Multi Choice Type Question What is the main characteristic of microservices architecture? Marks : 1 Negative Marks : 0 image Answer here Monolithic structure Decentralized and independent services Centralized database High coupling between components

The correct answer is:

**Decentralized and independent services** ✅

Microservices architecture is characterized by breaking an application into small, independent services that can be developed, deployed, and scaled independently, unlike a monolithic structure which is centralized and tightly coupled.

**You said:**

What technology is commonly used for communication between microservices? Marks : 1 Negative Marks : 0 image Answer here HTTP/HTTPS TCP/IP FTP UDP

The correct answer is:

**HTTP/HTTPS** ✅

Microservices commonly communicate over network protocols like HTTP/HTTPS using REST APIs or gRPC. While TCP/IP and UDP are lower-level protocols, HTTP/HTTPS is the standard for service-to-service communication in microservices.

**You said:**

Which logging level is typically used for messages that provide detailed information for debugging purposes? Marks : 1 Negative Marks : 0 image Answer here Error Debug Warning Info

The correct answer is:

**Debug** ✅

The **Debug** logging level is used to provide detailed information useful for diagnosing problems and tracing the application's execution during development.

**You said:**

Multi Choice Type Question Which of the following is a potential benefit of microservices? Marks : 1 Negative Marks : 0 image Answer here Increased development team size Tight coupling between services Improved scalability and deployment flexibility Shared database among all services

The correct answer is:

**Improved scalability and deployment flexibility** ✅

Microservices allow each service to be developed, deployed, and scaled independently, which improves flexibility and scalability. Tight coupling and shared databases are characteristics of monolithic architecture, not microservices.

**You said:**

Multi Choice Type Question Which built-in logging provider is available in ASP.NET Core by default? Marks : 1 Negative Marks : 0 image Answer here NLog Log4net Serilog Microsoft.Extensions.Logging

The correct answer is:

**Microsoft.Extensions.Logging** ✅

ASP.NET Core includes **Microsoft.Extensions.Logging** as the built-in logging provider by default. Third-party providers like NLog, Log4net, and Serilog can be integrated but are not included out of the box.

**You said:**

Multi Choice Type Question What is the purpose of an API endpoint URL? Marks : 1 Negative Marks : 0 image Answer here To specify the data format (e.g., JSON or XML) To handle business logic To identify the location of a specific resource To define the request headers

The correct answer is:

**To identify the location of a specific resource** ✅

An **API endpoint URL** specifies the address where a client can access a particular resource or service in an API. It does not define the data format, request headers, or handle business logic directly.

**You said:**

Multi Choice Type Question What is the main purpose of the ConfigureServices method in an ASP.NET Core startup class? Marks : 1 Negative Marks : 0 image Answer here To configure HTTP request pipeline To define routing for API endpoints To register application services with the built-in dependency injection container To handle database migrations

The correct answer is:

**To register application services with the built-in dependency injection container** ✅

In ASP.NET Core, the **ConfigureServices** method is used to add and configure services (like DbContext, logging, or custom services) so they can be injected throughout the application. Configuring the HTTP pipeline is done in **Configure**, not **ConfigureServices**.

**You said:**

Multi Choice Type Question Which design pattern is commonly used for service discovery in microservices architectures? Marks : 1 Negative Marks : 0 image Answer here Singleton Pattern Observer Pattern Registry Pattern Circuit Breaker Pattern

The correct answer is:

**Registry Pattern** ✅

In microservices architectures, the **Registry Pattern** (often implemented via a service registry like Eureka or Consul) is used for **service discovery**, allowing services to dynamically find and communicate with each other.

* **Singleton** is for single instance objects.
* **Observer** is for event notification.
* **Circuit Breaker** is for fault tolerance, not discovery.

**You said:**

Multi Choice Type Question In microservices architecture, what is the role of an API gateway? Marks : 1 Negative Marks : 0 image Answer here To authenticate users To centralize logging To manage API traffic and routing To store service configurations

The correct answer is:

**To manage API traffic and routing** ✅

An **API gateway** acts as a single entry point for clients, handling **routing, request aggregation, load balancing, and sometimes authentication**, making communication with multiple microservices simpler and more efficient.

**You said:**

Multi Choice Type Question What is the primary reason for using asynchronous communication between microservices? Marks : 1 Negative Marks : 0 image Answer here Increased latency Reduced scalability Improved responsiveness and performance Simplified debugging process

The correct answer is:

**Improved responsiveness and performance** ✅

Asynchronous communication allows microservices to **send requests without waiting for immediate responses**, which improves system responsiveness, throughput, and overall performance. It helps decouple services and handle high loads efficiently.

**You said:**

Multi Choice Type Question In a RESTful API, what HTTP method is typically used to update an existing resource? Marks : 1 Negative Marks : 0 image Answer here GET POST PUT DELETE

The correct answer is:

**PUT** ✅

In RESTful APIs, **PUT** is typically used to **update an existing resource** (or create it if it doesn’t exist, in some implementations).

* **GET** retrieves data.
* **POST** creates a new resource.
* **DELETE** removes a resource.