ONE PAGE SUMMARY

IT & Software › Microservices › Resilience Patterns

Introduction to Resilience Patterns

Master resilience patterns to enhance microservices reliability.

DEFINITION

> Resilience patterns are strategies used to ensure that microservices can withstand failures and continue functioning. They help maintain system stability and performance under adverse conditions.

KEY CONCEPTS

Circuit Breaker

A circuit breaker prevents a service from trying to execute an operation that is likely to fail, allowing it to recover.

Retry Pattern

The retry pattern automatically attempts to re-execute a failed operation a specified number of times before giving up.

Bulkhead

Bulkheads isolate different services or components to prevent a failure in one from affecting others.

Timeout

Setting a timeout ensures that a service does not wait indefinitely for a response, thus freeing up resources.

Fallback

A fallback provides an alternative response or action when a service fails, ensuring continuity.

EXAMPLES

→ Using a circuit breaker to prevent cascading failures in a payment service.

→ Implementing retries for a database connection that occasionally times out.

→ Applying bulkhead isolation in a microservices architecture to protect critical services.

MEMORY TIPS

★ Remember 'CBRFT' for Circuit Breaker, Retry, Fallback, Timeout.

★ Visualize a circuit breaker switch to recall its function in preventing failures.

★ Think of bulkheads in ships to remember service isolation.

COMMON MISTAKES

✗ Neglecting to set appropriate timeout values.

✗ Overusing retries without considering the underlying issue.

✗ Failing to implement fallback strategies for critical services.

QUICK RECAP

Resilience patterns are essential for maintaining microservices reliability. Key patterns include circuit breakers, retries, bulkheads, timeouts, and fallbacks. Understanding and applying these patterns helps prevent system failures and ensures consistent performance.