126. What is resilience in microservices?

Resilience is the ability of a system to recover from failures and continue functioning, possibly in a degraded mode.

127. What are common resilience patterns?

- Retry
- Circuit Breaker
- Timeout
- Bulkhead
- Rate Limiting
- Fallback

128. What is the Retry pattern?

It automatically retries a failed operation a configured number of times before giving up.

129. When should you use the Retry pattern?

Use it for transient failures like network timeouts or temporary service unavailability. Avoid for non-idempotent operations.

130. What is a Circuit Breaker?

A mechanism that stops calls to a failing service and allows it to recover. It has Closed, Open, and Half-open states.

131. Benefits of using Circuit Breaker

- Prevents cascading failures
- Allows services to recover gracefully
- Protects system resources

132. Tools that support Circuit Breaker in Java

- Resilience4j
- Hystrix (deprecated)
- Spring Cloud Circuit Breaker

133. What is the Bulkhead pattern?

It limits concurrent access to a service, preventing one service's failure from affecting others.

134. Why is Bulkhead pattern useful?

- Isolates failures
- Ensures high availability
- Prevents resource exhaustion

135. What is the Timeout pattern?

Limits the time a service waits for a response. Prevents long waits and fails fast on unresponsive services.

136. What is the Fallback pattern?

Provides an alternative response when a call fails, such as a cached value or default response.

137. What is a Rate Limiter?

Restricts the number of requests a service can handle over time to prevent overload and abuse.

138. Tools used for resilience in Spring Boot

- Resilience4j
- Spring Retry
- Spring Cloud Circuit Breaker

139. What is the difference between Retry and Circuit Breaker?

Retry: Keeps trying on failure.
Circuit Breaker: Stops trying after failure threshold to prevent overload.

140. How do you integrate Resilience4j in Spring Boot?

Add Resilience4j dependency and use annotations like @Retry, @CircuitBreaker, and fallback methods.