

# Обеспечение восстанавливаемости

## Технические инструменты:

Уже обсуждали:

- **Репликация данных:** Дублирование данных на нескольких узлах
- **Журналирование, мониторинг и аудит:** Подробное журналирование, мониторинг, операций и изменений.

Есть еще инструменты:

- **Автоматизация восстановления**

Автоматизация восстановления заключается в использовании скриптов, рабочих процессов и инструментов для минимизации времени и усилий, необходимых для восстановления системы после сбоя.

## Как работает:

Инструменты и скрипты мониторят состояние системы и автоматически применяют ранее определенные действия для восстановления нормальной работы в случае обнаружения сбоя.

## Варианты реализации:

1. **Скрипты для автоматического восстановления:** Можно написать скрипты, которые автоматически перезапускают сервисы, очищают кэш и т.д.
2. **Orchestration инструменты:** Инструменты вроде Kubernetes могут автоматически восстановить контейнеры и управлять ресурсами.

## Пример:

Система обнаруживает, что основной сервер базы данных не отвечает. Автоматический скрипт переключает трафик на резервный сервер и запускает процедуру восстановления основного сервера.

- **Rollback и Versioning**

Rollback — это возможность отката системы или кода до предыдущей, рабочей версии. Versioning — это практика сохранения различных версий кода или конфигураций для возможности быстрого восстановления.

Как работает:

1. **Rollback:** При обнаружении сбоя или ошибки нового кода система может быть быстро откатена до предыдущей, стабильной версии.
2. **Versioning:** Сохраненные версии позволяют выбрать нужную версию для восстановления.

Варианты реализации:

1. **Database Rollback:** В базах данных может быть реализован механизм для отката транзакций.
2. **Code Versioning:** Использование систем версионирования кода, таких как Git.
3. **Configuration Management:** Инструменты вроде Ansible могут хранить и управлять разными версиями конфигураций.

Пример:

После выпуска новой версии приложения выявлены серьезные баги. Система автоматически откатывается на предыдущую, стабильную версию, минимизируя время простоя и воздействие на пользователей.

В обоих случаях ключевым является подготовка: необходимо заранее создать скрипты, рабочие процессы и сценарии для возможности быстрого и эффективного восстановления. Это позволяет снизить человеческий фактор и время реакции при критических сбоях.

Организационные практики:

- **Планы по реагированию на инциденты:** Шаги и процедуры для восстановления после сбоев.
- **Команда восстановления (сейчас популярны SRE-инженеры):** Специализированная команда для операций по восстановлению.