1. What are microservices?

Answer: Microservice Architecture is an architectural style that structures an application

as a collection of small autonomous services, modeled around a business domain.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. What is monolithic architecture?

Answer: Monolithic Architecture is similar to a big container wherein all the software

components of an application are assembled together and tightly packaged.

Monolithic architecture is a traditional way of making software, in which all of an application’s

parts and functions are tightly tied together and packaged as a single unit.

Key characteristics of a Monolithic Architecture:

Single Codebase:

The entire application is developed and maintained within a single code repository.

Tight Coupling :

Components and modules within the application are tightly interconnected, making it difficult

to modify or replace individual parts without impacting the entire system.

Scalability Challenges :

Scaling a monolithic application can be challenging as all components need to be scaled together,

even if only a specific part requires more resources.

Monolithic Deployment :

The application is deployed as a whole, which can lead to longer deployment cycles and increased

downtime during updates.

Technology Stack :

Monolithic applications often use a uniform technology stack for the entire system.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3. What are the characteristics of microservices?

Answer: The characteristics of microservices include:

Modularity:

\* Microservices are small, independent components that focus on specific business functions.

Independence:

\* Each microservice operates independently, enabling separate development, deployment, and scaling.

Decentralized Data Management:

\* Microservices often manage their own databases, enhancing autonomy.

Autonomous Development and Deployment:

\* Microservices can be developed, deployed, and scaled independently.

Resilience:

\* Failures in one microservice do not affect the entire system due to fault isolation.

Technology Diversity:

\* Different microservices can use different technologies and programming languages.

Scalability:

\* Microservices can be individually scaled based on demand.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4. How does microservice architecture work?

Answer:

Clients – Different users from various devices send requests.

Identity Providers – Authenticates user or clients identities and issues security tokens.

API Gateway – Handles client requests.

Static Content – Houses all the content of the system.

Management – Balances services on nodes and identifies failures.

Service Discovery – A guide to find the route of communication between microservices.

Content Delivery Networks – Distributed network of proxy servers and their data centers.

Remote Service – Enables the remote access information that resides on a network of IT devices.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5. Define microservices architecture

Answer: Microservices architecture is an approach to building software applications as a collection

of small, independent services. These services, or microservices, work together through well-defined

APIs, enabling flexibility, scalability, and independent deployment of each component.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6. Explain Container In Microservices

Answer: Containers are a good way to manage microservice based application to develop and

deploy them individually. You can encapsulate your microservice in a container image along

with its dependencies, which then can be used to roll on-demand instances of microservice

without any additional efforts required.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7. What is the difference between Monolithic, SOA and Microservices Architecture?

Answer:

\* Monolithic - Monolithic Architecture is similar to a big container wherein all the software components of an

application are assembled together and tightly packaged.

\* SOA - A Service-Oriented Architecture is a collection of services which communicate with each other.

The communication can involve either simple data passing or it could involve two or more services

coordinating some activity.

\* Microservices- Microservice Architecture is an architectural style that structures an application

as a collection of small autonomous services, modeled around a business domain.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8. What is Spring Cloud?

Answer: Spring Cloud is a system in Microservices that integrates external systems. It builds

an application in a fast manner as it is a short-lived framework. It plays a vital role in the

architecture of Microservices as it is associated with a finite amount of data processing.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9. What are the three types of tests for Microservices?

Answer: The first test is the bottom level test that performs general test like unit and performance

tests. At the middle level, experimental tests like usability and stress tests are there. Moreover,

at the top level, acceptance testing takes place. Tests at all levels are automated.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

Answer: Netflix, Uber, Amazon, Twitter, Spotify, Google, LinkedIn

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11. What is OAuth?

Answer: OAuth stands for open authorization protocol. This allows accessing the resources

of the resource owner by enabling the client applications on HTTP services such as third-party

providers Facebook, GitHub, etc. So with this, you can share resources stored on one site with

another site without using their credentials.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12. What is End to End Microservices Testing?

Answer: End-to-end testing validates each and every process in the workflow is functioning properly.

This ensures that the system works together as a whole and satisfies all requirements.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13. What is DRY in Microservices architecture?

Answer: DRY stands for Don’t Repeat Yourself. It basically promotes the concept of reusing the

code. This results in developing and sharing the libraries which in turn result in tight coupling.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14. What is Coupling?

Answer: The measure of the strength of the dependencies between components is said to be

coupling. A good design is always said to have High Cohesion and Low Coupling.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15. Name three commonly used tools for Microservices?

Answer: The three commonly used tools for Microservices are:

-- Docker:

\* Docker is a tool that lets developers set up apps in lightweight, portable containers in

an automated way. It helps put the microservices and their dependencies into containers.

This makes sure that the microservices are the same in all environments and makes it

easier to launch and grow them.

--Kubernetes:

\* Kubernetes is a powerful tool for managing, scaling, and automating the launch of Docker

containers. It does this by orchestrating how containers work together. It has important tools

for running microservices in a live setting, such as load balancing, service discovery, self-healing,

and automatic scaling.

-- Spring Boot:

\* Spring Boot is a famous platform built on Java that makes it easier to build and launch microservices.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

16. What are the pros and cons of Microservice Architecture?

Answer:

-- Pros of Microservice Architecture :

\* Freedom to use different technologies

\* Each microservices focuses on single capability

\* Supports individual deployable units

\* Allow frequent software releases

-- Cons of Microservice Architecture

\* Increases troubleshooting challenges

\* Increases delay due to remote calls

\* Increased efforts for configuration and other operations

\* Difficult to maintain transaction safety