Interview Questions on High Availability, Fault Tolerance & Failover

1. What is High Availability (HA) and why is it important in system design?

Answer:
High Availability (HA) refers to the ability of a system or component to be continuously operational and available with minimal downtime. The goal is to ensure that services are accessible and functional as much as possible, with systems designed to handle failures gracefully without significant disruption.

Importance in System Design:
HA is crucial because modern systems, especially web-based applications, need to be reliable 24/7 to meet user expectations. Even short periods of downtime can lead to revenue loss, reputational damage, and customer dissatisfaction. Designing systems for HA means they can withstand failures without causing prolonged service interruptions, ensuring a seamless user experience.

---

2. Describe the difference between active-active and active-passive redundancy. When would you choose one over the other?

Answer:

- **Active-Active Redundancy:**
  In an active-active configuration, multiple instances of a service or system run in parallel, sharing the workload. If one instance fails, others can immediately take over without any service interruption. This setup is more complex but provides better fault tolerance and load distribution.

- **Active-Passive Redundancy:**
  In active-passive, one system (the active node) handles all the traffic, while the other system (the passive node) remains idle until a failure occurs. In case of failure, the passive system takes over. It's simpler to set up than active-active but may involve longer recovery times since the passive system must "take over" during failover.

When to choose one over the other:

- **Active-Active** is preferred when high fault tolerance and continuous operation are critical, and load balancing is essential. It's typically used in scenarios with high traffic or mission-critical applications.

- **Active-Passive** is more suitable for scenarios where cost is a concern, and the application can tolerate some downtime during failover, such as less critical systems or less frequent traffic.

---

3. What is fault tolerance, and how does it differ from high availability?

Answer:

- **Fault Tolerance:**
  Fault tolerance is the ability of a system to continue operating even if one or more of its components fail. A fault-tolerant system is designed to maintain functionality and minimize disruption during failures by employing redundancy, failover mechanisms, and error detection.

- **Difference from High Availability:**
  High availability focuses on minimizing downtime, but fault tolerance goes a step further by ensuring that the system continues operating during failures without any downtime at all. While HA might involve reducing downtime by shifting to backup systems, fault-tolerant systems are built to prevent disruptions even when a failure occurs.

Example:

- In HA, a server failure might trigger a failover, causing a brief interruption.

- In fault-tolerant systems, the failure is handled automatically without interrupting service, such as redundant power supplies or memory systems in hardware.

---

4. What is graceful degradation, and how does it improve the user experience during a system failure?

Answer:

Graceful Degradation refers to designing a system in such a way that it can still function with limited features or performance when a failure occurs. Instead of the system failing completely, the application reduces its functionality to provide users with an acceptable experience, even if not optimal.

Improvement to User Experience:

- **Example 1:** A shopping website that goes down for payment processing can still allow users to browse products and add items to the cart, rather than showing a

complete "site down" message.

- **Example 2:** In a video streaming service, if the video resolution is compromised due to bandwidth issues, the system might degrade to a lower resolution, ensuring the user can still watch the content without interruption.

This approach improves user satisfaction by avoiding complete failure, helping maintain trust even during issues.

---

5. How do load balancers contribute to high availability in a distributed system?

Answer:

Load balancers are critical for distributing incoming traffic across multiple servers or instances of a service. They ensure that the load is spread evenly, preventing any single instance from becoming overloaded, which could lead to performance degradation or failure.

Contribution to HA:

- **Fault Tolerance:** If one server or instance fails, the load balancer automatically redirects traffic to healthy servers, ensuring no downtime for the user.

- **Scalability:** Load balancers can scale the system horizontally by adding or removing instances based on traffic, ensuring the system remains responsive under varying loads.

- **Traffic Distribution:** By balancing requests, they ensure that no single component becomes a bottleneck, reducing the chances of system failure due to overloading.

---

6. What is failover, and how does it help maintain availability during system failure?

Answer:

Failover is the process by which a system automatically shifts to a backup system or component when the primary system fails. This ensures continuity of service during failures without requiring manual intervention.

How it helps maintain availability:

- **Minimizes Downtime:** The system quickly switches to a secondary server, database, or resource without causing significant downtime.

- **Improves Fault Tolerance:** Failover mechanisms ensure that systems continue running even when certain components (such as a server, database, or network)

become unavailable.

- **Example:** In a web application, if the primary server goes down, a failover system immediately shifts traffic to a secondary server, allowing the service to continue without interruption.

---

7. How can health monitoring and self-healing systems help improve system reliability and availability?

Answer:

Health Monitoring refers to continuously checking the health of various system components (such as servers, databases, and services) to ensure they are functioning correctly. If a failure is detected, the system can trigger recovery actions or alerts.

Self-Healing Systems automatically detect issues and take corrective actions without human intervention. These systems might restart failed services, replace unhealthy nodes, or scale resources dynamically to mitigate failures.

How they help:

- **Proactive Fault Detection:** Health monitoring allows systems to detect and address issues before they lead to significant failures.

- **Automated Recovery:** Self-healing systems can automatically recover from failures, reducing downtime and minimizing the need for manual intervention.

- **Improves Availability:** Continuous monitoring ensures that the system can detect failures quickly and take action to keep services available.

Example:

- A cloud platform like AWS uses health checks to ensure EC2 instances are operational. If an instance fails, the system automatically replaces it with a new one, maintaining service availability.