

Надежность в архитектуре веб-приложений — это способность системы функционировать корректно и без сбоев в течение определенного времени при определенных условиях. Включает в себя также готовность к восстановлению после возможных сбоев. Надежность — это комплексный атрибут, и для его оценки используются различные метрики и критерии.

На уровне Solution Architect особое внимание уделяется следующим метрикам: **Доступность (Availability)**, **Отказоустойчивость (Fault Tolerance)**, и **Восстанавливаемость (Recoverability)**.

Доступность (Availability)

Доступность — это мера, показывающая, насколько система способна предоставить свои услуги пользователям, когда это необходимо. Это не просто время без отказов; это комплексная характеристика, включающая в себя множество аспектов, таких как резервирование ресурсов, географическое распределение, автоматическое и ручное восстановление, а также планирование обслуживания.

Примеры кейсов

- Е-commerce платформа в праздничные дни:** Для обеспечения высокой доступности веб-сайта в праздничные дни, когда количество посетителей резко возрастает, нужно предварительно подготовиться: добавить дополнительные серверы, оптимизировать базы данных и так далее.
- Стриминговый сервис:** Если стриминговый сервис недоступен даже небольшое количество времени, это может привести к потере большого числа пользователей и доходов. Доступность в этом контексте может включать в себя распределенные системы кэширования и автоматическое переключение между серверами в разных регионах.
- Финансовые системы:** В финансовых системах высокая доступность критична. Решения могут включать в себя использование горячего резервирования с автоматическим переключением на резервный сервер в случае сбоя основного.

Отказоустойчивость (Fault Tolerance)

Отказоустойчивость — это свойство системы продолжать функционировать корректно в условиях частичных сбоев, будь то проблемы с отдельными компонентами, сетевыми задержками или даже ошибками в программном коде. Это не только техническая характеристика; она включает в себя организационные и процессные аспекты, такие как мониторинг, оповещения и процедуры восстановления.

Примеры кейсов

- 1. **Микросервисная архитектура:** Если один из микросервисов выходит из строя, остальная система продолжает работать, возможно, с некоторыми ограничениями. Сбой одного сервиса не ведет к поломке всей системы.
- 2. **Системы телекоммуникаций:** Отказоустойчивость может быть обеспечена через резервирование каналов и маршрутизаторов, так что в случае сбоя трафик автоматически перенаправляется.
- 3. **Базы данных:** В системах с высокой степенью отказоустойчивости часто используются распределенные базы данных с репликацией и шардированием для обеспечения работы даже при потере одного или нескольких узлов.

Восстанавливаемость (Recoverability)

Восстанавливаемость — это способность системы возвращаться в рабочее состояние после сбоя в минимально возможное время. Это включает в себя автоматическое и ручное восстановление, а также стратегии резервного копирования и хранения данных.

Примеры кейсов

- 1. **Катастрофические события:** В случае катастрофических событий (например, пожара в дата-центре), необходимы специализированные дисастер-рекавери планы, которые позволяют быстро восстановить работу.
- 2. **Восстановление после сбоя программного обеспечения:** Если после обновления программного обеспечения происходит сбой, должна быть возможность быстрого отката к предыдущей версии.
- 3. **Атаки и безопасность:** После кибератак, таких как DDoS-атаки или взломы, система должна иметь возможность быстро восстановить свою работу и устранить уязвимости.

Эти метрики и кейсы являются критическими для обеспечения надежности любой системы и должны быть тщательно рассмотрены на уровне архитектуры решения. В больших корпорациях невозможно выпустить новый сервис в продакшен (для пользователей), не обсудив конкретные значения этих метрик. **Но какие они могут быть?**