Настоящий документ описывает наиболее вероятные сценарии отказа в работе системы/сервиса, их потенциальное влияние и последовательность действий для их устранения и минимизации ущерба. Цель - обеспечить быстрое и эффективное восстановление работоспособности.

**Сценарий 1: Сбой в работе базы данных (БД).**

Влияние данного сбоя критично, может привести к отказу работы основного функционала и недоступности данных для пользователей.

Признаками являются:

* Высокая нагрузка на CPU/память сервера БД.
* Медленные или завершающиеся ошибкой запросы к БД.
* Появление ошибок типа "Connection timeout", "Database not available" в логах приложения.
* Полная недоступность приложения.

План действий для устранения сбоя и его последствий:

1. Обнаружение: Мониторинг обнаруживает аномалии. Автоматическое оповещение команде.
2. Изоляция проблемы:

* Проверить доступность сервера БД (ping, подключение).
* Проверить логи БД на наличие ошибок (например, нехватка места, блокировки).
* Определить, является ли проблема аппаратной (память, диск) или связана с конкретным "тяжелым" запросом.

1. Восстановление (краткосрочное):

* Если нехватка ресурсов: перезапустить экземпляр БД. Увеличить ресурсы (если облако).
* Если "тяжелый" запрос: выявить и остановить проблемный процесс/запрос.
* Если полный сбой: переключиться на реплику базы данных.

1. Восстановление (долгосрочное):

* Проанализировать корневую причину.
* Оптимизировать проблемные запросы или индексы.
* Увеличить мощность сервера или настроить более эффективное масштабирование.
* Улучшить мониторинг для более раннего обнаружения аналогичных проблем.

**Сценарий 2: Потеря сетевой связности или DDoS-атака.**

Данное происшествие имеет высокое влияние. В следствии происшествия пользователи не смогут получить доступ к сервису, и компания получит репутационный ущерб.

Признаки данного происшествия:

* Резкий всплеск входящего сетевого трафика.
* Высокая нагрузка на сетевые интерфейсы балансировщика нагрузки или брандмауэра.
* Рост количества HTTP-ошибок 5xx.
* Недоступность сервиса извне при работоспособности внутренних компонентов.

План действий:

1. Обнаружение: Срабатывание мониторинга сетевой активности. Оповещение команды и ИБ-специалистов.
2. Изоляция проблемы:

* Определить источник трафика (легитимный или вредоносный).
* Проверить доступность upstream-провайдеров.
* Проанализировать логи брандмауэра и балансировщика нагрузки.

1. Восстановление (краткосрочное):

* Если DDoS: Включить защиту от DDoS у облачного провайдера или хостинга. Применить гео-фильтрацию или blackhole-маршрутизацию для трафика из определенных регионов.
* Если сбой сети: Переключиться на резервный канал связи (если есть).
* Временно заблокировать IP-адреса - источники атаки на уровне брандмауэра.

1. Восстановление (долгосрочное):

* Проанализировать векторы атаки.
* Настроить постоянные правила WAF (Web Application Firewall).
* Пересмотреть и усилить сетевую архитектуру для отказоустойчивости.

**Сценарий 3: Сбой внешней зависимости (API, микросервис, платежный шлюз).**

Данное происшествие имеет среднее или высокое влияние, в зависимости от важности части функционала приложения, которая перестанет работать в следствии данного сбоя.

План действий:

1. Обнаружение: Мониторинг внешних вызовов показывает сбой. Оповещение команды.
2. Изоляция проблемы:

* Проверить статус-страницу внешнего сервиса.
* Протестировать подключение к API извне.
* Определить, является ли сбой полным или частичным.

1. Восстановление (краткосрочное):

* Реализовать механизм циркуляции выключателя в коде приложения, чтобы избежать лавинообразного отказа.
* Временно отключить не критичный функционал, зависящий от этого сервиса.
* Использовать резервный или фолбэк-сервис, если он предусмотрен (например, альтернативный SMS-шлюз).

1. Восстановление (долгосрочное):

* Связаться с поставщиком сервиса для выяснения причин.
* Увеличить таймауты и улучшить обработку ошибок на стороне приложения.
* Рассмотреть возможность использования асинхронной обработки запросов к данному сервису.

**Сценарий 4: Потеря данных на файловом хранилище.**

Данное происшествие имеет критическое влияние. Может привести к безвозвратной потере пользовательских файлов, статического контента, конфигураций.

Признаки:

* Ошибки приложения при обращении к файлам ("File not found").
* Пользователи сообщают о битых изображениях или отсутствии файлов.
* Срабатывание мониторинга, отслеживающего целостность данных.

План действий:

1. Обнаружение: Получение оповещения от мониторинга или сообщений от пользователей. Немедленная эскалация.
2. Изоляция проблемы:

* Определить масштаб потери (один файл, каталог, все хранилище).
* Выяснить причину (ошибка человека, сбой диска, вредоносное ПО).

1. Восстановление (краткосрочное):

* Немедленно остановить любые операции записи в пораженную область.
* Восстановить данные из последней проверенной резервной копии.
* Если сбой аппаратный - заменить вышедший из строя диск.

1. Восстановление (долгосрочное):

* Проверить и исправить процедуры резервного копирования (регулярность, тестирование восстановления).
* Внедрить систему контроля целостности данных.
* Рассмотреть возможность использования реплицируемого или географически распределенного хранилища.

**Сценарий 5: Ошибочное развертывание (Deployment) новой версии.**

Имеет высокое влияние. Грозит появлением багов, деградацией производительности, частичной недоступности функционала.

Признаки:

* Рост количества ошибок (в т.ч. новых) сразу после деплоя.
* Падение ключевых метрик производительности (response time, success rate).
* Жалобы пользователей на новый функционал.

План действий:

1. Обнаружение: Срабатывание мониторинга, отслеживающего метрики здоровья после деплоя. Оповещение команды.
2. Изоляция проблемы:

* Связать время начала проблем с временем деплоя.
* Проанализировать логи нового кода для выявления ошибки.

1. Восстановление (краткосрочное):

* Немедленный откат на предыдущую стабильную версию приложения. Это приоритет №1.
* Если откат невозможен, выпустить "горячий фикс" для устранения критической ошибки.

1. Восстановление (долгосрочное):

* Провести анализ постмортем для выявления причин: почему баг не был отловлен на стадиях тестирования и CI/CD.
* Улучшить процесс тестирования (добавить интеграционные, нагрузочные тесты).
* Внедрить стратегии постепенного развертывания (canary-релизы, feature flags).

Для хранения скриптов и deb-пакетов можно использовать Gitlab. Он очень удобен и полезен благодаря своему инструменталу.

Функции gitlab:

* Хранение кода (репозиторий) — можно хранить исходники проектов, вести ветвление, откатываться к нужным версиям, проводить сравнения и код-ревью.
* Планирование и трекинг задач - GitLab Issues позволяют быстро фиксировать баги, назначать ответственных, ставить сроки и следить за прогрессом. Есть доски (Kanban) для визуального управления задачами.
* CI/CD-пайплайны - автоматизируют сборку, тесты и деплой. Каждый пуш в репозиторий может запустить проверку и сборку кода.
* Контроль доступа - гибкая система ролей помогает разграничить права участников, что защищает от случайных удалений веток или нежелательных слияний кода.
* Вики и документация - у каждого проекта есть встроенная Wiki, можно вести инструкции для разработчиков, тестировщиков и пользователей.
* Управление релизами - GitLab помогает отмечать конкретные версии кода тегами, создавать релизы и хранить всю историю изменений.