross-cutting. The concerns of cross-cutting are the ways that extent numerous roads of a particular application. In this context, it is interesting to note that the concerns of cross-cutting usually fluctuate from the business logic of a particular application. This is the reason that the separation of the concerns related to the cross-cutting nature is always a primary step that has to be taken in the application development process. AOP is also known as characteristic oriented programming that goes a long way in making sure that business logic is applied correctly to enterprise based applications.

**Q19) Define the process with the help of which you can deploy exception handling in the realm of Microservices**  
It is important to note that Microservices and spring usually provides a unique manner in which you can control the Controller Advice. You can say that you have the experience of deploying a class of Controller Advice with the help of the exceptions that are being used by the class of controllers.

**Q20) Describe the advantages of Microservices nowadays**  
There are various advantages of using Microservices. They are as follows:

* The tiny base of code is quite easy to maintain
* It is quite easy to scale as it has only an individual component
* It also supports the phenomenon of independent implementation
* It drastically reduces the installation time
* It renders good support for the parallel and the support teams
* Another main advantage of this Microservices is in the form of the fact that it has the ability to facilitate technological diversity

**Q21) Shed light on the various types of IT characteristics that are available in Microservices**  
With the help of Microservices, your business can scale new heights as it integrates IT planning and execution of projects. Here is the list of important IT characteristics those are available in Microservices.

* Planning of various types of IT projects
* Management of facilities
* Implementation of several types of [Agile Practices](https://mindmajix.com/agile-project-management)
* The allocation of shared services
* Management of the business cases
* Maintaining transparency is various aspects of cost along with the planning of IT management

**Q22)  What are the uses of reports and dashboards in the environment of Microservices?**  
It is vital for you to note that Microservices comprises of a plethora of publishing features. This includes a variety of charts, PDFs, and dashboards. With the help of dashboards and reports in Microservices, you can easily analyze scenarios and facilitate various types of executive packs.

**1) Explain microservices architecture**

Microservice Architecture is an architectural development style which builds an application as a collection of small autonomous services developed for a business domain.

**2) Name three commonly used tools for Microservices**

* Wiremock, 2.) Docker and 3.) Hysrix are important Microservices tool.

**3) What is Monolithic Architecture?**

Monolithic architecture is like a big container in which all the software components of an application are clubbed inside a single package.

**4) What are the advantages of microservices?**

Here, are some significant advantages of using Microservices:

* Technology diversity, e., Microservices can mix easily with other frameworks, libraries,  and databases
* Fault isolation, e., a process failure should not bring the whole system down.
* Greater support for smaller and parallel team
* Independent deployment
* Deployment time reduce

**5) What is Spring Cloud?**

Spring cloud is an Integration software that integrates with external systems. It allows microservices framework to build applications which perform restricted amounts of data processing.[](https://career.guru99.com/wp-content/uploads/2018/06/process.png)

**6) Discuss uses of reports and dashboards in the environment of Microservices**

Reports and dashboards help in monitoring and upkeep of Microservices. Tons of Application Monitoring Tools assist in this.

**7) What are main differences between Microservices and Monolithic Architecture?**

|  |  |
| --- | --- |
| **Microservices** | **Monolithic Architecture** |
| Service Startup is fast | Service startup takes time |
| Microservices are loosely coupled architecture. | Monolithic architecture is mostly tightly coupled. |
| Changes done in a single data model does not affect other Microservices. | Any changes in the data model affect the entire database |
| Microservices  focuses  on products, not projects | Monolithic put emphasize over the whole project |

**8) What are the challenges faced while using Microservices?**

* Microservices always rely on each other. Therefore, they need to communicate with each other.
* As it is distributed system, it is a heavily involved model.
* If you are using Microservice architecture, you need to ready for operations overhead.
* You need skilled professionals to support heterogeneously distributed microservices.

**9) In which cases microservice architecture best suited?**

Microservice architecture is best suited for desktop, web, mobile devices, Smart TVs, Wearable, etc.

**10) Tell me the name of some famous companies which are using Microservice architecture**

Most large-scale websites like Twitter, Netflix, Amazon, have advanced from a monolithic architecture to a microservices architecture.

**11) What are the characteristics of Microservices?**

* Essential messaging frameworks
* Decentralized Governance
* Easy Infrastructure automation
* Design for failure
* Infrastructure automation

**12) What is RESTful?**

Representational State Transfer (REST)/RESTful web services is an architectural style that helps computer systems to communicate over the internet. These web services make microservices easier to understand and implement.

**13) Explain three types of Tests for Microservices?**

In Microservice architecture tests are divided into three broad categories:

* At the bottom level test, we can perform a general test like performance and unit tests. These kinds of tests are entirely automated.
* At the middle level, we can perform exploratory tests like the stress tests and usability tests.
* At the top level, we can conduct acceptance tests which are mostly fewer in numbers. It also helps stakeholders to know about different software features.

**14) What are Client certificates?**

Client certificates is a digital certificate used to make authenticated requests to a remote server. It is termed as a client certificate.

**15) Explain the use of PACT in Microservices architecture?**

It is an open source tool which allows testing interactions between service providers and consumers. However, it is separated from the contract made. This increases the reliability of the Microservices applications.

**16) What is the meaning of OAuth?**

OAuth means open authorization protocol. This protocol allows you to access the client applications on HTTP for third-party providers GitHub, Facebook, etc. It helps you to share resources stored on one site with another site without the need for their credentials.

**17) What is End to End Microservices Testing?**

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

**18) Why are Container used in Microservices?**

Containers are easiest and effective method to manage the microservice based application. It also helps you to develop and deploy individually. Docker also allows you to encapsulate your microservice in a container image along with its dependencies. Microservice can use these elements without additional efforts.

**19) What is the meaning of Semantic monitoring in Microservices architecture?**

Semantic monitoring combines automated tests with monitoring of the application. It allows you to find out reasons why your business is not getting more profits.

**20) What is a CDC?**

CDC is Consumer-Driven Contract. It is a pattern for developing Microservices so that external systems can use them.

**21) What is the use of Docker?**

Docker offers a container environment which can be used to host any application. This software application and the dependencies that support it which are tightly-packaged together.

**22) What are Reactive Extensions in Microservices?**

Reactive Extensions is also called Rx. It is a design pattern which allows collecting results by calling multiple services and then compile a combined response. Rx is a popular tool in distributed systems which works exactly opposite to legacy flows.

**23) Explain the term ‘Continuous Monitoring.’**

Continuous monitoring is a method which is used for searching compliance and risk issues associated with a company’s operational and financial environment. It contains human, processes, and working systems which support efficient and actual operations.

**24) How independent micro-services communicate with each other?**

It depends upon your project needs. However, in most cases, developers use HTTP/REST with JSON or Binary protocol. However, they can use any communication protocol.