resources stored on one site with another site without their credentials.

IGNORE ANY ADVERTISMENT AND READ ABOVE AND PREPARE ME THESE ANSWERS IN SIMPLE AND EASY TO REMEMBER LANGUAGE ALSO GIVIN SUFFICIENT EXPLANATION

Here are simplified and easy-to-remember answers based on the points you provided:

---

1) What are Microservices?

Microservices is an architectural style used to build applications as a collection of small, independent services. Each service performs a specific function, can be deployed independently, and communicates with other services through lightweight mechanisms like APIs.

Key Features:

- Loosely coupled
- Easy to maintain and test
- Scalable and flexible
- Each service is owned by a small team

---

2) Benefits of Microservices

- **Faster development:** Teams can build and deploy services quickly.
- **Independent updates:** Changes in one service don't break others.
- **Scalability:** Each service can be scaled independently.
- **Technology flexibility:** Teams can use different tech stacks for different services.

---

3) Three Common Tools for Microservices

- **WireMock:** Used for testing APIs by simulating responses.
- **Docker:** Helps containerize services to run them independently.
- **Hystrix:** Handles failures in microservices to prevent cascading issues.

---

4) Main Components of Microservices

- **Containers & Orchestration:** Docker, Kubernetes to manage microservices.
- **API Gateway:** Manages communication between services.
- **Infrastructure as Code (IaC):** Automates infrastructure setup.
- **Cloud Infrastructure:** Platforms like AWS, Azure for hosting services.

---

5) How Microservices Architecture Works

Microservices architecture breaks an application into small, independent services. Each service does one specific task and can be updated or replaced without affecting others. This architecture scales well and supports frequent updates.

---

6) Advantages of Microservices

- **Technology diversity:** Use different technologies for different services.
- **Fault isolation:** A failure in one service won't crash the entire app.
- **Faster deployment:** Services can be developed and deployed quickly.
- **Independent teams:** Small teams can focus on specific services.

---

7) What is Monolithic Architecture?

Monolithic architecture is a single, large application where all components are tightly connected. This structure makes it harder to scale and update, as changes in one area affect the entire system.

---

8) Biggest Challenges in Microservice Deployment

- **Business challenges:** High initial investment and infrastructure setup.
- **Technical challenges:** Complex communication between services, difficult debugging, and the need for skilled professionals.

---

9) What is Spring Cloud?

Spring Cloud is a framework for building microservices in Java. It helps microservices communicate with external systems and manage configuration, load balancing, and more.

---

10) Microservices vs. Monolithic Architecture

Microservices	Monolithic Architecture
Loosely coupled, independent services	Tightly coupled components
Fast startup	Slow startup
Easy to scale and update parts	Hard to update, as changes affect all
Focuses on business capabilities	Focuses on the whole project

---

11) When is Microservice Architecture Best Suited?

Microservices are ideal for building applications for devices like desktops, mobiles, smart TVs, wearables, etc., where scalability and flexibility are important.

---

12) Advantages and Disadvantages of Microservices

Advantages:

- Improved scalability
- Increased agility for quick updates

- Easy debugging and fault isolation
- Independent deployment by small teams

Disadvantages:

- Complex testing and deployment
- Hard to manage inter-service dependencies
- Requires skilled professionals

---

13) Companies Using Microservices

Big companies like **Netflix**, **Amazon**, and **Twitter** use microservices to handle high traffic and constantly changing requirements.

---

14) What is RESTful?

RESTful is an architectural style for designing APIs. It allows systems to communicate over the web using standard HTTP methods like GET, POST, PUT, DELETE.

---

15) Microservices Deployment Strategies

- **Multiple Service Instances per Host:** Run multiple instances of services on one or more hosts.
- **Service Instance per Container:** Each service runs in its own container.
- **Serverless Deployment:** Services run on-demand in serverless environments like AWS Lambda.

---

16) Types of Tests in Microservices

- **Bottom-Level Tests:** Unit tests and performance tests.
- **Middle-Level Tests:** Stress tests and usability tests.
- **Top-Level Tests:** Acceptance tests for verifying app features.

---

17) SOA vs Microservices

- **SOA (Service-Oriented Architecture):** Services communicate through a centralized middleware (like ESB).
- **Microservices:** Each service is independent, smaller, and communicates directly over lightweight protocols like HTTP.

---

18) What is a Client Certificate?

A client certificate is a digital certificate used by microservices to authenticate themselves when communicating with other services securely.

---

19) What is Domain-Driven Design (DDD)?

Domain-Driven Design focuses on building software by modeling it closely around the business domain. It emphasizes collaboration with domain experts to create a flexible, evolving system.

20) What is PACT?

PACT is an open-source tool for testing the interactions between services (consumers and providers) in a microservice architecture. It helps ensure services work together correctly.

---

21) What is OAuth?

OAuth is an open authorization protocol that allows applications to access resources on behalf of users without sharing their passwords. It's used in services like Facebook and GitHub for login and resource access.

These answers are simplified to help you remember the key points for interviews or explanations.

22) 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.

23) 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. This can be done by:

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

24) 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.

25) What is the difference between Coupling and Cohesion?
Difference between Coupling and Cohesion

Coupling: Coupling is specified as a relationship between module A and another module B. There are mainly three types of coupling. Any module can be highly coupled (highly dependent), loosely coupled, and uncoupled with other modules. The best coupling is the loose coupling which can be achieved through interfaces.

Cohesion: Cohesion is the relationship between 2 or more parts within a module. The high cohesion within a module specifies that the module can perform a specific task with maximum

IGNORE ANY ADVERTISMENT AND READ ABOVE AND PREPARE ME THESE ANSWERS IN SIMPLE AND EASY TO REMEMBER LANGUAGE ALSO GIVIN SUFFICIENT EXPLANATION

Here are simplified answers to your questions, explained in easy-to-remember language:

---

22) What is Spring Boot? Why is it used?

Spring Boot is a framework built on top of Spring that simplifies building applications. It provides default configurations so you can quickly set up and run your app without a lot of manual setup. It's great for reducing development time and creating stand-alone, production-ready applications.

---

23) How to override default properties in Spring Boot?

You can override default properties in the `**application.properties**` file. For example:

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

This helps you customize file locations or other configurations.

---

24) What is end-to-end Microservices testing?

End-to-end testing checks that all parts of your system (microservices) work well together. It ensures that the entire workflow functions correctly and meets the system's requirements.

---

25) Difference between Coupling and Cohesion?

- **Coupling:** How dependent one module is on another. Loose coupling is better because modules can function independently.
- **Cohesion:** How well parts within a module work together. High cohesion means the module is more focused and efficient.

---

26) What is the use of containers in Microservices?

Containers help manage microservices by packaging them with all their dependencies. Tools like Docker allow developers to build, deploy, and scale microservices independently.

---

27) What is Spring Cloud?

Spring Cloud provides tools for managing distributed systems (e.g., microservices). It helps with problems like load balancing, service discovery, configuration management, and circuit breaking.

---

28) What is semantic monitoring in Microservices?

Semantic monitoring combines testing and monitoring to ensure business processes are functioning properly. It focuses on the reasons why a business might not be achieving its goals.

---

29) What is a Distributed Transaction?

A distributed transaction involves multiple services or databases. A transaction manager ensures that all parts either succeed or fail together, maintaining consistency.

---

30) What is CDC (Consumer-Driven Contract)?

CDC is a pattern where consumers (clients) define how they expect the services they depend on to behave. It ensures that microservices can work together smoothly.

---

31) What is Reactive Extension in Microservices?

Reactive Extension (Rx) is a pattern that lets you gather responses from multiple services and combine them into one. It helps in building efficient, non-blocking systems.

---

32) How to configure logging in a Spring Boot application?

You can set logging levels in `**application.properties**` like this:

```properties
  logging.level.root=WARN
```

This helps control how much information is logged.

---

33) What is Continuous Monitoring?

Continuous monitoring checks the health, security, and performance of systems continuously. It helps detect issues in real time, ensuring a company's systems are always compliant and running smoothly.

---

34) What is Mike Cohn's Test Pyramid?

It's a strategy for efficient testing:

- **Unit tests:** Fast and isolated, form the base of the pyramid.
- **Service tests:** Test business logic.
- **UI tests:** Slow and cover end-to-end functionality, at the top of the pyramid.

---

35) How do microservices communicate with each other?

Microservices typically communicate using HTTP/REST with JSON, or sometimes a more efficient protocol like gRPC. The communication method depends on the project's needs.

---

36) How to implement Spring Security in a Spring Boot app?

1. Add `spring-boot-starter-security` to your `pom.xml`.
2. Create a class that extends `WebSecurityConfigurerAdapter` and override necessary methods to customize security.

---

37) What is Ubiquitous Language?

It's a common language used by both developers and business users to describe the domain. It ensures everyone is on the same page and helps translate domain language into code.

---

38) Difference between REST and Microservices?

- **REST:** A way to design APIs over HTTP, commonly used in microservices.
- **Microservices:** A software architecture pattern where each service is independent, deployable, and scalable on its own. REST is just one way to implement communication between microservices.

---

39) What is Idempotence?

Idempotence means doing something multiple times has the same result as doing it once. It ensures reliability when dealing with remote services or data sources.

---

40) What is Actuator in Spring Boot?

Spring Boot Actuator provides endpoints to monitor and manage your application's health, metrics, and more. It gives production-ready features to expose operational information via HTTP or JMX.

---

41) What is Bounded Context in Domain-Driven Design?

Bounded Context divides a large model into smaller contexts where each has its own domain and rules. This makes it easier to manage and maintain large applications.

---

42) What is PACT in Microservices Architecture?

PACT is a contract between a microservice provider and a consumer. It ensures the two can communicate correctly and efficiently. PACT testing verifies this contract.

---

43) What is Two-Factor Authentication (2FA)?

2FA adds an extra layer of security by requiring two forms of verification:

1. **Something you know** (password or PIN).
2. **Something you have** (phone, OTP).

3. **Something you are** (fingerprint).

---

44) Why are Reports and Dashboards important in Microservices?

They help monitor the health, performance, and compliance of microservices, track the versions deployed, and provide insight into how different services interact.

---

45) What is Canary Releasing?

Canary releasing means deploying new features or code to a small set of users first. If everything goes well, it's then released to all users. It's a safe way to release updates.

---

46) Why do developers hesitate to use Microservices?

- **Complexity:** Microservices require handling distributed systems and communication between services.
- **Higher costs:** More infrastructure and skilled professionals are needed.
- **Operations overhead:** More effort is required to manage and monitor many services.

---

47) What are non-deterministic tests?

Non-deterministic tests are tests that sometimes pass and sometimes fail without consistent reasons. To eliminate them, we can quarantine flaky tests, fix timing issues, and isolate external dependencies.

---

48) What is WebMvcTest in Spring MVC?

`` `@WebMvcTest` `` is used to test Spring MVC components like controllers. It launches only the controller you want to test, without starting the entire application context.

---

49) What is Eureka in Microservices?

Eureka is a service registry used in microservices architecture. Services register themselves with Eureka, and it helps them discover each other for communication.

---

50) What is DRY in Microservices?

DRY stands for **Don't Repeat Yourself**. It encourages reusing code to reduce duplication and improve maintainability.

---

51) How to balance server-side load using Spring Cloud?

You can use **Netflix Zuul**, a JVM-based router, to balance the server load by routing requests efficiently across services.

---

52) What is Cross-functional Testing?

It tests non-functional aspects like performance, security, and usability, ensuring the system meets all requirements beyond functionality.

---

53) What is Netflix Hystrix?

Hystrix helps manage failures in a distributed system by isolating services and preventing cascading failures. It ensures the application continues to function even when some services fail.

---

54) What is a Tasklet in Microservices?

A Tasklet is a simple interface in Spring Batch used for executing single tasks, like cleaning up resources before or after batch processing steps.

These explanations should help you prepare in a simpler, more digestible way!