УТВЕРЖДАЮ
Генеральный директор
ООО «НАЙС СОФТ ГРУПП»
_____ С.С. Беликов
«____» _____ 2026 г.

ПРОГРАММНЫЙ КОМПЛЕКС «НАЙС СОФТ ИИ СТУДИЯ»

версия 1.0.0

ОПИСАНИЕ ПОРЯДКА ОБНОВЛЕНИЯ

НССГ.10001-01 93 01-ЛУ

Лист утверждения

Красногорск
2026

Утвержден НССГ.10001-01 93 01-ЛУ

ООО «НАЙС СОФТ ГРУПП»

ПРОГРАММНЫЙ КОМПЛЕКС «НАЙС СОФТ ИИ СТУДИЯ»

Версия 1.0.0

ОПИСАНИЕ ПОРЯДКА ОБНОВЛЕНИЯ

НССГ.10001-01 93 01

Листов 53

Красногорск
2026

АННОТАЦИЯ

Настоящий документ описывает порядок получения, проверки, подготовки, установки, контроля и отката обновлений программного комплекса «Найс Софт ИИ Студия» версии 1.0.0. Документ предназначен для администраторов, специалистов эксплуатации, служб информационной безопасности, технической поддержки и сотрудников, выполняющих выпуск и квалификацию обновлений.

В документе зафиксированы каналы выпуска, криптографическая проверка метаданных, работа с неизменяемыми контейнерными образами, предварительная проверка инфраструктуры, обязательное резервное копирование, режим обслуживания, договор миграции данных, контроль работоспособности, автоматический и ручной откат, а также обновление в закрытом и полностью изолированном контуре.

Ключевой принцип

Обновление ИИ Студии является транзакционной операцией: новая версия считается рабочей только после успешного запуска контейнеров, прохождения контроля работоспособности и внутренней самопроверки платформы. При неуспехе выполняется автоматический возврат к предыдущей версии; если выпуск изменяет данные несовместимым образом, дополнительно восстанавливается проверенная резервная копия, созданная до обновления.

СТАТУС ДОКУМЕНТА

Документ является корпоративным эксплуатационно-техническим документом комплекта программной документации ИИ Студии. Он дополняет руководство администратора, руководство по установке, системные требования, матрицу совместимости, политику жизненного цикла и описание информационной безопасности.

Оформление документа выполнено с учётом общих требований ЕСПД к программным документам. Механизмы защищённого выпуска и обновления описываются с учётом требований к безопасной разработке программного обеспечения.

Нормативный документ	Назначение
ГОСТ 19.101-2024	Виды программ и программных документов ЕСПД.
ГОСТ 19.105-78	Общие требования к программным документам.
ГОСТ 19.106-78	Требования к программным документам, выполненным печатным способом.
ГОСТ Р 56939-2024	Общие требования к безопасной разработке программного обеспечения и устранению уязвимостей.

Официальные источники Росстандарта: https://protect.gost.ru/gost/details/500f8c05-0b04-40fc-a049-4212eae807c3 ; https://protect.gost.ru/gost/details/237d9e0c-a8a8-40a2-a2f5-f8182da346a8 ; https://protect.gost.ru/gost/details/f3818925-a96f-4f55-96e9-46b44720ee64

СОДЕРЖАНИЕ

1 Общие положения 6
2 Термины и определения 7
3 Основные принципы системы обновлений 8
4 Нумерация версий и виды выпусков 9
5 Каналы обновлений 10
6 Источник обновлений и структура публикации 11
7 Криптографическая модель доверия 12
8 Политика контейнерных образов 13
9 Состав выпуска обновления 14
10 Архитектура клиентского механизма обновления 15
11 Проверка наличия обновления 16
12 Предварительная проверка перед обновлением 17
13 Подготовка и предварительная загрузка обновления 18
14 Обязательное резервное копирование 19
15 Режим обслуживания 20
16 Договор миграции данных 21
17 Переключение на новую версию 22
18 Контроль работоспособности и самопроверка 23
19 Успешное завершение обновления 24
20 Автоматический откат 25
21 Ручной откат 26
22 Возврат данных при несовместимой миграции 27
23 Смена канала обновлений 28
24 Понижение версии и запрет неявного возврата 29
25 Исправления безопасности и аварийные выпуски 30
26 Обновление в закрытом контуре 31
27 Полностью изолированное обновление 32
28 Сетевые требования обновления 33
29 Дисковое пространство и кэширование 34
30 Права доступа и административный аудит 35
31 Мониторинг обновлений 36
32 Ошибки и восстановление после сбоя 37
33 Публикация выпуска на стороне НАЙС СОФТ 38
34 Ротация ключа подписи выпусков 39
35 Сторонние компоненты и внешние реестры 40
36 Совместимость и ограничения 41
37 Пошаговая процедура обновления администратором 42
38 Критерии приёмки обновления 43
39 Хранение сведений и прослеживаемость 44
40 Взаимосвязь с политикой жизненного цикла 45
Приложение А - Состояния процесса обновления 46
Приложение Б - Контрольный лист обновления 47
Приложение В - Матрица решения об откате 48
Приложение Г - Состав автономного пакета обновления 49
Приложение Д - Структура манифеста выпуска 50
Приложение Е - Форма протокола обновления 51
Приложение Ж - Диагностические команды и точки контроля 52
Лист регистрации изменений 53

1 Общие положения

Система обновлений ИИ Студии предназначена для контролируемой замены исполняемого программного окружения без сборки исходного кода на сервере заказчика. Обновление выполняется готовыми контейнерными образами и неизменяемым пакетом среды выполнения, сведения о которых закреплены в подписанном манифесте выпуска.

Обновление разделено на независимые этапы: обнаружение выпуска, проверка метаданных и подписи, предварительная проверка сервера, подготовительная загрузка, обязательное резервное копирование с проверкой, включение режима обслуживания, переключение контейнерной среды, контроль работоспособности, фиксация новой версии либо автоматический возврат.

- клиентская установка не выполняет сборку контейнеров и не компилирует исходный код;
- каждый устанавливаемый контейнер закреплён за точным SHA-256 digest;
- метаданные выпуска и выбранного канала проверяются по подписи Ed25519;
- перед изменением рабочей версии создаётся и проверяется зашифрованная резервная копия;
- предыдущая управляемая среда выполнения сохраняется для безопасного возврата;
- изменения данных описываются явным договором миграции;
- успех определяется не фактом запуска команды, а прохождением контроля работоспособности и самопроверки платформы.

Область применения

Процедура относится к управляемым продуктивным установкам ИИ Студии 1.0.0. Сборка из исходных файлов на стенде разработчика является отдельным процессом разработки и не считается штатным механизмом обновления заказчика.

2 Термины и определения

Термин	Определение
Выпуск	Зафиксированный набор контейнерных образов, файлов среды выполнения, манифеста, подписей и сведений о миграции, имеющий уникальную версию.
Канал обновлений	Логический поток публикации выпусков: стабильный, кандидат или разработка.
Манифест	Подписанный документ, содержащий версию, состав выпуска, неизменяемые ссылки на образы, требования, сведения о миграции и контрольные значения.
Подготовленное обновление	Выпуск, метаданные, пакет среды выполнения и контейнерные образы которого уже загружены и проверены, но рабочая версия ещё не переключалась.
Среда выполнения	Набор compose-файлов, политик и связанных неизменяемых контейнерных образов, необходимый для запуска конкретного выпуска.
Контроль работоспособности	Автоматическая проверка состояния контейнеров и критических служб после переключения.
Самопроверка платформы	Функциональная проверка ключевых внутренних связей ИИ Студии после старта новой среды.
Откат	Возврат к предыдущей управляемой версии среды выполнения; при необходимости - с восстановлением данных из резервной копии.
Страховочная резервная копия	Копия текущего состояния, создаваемая непосредственно перед ручным откатом для возможности отменить сам откат.
Договор миграции	Подписанная часть метаданных выпуска, определяющая характер изменений данных и правила автоматического/ручного возврата.

3 Основные принципы системы обновлений

Механизм обновления построен по принципу «сначала доказательство готовности - затем изменение рабочей среды». Подготовительные действия не должны изменять активную версию продукта.

Принцип	Реализация
Неизменяемость	Образы задаются ссылками repository@sha256; опубликованный стабильный выпуск не переписывается.
Подлинность	Описание канала и манифест выпуска подписываются Ed25519.
Атомарность публикации	Сначала публикуются неизменяемые файлы выпуска и подписи, указатель канала активируется последним.
Предварительная подготовка	Пакет и образы можно загрузить заранее без переключения рабочей версии.
Резервирование	Установка блокируется, если служба резервного копирования не готова либо созданная копия не прошла проверку.
Транзакционный возврат	Неуспешная новая версия автоматически заменяется предыдущей средой; несовместимые изменения данных возвращаются из pre-update backup.
Наблюдаемость	Этапы, процент выполнения, ошибки, версия и состояние отката доступны административному интерфейсу и метрикам.
Минимальные права	Изменение канала и установка доступны только администратору через серверный контур управления.

4 Нумерация версий и виды выпусков

Публичные версии продукта используют схему MAJOR.MINOR.PATCH. Исправление ранее опубликованной версии всегда получает новый номер; опубликованный стабильный выпуск не заменяется содержимым под тем же номером.

Вид	Пример	Назначение
Основной выпуск	1.0.0 -> 2.0.0	Крупные изменения совместимости, архитектуры или продукта.
Функциональный выпуск	1.0.0 -> 1.1.0	Новые совместимые функции и расширения.
Исправление	1.0.0 -> 1.0.1	Исправления дефектов, безопасности и небольшие совместимые изменения.
Кандидат	1.0.1-rc...	Приёмочное и пилотное тестирование перед стабильной публикацией.
Разработка	1.0.1-dev.<метка>	Внутренние сборки для разработки и ранней проверки; каждая сборка уникальна.

Запрет перезаписи

Если после публикации версии 1.0.0 обнаружен дефект, исправление публикуется как 1.0.1. Повторная публикация иного содержимого под версией 1.0.0 не допускается.

5 Каналы обновлений

Канал	Идентификатор	Назначение	Рекомендация
Стабильный	stable	Проверенные публичные выпуски для рабочих установок.	Рекомендуемый канал продуктивной эксплуатации.
Кандидат	candidate	Релизы, прошедшие внутренние проверки и переданные на пилот/приёмку.	Использовать на тестовых и пилотных стендах.
Разработка	development	Ранние подписанные сборки текущей линии разработки.	Только разработка и контролируемое раннее тестирование.

Администратор выбирает канал в разделе «Параметры -> Обновление -> Канал обновлений». Выбор сохраняется в постоянном хранилище настроек и переживает перезапуск/пересоздание контейнеров. Смена канала сама по себе не загружает и не устанавливает выпуск.

Выбор предварительного канала или канала разработки требует дополнительного явного подтверждения. Смена канала регистрируется в административном журнале. Если выбранный канал содержит выпуск старше уже установленного, автоматическое понижение версии не предлагается.

6 Источник обновлений и структура публикации

Каноническая точка обнаружения выпусков располагается на официальном узле продукта studio.ncsgp.ru. Для стабильного канала используется HTTPS-описание канала, которое ссылается на конкретный неизменяемый каталог выпуска.

```
https://studio.ncsgp.ru/updates/v1/channels/stable.json
https://studio.ncsgp.ru/updates/v1/channels/candidate.json
https://studio.ncsgp.ru/updates/v1/channels/development.json

https://studio.ncsgp.ru/updates/v1/releases/<version>/manifest.json
```

Объект	Назначение
channels/<channel>.json	Текущий указатель выбранного канала.
channels/<channel>.json.sig	Подпись Ed25519 описания канала.
channels/<channel>.json.sha256	Контрольная сумма опубликованных байтов.
releases/<version>/manifest.json	Канонический манифест конкретного выпуска.
manifest.json.sig	Подпись манифеста.
runtime-bundle.tar.gz	Пакет файлов среды выполнения без исходной сборки.
compose.runtime.yml	Состав запуска с неизменяемыми образами.
changelog.md	Описание изменений выпуска.
publication.json	Технические сведения о публикации и происхождении артефактов.

7 Криптографическая модель доверия

Обязательной криптографической основой протокола обновления является подпись Ed25519. Закрытый ключ подписи не включается в исходный код, архив поставки, контейнеры или пакет среды выполнения. Клиентская служба обновления содержит только доверенный открытый ключ либо набор доверенных открытых ключей.

- проверяется подпись описания выбранного канала;
- проверяется подпись манифеста конкретного выпуска;
- проверяется SHA-256 пакета среды выполнения;
- каждый контейнер дополнительно закреплён точным digest, находящимся внутри подписанного манифеста;
- для собственных образов может применяться дополнительная подпись Cosign, но она не заменяет обязательную подпись манифеста;
- сторонние образы не переподписываются ключом НАЙС СОФТ: доверие обеспечивается подписанным закреплением исходного repository@sha256.

Модель угроз

Подмена только адреса контейнерного реестра или тега недостаточна для успешной атаки: клиент принимает только точный digest из подписанного манифеста. Подмена манифеста либо указателя канала выявляется по подписи.

8 Политика контейнерных образов

Собственные контейнеры НАЙС СОФТ публикуются в registry.ncsgp.ru/nicesoft-ai-studio. Сторонние контейнеры сохраняют исходный реестр и фиксируются только точным digest. Это позволяет не выдавать стороннее ПО за собственную сборку и сохраняет прослеживаемость происхождения компонентов.

Категория	Где хранится	Как фиксируется
Собственные службы НАЙС СОФТ	Harbor registry.ncsgp.ru/nicesoft-ai-studio	repository@sha256 в подписанном манифесте; при наличии - дополнительная подпись Cosign.
Сторонние службы	Исходный реестр поставщика	Исходный repository@sha256 в подписанном манифесте.
Метаданные выпуска	studio.ncsgp.ru/updates/v1	Подпись Ed25519 + SHA-256; при необходимости дополнительное OCI-зеркало метаданных.

На сервере заказчика запрещены сборка образов, применение Dockerfile из пакета обновления и использование незакреплённых тегов. Пакет среды выполнения проверяется на отсутствие Dockerfile, символьных/жёстких ссылок и небезопасных путей.

9 Состав выпуска обновления

Компонент	Обязательность	Назначение
Подписанный манифест	Обязательно	Версия, требования, состав, образы, миграции, ссылки, контрольные значения.
Подписанный указатель канала	Обязательно	Связывает канал с опубликованным выпуском.
Пакет среды выполнения	Обязательно	Файлы, необходимые для запуска без сборки на клиенте.
Контейнерные образы	Обязательно	Готовые неизменяемые исполняемые артефакты.
Описание изменений	Обязательно для публичного выпуска	Перечень изменений, исправлений и известных ограничений.
Договор миграции	Обязательно	Правила обращения с данными и отката.
SBOM/лицензионные сведения	Для официальной поставки	Прослеживаемость компонентов и лицензий.
Дополнительное OCI-зеркало метаданных	Опционально	Аудит и резервное хранение метаданных.

10 Архитектура клиентского механизма обновления

Обновление разделено между постоянно работающей службой обновлений и одноразовым исполнителем переключения. Служба обновлений получает и проверяет метаданные, выполняет предварительные проверки, загружает пакет и контейнеры, создаёт резервную копию и запускает доверенный одноразовый исполнитель.

Слой	Ответственность
Административный интерфейс	Показывает текущую/доступную версию, канал, сведения о подписи, предварительную проверку, подготовку, установку и состояние отката.
Служба обновлений	Проверяет канал/манифест, требования, дисковое пространство, Docker, резервирование; загружает пакет и образы; создаёт pre-update backup.
Одноразовый исполнитель	Переключает runtime, выполняет контроль работоспособности и самопроверку, фиксирует новую версию либо возвращает предыдущую.
Служба резервного копирования	Создаёт, проверяет и при необходимости восстанавливает зашифрованную резервную копию.
Контур обслуживания	Временно блокирует изменяющие пользовательские операции в критической фазе обновления/восстановления.

Разделение ролей

Даже подготовленный выпуск не переключает рабочую версию. Фактическое изменение выполняется только после отдельной команды установки и прохождения обязательного резервирования.

11 Проверка наличия обновления

При проверке обновлений служба получает описание выбранного канала, проверяет его подпись и извлекает ссылку на выпуск. Затем загружается манифест выпуска и проверяется его подпись. До успешной проверки подписей выпуск не признаётся устанавливаемым.

- определяется текущая установленная версия;
- определяется последняя версия выбранного канала;
- выполняется сравнение версий без автоматического понижения;
- проверяется состояние выпуска final;
- проверяется допустимый реестр собственных образов;
- проверяется запрет сборки и компиляции на сервере заказчика;
- проверяется наличие и корректность договора миграции и отката.

12 Предварительная проверка перед обновлением

Перед подготовкой/установкой администратор может выполнить предварительную проверку. Неуспех любого обязательного пункта блокирует безопасную установку.

Проверка	Критерий успеха
Метаданные	Описание канала и выпуска получены и корректно разобраны.
Новая версия	Целевая версия новее текущей.
Подпись	Подписи Ed25519 подтверждены доверенным ключом.
Политика готовых образов	Сборка на хосте запрещена, ссылки неизменяемые.
Архитектура	Архитектура сервера присутствует в совместимости выпуска; для 1.0.0 основной профиль x86_64/amd64.
Свободное место	Доступно не меньше значения, указанного манифестом; базовая политика генератора - 20 ГиБ.
Docker Engine	Служба Docker доступна и работоспособна.
Резервное копирование	Служба не занята, настроен ключ восстановления.
Контейнеры	Все устанавливаемые образы закреплены за SHA-256 digest.
Разрешение выпуска	Отсутствуют блокирующие причины политики выпуска.

13 Подготовка и предварительная загрузка обновления

Этап подготовки предназначен для уменьшения времени обслуживания. Он может быть выполнен заранее в рабочее время, поскольку не изменяет активную версию ИИ Студии.

1. Загрузить пакет среды выполнения по адресу из подписанного манифеста либо из локального автономного источника.
2. Проверить SHA-256 пакета.
3. Проверить внутреннюю структуру архива и запрет небезопасных путей/ссылок/Dockerfile.
4. Проверить compose-политику: отсутствие build и наличие только неизменяемых image-ссылок.
5. Поочерёдно скачать необходимые контейнерные образы.
6. После каждого скачивания подтвердить соответствие локального образа ожидаемому digest.
7. Зафиксировать состояние «обновление подготовлено» с версией, каналом, хэшем манифеста и идентификатором ключа подписи.

Важно

Подготовка заканчивается сообщением «Рабочая версия не переключалась». Уже загруженные слои контейнеров могут быть повторно использованы при последующей установке.

14 Обязательное резервное копирование

Непосредственно перед переключением рабочей среды служба обновлений создаёт отдельную резервную копию текущего состояния. Создание архива недостаточно: обновление продолжается только после самостоятельной операции проверки созданной копии.

8. Запросить создание резервной копии через внутреннюю службу резервного копирования.
9. Дождаться завершения создания и получить уникальный идентификатор архива.
10. Запустить проверку целостности созданного архива.
11. Убедиться, что проверенная копия имеет тот же идентификатор и статус OK.
12. Сохранить идентификатор как pre-update backup для возможного автоматического или ручного возврата данных.

Резервная копия ИИ Студии имеет формат .nsbackup, шифруется, а её целостность защищается отдельным механизмом контроля. Ключ восстановления конкретной установки не хранится внутри архива и должен быть заранее сохранён администратором вне сервера.

Условие блокировки

Если ключ восстановления резервных копий не настроен, служба резервирования занята либо проверка нового архива неуспешна, установка обновления не должна начинаться.

15 Режим обслуживания

Перед переключением контейнеров включается режим обслуживания. Его назначение - исключить новые изменяющие операции в момент, когда состояние служб и данных может временно изменяться.

- обычные пользовательские операции получают сообщение о техническом обслуживании;
- не запускаются новые задания планировщика автоматизаций;
- изменяющие действия интеграций и автоматизаций блокируются;
- административный канал резервного копирования и наблюдения за операцией сохраняется;
- режим обслуживания снимается только после успешного завершения новой версии либо успешного возврата предыдущей.

16 Договор миграции данных

Каждый выпуск сопровождается явным описанием воздействия на постоянные данные. Оно не выводится из номера версии и не определяется эвристически: правила возврата задаются договором миграции, входящим в подписанные метаданные выпуска.

Класс	Изменение данных	Автоматический неуспех	Ручной откат
none	Нет несовместимой миграции	Возврат только среды выполнения	Возврат только среды выполнения
backward_compatible	Возможны совместимые добавочные изменения	Как указано договором; обычно среда выполнения	Как указано договором
restore_required	Изменения не гарантируют обратную совместимость	Восстановление pre-update backup + предыдущая среда	Восстановление pre-update backup + предыдущая среда

Для добавления новых параметров среды допускается только добавочная миграция: существующие значения не ротируются и не перезаписываются автоматически. Новые секреты, сформированные при обновлении, могут сохраняться при откате, если предыдущая версия их не использует; это предотвращает изменение идентичности уже созданных новых постоянных хранилищ.

17 Переключение на новую версию

Переключение выполняется доверенным одноразовым исполнителем обновления. Он получает только заранее подготовленные пути, версию, неизменяемую ссылку собственного образа исполнителя, идентификатор резервной копии, договор миграции и путь к каноническому установочному файлу .env.

- канонический .env остаётся в каталоге установки и не входит в опубликованный пакет;
- новые обязательные каталоги постоянных данных создаются до запуска новой версии;
- Compose запускается без сборки образов;
- исполнитель работает с конкретной подготовленной средой выполнения, а не с плавающим тегом;
- после запуска выполняется ожидание состояния контейнеров и контроль критических служб.

```
Логическая последовательность:
подготовленный выпуск -> pre-update backup -> обслуживание
-> одноразовый исполнитель -> новая среда выполнения
-> контроль работоспособности -> самопроверка платформы
-> фиксация версии -> снятие обслуживания
```

18 Контроль работоспособности и самопроверка

Успешный запуск контейнеров сам по себе недостаточен. После переключения выполняется двухуровневая проверка: состояние контейнеров/служб и прикладная самопроверка платформы внутри новой версии.

Этап	Что подтверждается
Контроль контейнеров	Критические контейнеры запущены и достигают ожидаемого состояния.
Внутренние службы	Доступны необходимые внутренние сетевые адреса и сервисы.
Самопроверка платформы	Ключевые связи между компонентами ИИ Студии функционируют в собранной конфигурации.
Контур управления	Административная команда управления переводится на новую рабочую среду только после проверок.

Базовое значение времени ожидания контроля работоспособности для выпуска 1.0.0 - 300 секунд. Конкретный выпуск может содержать иное допустимое значение в подписанных метаданных в пределах безопасной политики.

19 Успешное завершение обновления

13. Новая среда запускается и проходит контроль работоспособности.
14. Выполняется самопроверка платформы.
15. Контур командного управления переключается на новую среду.
16. В файл состояния текущего выпуска атомарно записываются новая версия, каталог среды, предыдущая версия/каталог, идентификатор pre-update backup, хэш манифеста, активные профили и договор миграции.
17. Снимается режим обслуживания.
18. Административный интерфейс получает состояние «Обновление установлено».

Прослеживаемость

Состояние новой версии сохраняет ссылку на предыдущую управляемую среду и pre-update backup. Благодаря этому ручной откат не требует поиска случайной старой копии или повторной загрузки неподтверждённого исполнителя.

20 Автоматический откат

Если новая версия не проходит контроль работоспособности или самопроверку, одноразовый исполнитель автоматически инициирует возврат. Пользовательское подтверждение для такого возврата не требуется: это часть одной транзакции установки.

Ситуация	Действие
Данные обратно совместимы	Останавливается/заменяется новая среда, запускается предыдущая среда и повторно проходит контроль работоспособности.
Договор требует возврата данных	Поднимаются только необходимые службы восстановления предыдущей версии, восстанавливается pre-update backup, затем запускается полный предыдущий runtime и проходит контроль.
Откат тоже завершился ошибкой	Операция переводится в критическое состояние, режим обслуживания остаётся включённым, требуется администратор/техническая поддержка.

Сообщение об ошибке содержит первичную причину отказа новой версии и результат возврата. Неуспешный выпуск не должен фиксироваться как текущая рабочая версия.

21 Ручной откат

После успешного управляемого обновления администратор может вернуться к предыдущей сохранённой версии. Для предотвращения случайной потери данных ручной откат требует явного подтверждения словом «ОТКАТИТЬ».

19. Проверить наличие предыдущей управляемой версии и её каталога среды выполнения.
20. Определить по договору миграции, требуется ли возврат данных.
21. Создать и проверить страховочную резервную копию текущего состояния.
22. Проверить сохранённую неизменяемую ссылку на одноразовый исполнитель текущего выпуска.
23. Включить режим обслуживания.
24. При необходимости восстановить исходный pre-update backup предыдущей версии.
25. Запустить предыдущую среду и проверить её работоспособность.
26. Переключить контур управления и зафиксировать новое состояние отката.

Страховочный механизм

Если ручной откат не удаётся, система пытается вернуть исходную текущую версию. Если возврат данных уже начался, для восстановления исходного состояния используется только что созданная страховочная резервная копия.

22 Возврат данных при несовместимой миграции

При договоре миграции restore_required возврат только контейнеров недостаточен. Данные должны быть согласованы с версией программной среды. Поэтому восстановление выполняется до полного запуска предыдущей версии.

- используется только идентификатор проверенной pre-update backup, сохранённый при установке;
- восстановление выполняется внутренней службой резервного копирования;
- режим обслуживания сохраняется на всём протяжении восстановления;
- после возврата данных предыдущая среда обязательно проходит полный контроль работоспособности;
- для ручного отката текущее состояние предварительно сохраняется в страховочной копии.

При восстановлении автоматизаций задания, которые на момент резервирования находились в промежуточном состоянии выполнения, не возобновляются молча: это исключает непреднамеренное повторение внешнего побочного действия после восстановления.

23 Смена канала обновлений

Смена канала выполняется только администратором. Для каналов «Кандидат» и «Разработка» требуется дополнительное подтверждение. После смены очищается только состояние подготовленного выпуска прежнего канала; уже загруженные слои Docker не удаляются.

Если в новом канале нет версии новее установленной, служба отображает отсутствие доступного обновления. Неявный переход на более старую версию не выполняется.

Действие	Результат
stable -> candidate	Проверка кандидатного выпуска после подтверждения; установка не запускается.
candidate -> stable	Возврат к стабильной политике; если stable не новее текущей версии, понижение не предлагается.
stable -> development	Разрешается только после явного подтверждения повышенного риска.
Смена канала при подготовленном выпуске	Prepared-state старого канала очищается; рабочая версия неизменна.

24 Понижение версии и запрет неявного возврата

Обычная проверка обновлений предназначена только для перехода на более новую версию. Понижение версии через смену канала или повторную проверку не выполняется. Возврат к предыдущей версии является отдельной управляемой операцией отката с собственными проверками и резервированием.

Почему это важно

Понижение версии может быть несовместимо со схемой данных, конфигурацией и новыми постоянными хранилищами. Поэтому оно не маскируется под обычное «обновление» и всегда проходит явный договор отката.

25 Исправления безопасности и аварийные выпуски

Критическое исправление безопасности публикуется как новый выпуск с новым номером версии. Даже микроскопическое исправление не является основанием для перезаписи уже опубликованного стабильного артефакта.

- исправление проходит стандартное формирование подписанного манифеста;
- для стабильного канала требуется явная публикация и подпись;
- в описании изменений указывается характер устранённой проблемы и необходимые действия администратора;
- если исправление требует немедленной установки, это указывается в политике жизненного цикла/уведомлении безопасности;
- аварийный выпуск не отменяет резервное копирование и контроль работоспособности.

26 Обновление в закрытом контуре

Закрытый контур может иметь ограниченный исходящий доступ только к разрешённым источникам обновлений и реестрам. В таком варианте администратор заранее настраивает межсетевую политику, доверенные центры сертификации и при необходимости корпоративный прокси.

Ресурс	Назначение
studio.ncsgp.ru	Получение описания канала, манифеста и файлов выпуска.
registry.ncsgp.ru	Получение собственных контейнеров НАЙС СОФТ.
Исходные реестры сторонних образов	Получение закреплённых сторонних образов, если они не перенесены во внутренний реестр.
Корпоративный DNS/NTP	Корректное разрешение имён и время для TLS/журналирования.

Для более строгого контура допускается предварительное зеркалирование всех необходимых артефактов в разрешённую внутреннюю инфраструктуру при сохранении исходных digest и подписанных метаданных.

27 Полностью изолированное обновление

В полностью изолированном контуре обновление выполняется из подготовленного автономного комплекта. Внешний Интернет не требуется, но все необходимые артефакты должны быть доставлены внутрь контура до начала работ.

27. На подключённой доверенной площадке получить официальный подписанный выпуск и контейнерные образы.
28. Проверить подписи и контрольные суммы до переноса.
29. Сформировать автономный комплект с манифестом, подписями, пакетом среды, changelog, SBOM/лицензиями и всеми образами.
30. Перенести комплект на контролируемом носителе по правилам организации.
31. Импортировать образы во внутренний реестр/локальную контейнерную среду без изменения digest.
32. Разместить метаданные в локальном источнике обновлений, поддерживаемом установкой.
33. Выполнить ту же предварительную проверку, резервирование, переключение и health gate, что и при сетевом обновлении.

Равенство требований

Автономный режим изменяет способ доставки, но не отменяет подпись, SHA-256, резервирование, договор миграции, контроль работоспособности и транзакционный откат.

28 Сетевые требования обновления

Направление	Протокол	Назначение	Обязательность
Сервер -> studio.ncsgp.ru	HTTPS/TCP 443	Метаданные и файлы выпуска	Да для удалённого источника
Сервер -> registry.ncsgp.ru	HTTPS/TCP 443	Собственные контейнеры	Да для удалённого источника
Сервер -> сторонний registry	HTTPS/TCP 443	Сторонние закреплённые контейнеры	По составу выпуска
Сервер -> корпоративный прокси	HTTP(S)	Контролируемый выход в сеть	Если применяется
Внутри Docker-сетей	Внутренние HTTP(S)	Резервирование, мониторинг, службы продукта	Автоматически

Входящее соединение из Интернета к службе обновлений не требуется. Управление обновлением выполняется через существующий административный интерфейс ИИ Студии.

29 Дисковое пространство и кэширование

Для безопасного обновления одновременно могут потребоваться текущие контейнерные слои, новые слои, пакет среды выполнения, предыдущий runtime-каталог и новая резервная копия. Поэтому минимальное свободное место проверяется до установки.

Базовая политика генератора выпуска 1.0.0 требует не менее 20 ГиБ свободного пространства. Фактическое требование может быть увеличено манифестом конкретного выпуска. Для серверов с крупными локальными моделями необходим отдельный запас пространства под модели и контейнерные слои.

- уже загруженные совпадающие слои повторно не скачиваются;
- смена канала не удаляет контейнерный кэш;
- подготовка выпуска может быть выполнена заранее;
- предыдущий runtime сохраняется как управляемая точка возврата;
- очистка старых данных должна проводиться только по утверждённому регламенту после истечения окна отката.

30 Права доступа и административный аудит

Операции обновления относятся к административным. Пользовательская модель не должна иметь прямой возможности менять канал или инициировать переключение рабочей версии.

Операция	Требование
Просмотр версии/состояния	Доступ определяется интерфейсом; административные подробности ограничиваются ролью.
Смена канала	Только администратор; предварительные каналы требуют явного подтверждения.
Подготовка обновления	Администратор.
Установка	Администратор.
Ручной откат	Администратор + явное подтверждение «ОТКАТИТЬ».
Публикация стабильного выпуска	Только доверенная инфраструктура НАЙС СОФТ с внешним закрытым ключом подписи.

Изменения канала и ключевые административные действия фиксируются в аудите. Технические журналы не должны содержать закрытые ключи подписи, пароли реестров или секреты пользовательских интеграций.

31 Мониторинг обновлений

Служба обновлений публикует внутренние показатели состояния. В частности, контролируются возможность установки выпуска, состояние подготовленного выпуска, текущая операция, этап, ошибка и доступность отката.

Показатель	Смысл
update_installable	Выпуск прошёл подпись и основные политики и потенциально может быть установлен.
update_prepared	Пакет и контейнеры целевой версии подготовлены и проверены.
job running/phase/progress	Текущий этап длительной операции.
rollback available	Сохранена предыдущая управляемая версия для возврата.
requires_data_restore	Для отката требуется возврат данных из pre-update backup.

Мониторинг не заменяет протокол обновления: для производственной эксплуатации после установки рекомендуется фиксировать номер версии, дату, результат проверок, идентификатор резервной копии и итог приёмочного теста.

32 Ошибки и восстановление после сбоя

Этап ошибки	Ожидаемое поведение
Получение/подпись метаданных	Рабочая версия не затрагивается; установка блокируется.
Загрузка пакета/образа	Рабочая версия не затрагивается; подготовка завершается ошибкой.
Создание/проверка backup	Переключение не начинается.
Запуск новой версии	Автоматический откат к предыдущей.
Самопроверка новой версии	Автоматический откат.
Возврат данных	Режим обслуживания сохраняется до завершения; при критическом отказе требуется восстановление администратором.
Автоматический откат тоже неуспешен	Режим обслуживания оставляется включённым, фиксируется критическая ошибка, требуется техническая поддержка.

После критической ошибки запрещается выполнять непроверенное ручное удаление томов или каталогов. Сначала необходимо сохранить журналы, состояние current-release/executor-state, список контейнеров и доступные резервные копии.

33 Публикация выпуска на стороне НАЙС СОФТ

Процесс публикации отделён от клиентской установки. На сервер заказчика не передаются исходные Dockerfile и не выполняется сборка. Перед публикацией собственные образы собираются в доверенной среде НАЙС СОФТ, а сторонние образы разрешаются в точные upstream digest.

34. Подготовить доверенный набор открытых ключей службы обновлений.
35. Получить закреплённые сторонние образы.
36. Собрать собственные контейнеры НАЙС СОФТ.
37. Присвоить выпуску уникальную версию.
38. Классифицировать образы на собственные и сторонние.
39. Опубликовать только собственные build-образы в Harbor.
40. Получить точные digest всех образов.
41. Сформировать пользовательский compose только с repository@sha256.
42. Сформировать runtime-bundle без Dockerfile/build.
43. Сформировать и подписать manifest.json Ed25519.
44. Сформировать и подписать описание целевого канала.
45. Опубликовать неизменяемый каталог выпуска.
46. Атомарно обновить индекс и подписи.
47. Активировать указатель канала последним.
48. Сравнить публично доступные байты метаданных и подписей с локальными исходными файлами.

Атомарная публикация

Если ошибка происходит до активации нового указателя канала, клиент продолжает видеть предыдущий рабочий выпуск. Указатель канала публикуется только после полной доступности всех неизменяемых файлов новой версии.

34 Ротация ключа подписи выпусков

Ротация ключа подписи выполняется двухфазно, чтобы не создать ситуацию, когда сервер публикации уже использует новый ключ, а установленные клиенты ещё не доверяют его открытому ключу.

49. Сформировать новый закрытый ключ вне проекта и репозитория.
50. Добавить новый открытый ключ в доверенный набор будущего выпуска, продолжая подписывать выпуск прежним активным ключом.
51. Опубликовать и установить переходный выпуск, распространяющий доверие к новому открытому ключу.
52. После подтверждения переходного выпуска активировать новый закрытый ключ для последующих публикаций.
53. Старый открытый ключ сохранить на установленный период совместимости/льготы, затем удалить отдельным плановым выпуском.

Закрытые ключи должны храниться вне Git и артефактов релиза с ограниченными правами доступа. Файл закрытого ключа на build-host имеет ограниченные права и не копируется в контейнеры.

35 Сторонние компоненты и внешние реестры

Обновление ИИ Студии может включать переход на новые закреплённые версии сторонних контейнеров. Такие компоненты сохраняют исходное происхождение и лицензии. НАЙС СОФТ не выдаёт перепакованный сторонний образ за собственный.

- манифест содержит исходный реестр и точный digest стороннего образа;
- учётные данные Harbor НАЙС СОФТ не передаются публичным сторонним реестрам;
- изменение стороннего компонента отражается в описании выпуска и SBOM;
- для закрытых контуров сторонние образы могут быть заранее перенесены во внутренний реестр с контролем идентичности digest;
- если исходный реестр недоступен, подготовка выпуска завершается до переключения рабочей версии.

36 Совместимость и ограничения

Обновление допускается только для конфигурации, которая проходит требования манифеста и системные требования продукта. Механизм не должен автоматически обходить несовместимую архитектуру, недостаток диска, отсутствие Docker или неготовность резервного копирования.

Ограничение	Политика
Архитектура CPU	Целевая архитектура должна присутствовать в подписанном выпуске; основной профиль 1.0.0 - amd64/x86_64.
Операционная система	Должна соответствовать системным требованиям и матрице совместимости; основная ОС - НАЙС.ОС CLOUD.
Контейнерная среда	Требуется работоспособный Docker Engine/Compose; сборка на клиенте запрещена.
Свободное место	Не меньше значения манифеста.
Резервирование	Ключ восстановления настроен, служба свободна, pre-update backup проверена.
Понижение версии	Через обычный механизм обновления не выполняется.
Изменение данных	Только в соответствии с подписанным договором миграции.

37 Пошаговая процедура обновления администратором

54. Открыть «Параметры -> Обновление» и проверить текущий канал. Для продуктивной среды рекомендуется «Стабильный».
55. Нажать проверку обновлений и убедиться, что отображаются целевая версия, подтверждённая подпись и описание изменений.
56. Изучить changelog, известные ограничения, договор миграции и требования к ресурсам.
57. Выполнить предварительную проверку; устранить все обязательные ошибки.
58. При необходимости заранее выполнить подготовку/загрузку выпуска и убедиться в состоянии «обновление подготовлено».
59. Проверить наличие внешней актуальной резервной копии и сохранённого ключа восстановления в соответствии с политикой организации.
60. Выбрать окно обслуживания и предупредить пользователей, если обновление может вызвать кратковременный перерыв.
61. Запустить установку целевой версии.
62. Наблюдать за этапами создания/проверки резервной копии, включения обслуживания, запуска новой версии и контроля работоспособности.
63. После сообщения об успешной установке выполнить приёмочную проверку по разделу 38.
64. Зафиксировать выпуск в протоколе обновления.
65. Не удалять точку возврата до окончания установленного организацией периода наблюдения.

Ориентир простоя

Генератор выпуска 1.0.0 использует оценочный ориентир кратковременного переключения около 90 секунд. Это не SLA и не гарантия: фактическое время зависит от числа контейнеров, скорости диска, сети, миграций и выбранной конфигурации.

38 Критерии приёмки обновления

Проверка	Критерий
Версия	Интерфейс и состояние системы показывают ожидаемую версию.
Службы	Критические контейнеры работают без циклических перезапусков.
Вход	Администратор и тестовый пользователь могут войти в систему.
Модель	Выполняется контрольный запрос через активную модель.
RAG	Контрольный документ индексируется/находится либо ранее созданная база доступна.
Интеграции	Контрольная безопасная read-only операция выполняется как минимум для используемых критичных интеграций.
Автоматизации	Планировщик и безопасная тестовая автоматизация функционируют.
Мониторинг	Метрики и состояние системы доступны.
Резервирование	Pre-update backup имеет подтверждённый статус; новая резервная копия после обновления создаётся по регламенту.
Аудит	Ключевые административные действия отражены в журнале.
Откат	Система сообщает о доступности/условиях отката к предыдущей версии.

39 Хранение сведений и прослеживаемость

Для каждой производственной установки рекомендуется хранить историю выполненных обновлений, включающую исходную и целевую версию, дату/время, канал, хэш манифеста, идентификатор проверенного ключа подписи, идентификатор pre-update backup, результат health gate, результат самопроверки и сведения о ручном откате, если он выполнялся.

На стороне производителя для официального выпуска сохраняются исходные подписанные метаданные, changelog, runtime-bundle/hash, publication metadata, SBOM и сведения о сборке собственных образов. При наличии OCI-зеркала оно используется как дополнительный источник аудита, но не заменяет канонический HTTPS-канал обнаружения.

40 Взаимосвязь с политикой жизненного цикла

Настоящий документ описывает технический механизм обновления. Периоды выпуска исправлений, сроки поддержки веток, порядок окончания сопровождения и правила уведомления о прекращении поддержки устанавливаются отдельным документом «Политика жизненного цикла».

- стабильный канал содержит поддерживаемые публичные выпуски;
- политика жизненного цикла определяет, какие версии имеют право на исправления и обновления безопасности;
- механизм обновления не превращает неподдерживаемую версию в поддерживаемую;
- переход между основными ветками может требовать отдельного маршрута миграции;
- дата окончания поддержки не отключает установленный экземпляр автоматически, если это не предусмотрено лицензией; она определяет объём выпускаемых исправлений и поддержки.

Приложение А - Состояния процесса обновления

Состояние	Рабочая версия изменена	Что означает
Обновление доступно	Нет	Подпись и метаданные проверены, выпуск новее текущего.
Предварительная проверка пройдена	Нет	Инфраструктура удовлетворяет обязательным условиям.
Подготовка	Нет	Загружаются и проверяются пакет и контейнеры.
Обновление подготовлено	Нет	Все артефакты локально готовы к установке.
Создание резервной копии	Нет	Создаётся обязательная pre-update backup.
Режим обслуживания	Переходная фаза	Пользовательские изменения временно блокируются.
Запуск новой версии	Да, временно	Новая среда запущена, но ещё не признана успешной.
Контроль работоспособности	Да, условно	Проверяется новая версия.
Обновление установлено	Да	Новая версия зафиксирована как рабочая.
Автоматический откат	Возвращается	Новая версия не прошла проверку.
Критическая ошибка	Не определено до восстановления	Автоматический возврат тоже не завершён; обслуживание остаётся включённым.

Приложение Б - Контрольный лист обновления

№	Проверка	Отметка
1	Текущая версия и канал зафиксированы.	☐
2	Описание изменений прочитано.	☐
3	Подпись выпуска подтверждена.	☐
4	Предварительная проверка завершилась без обязательных ошибок.	☐
5	Свободное место достаточно.	☐
6	Docker Engine работоспособен.	☐
7	Ключ восстановления резервных копий сохранён вне сервера.	☐
8	Служба резервного копирования свободна.	☐
9	Подготовка выпуска выполнена (при использовании предварительной загрузки).	☐
10	Окно обслуживания согласовано.	☐
11	Установка запущена администратором.	☐
12	Pre-update backup создана и проверена.	☐
13	Новая версия прошла health gate.	☐
14	Самопроверка платформы успешна.	☐
15	Выполнен контрольный вход.	☐
16	Выполнен контрольный запрос к модели.	☐
17	Проверены критичные интеграции.	☐
18	Проверены мониторинг и журналы.	☐
19	Условия ручного отката отображаются корректно.	☐
20	Заполнен протокол обновления.	☐

Приложение В - Матрица решения об откате

Событие	Рекомендуемое действие	Данные
Health gate новой версии не пройден	Автоматический откат	По договору миграции: runtime-only либо restore pre-update backup.
Самопроверка платформы не пройдена	Автоматический откат	Аналогично договору миграции.
После успешной установки обнаружен критический дефект	Ручной откат после оценки поддержки	Перед откатом обязательна страховочная backup текущего состояния.
Некритичный дефект с обходным решением	Не откатывать автоматически; открыть обращение	Сохранить текущие данные, дождаться исправления либо планового отката.
Ошибка стороннего сервиса без дефекта ИИ Студии	Откат обычно не требуется	Диагностировать внешнюю зависимость.
Нарушение целостности данных после миграции	Остановить изменения, привлечь поддержку, оценить restore_required	Не выполнять самостоятельное удаление томов.
Ручной откат не завершён	Автоматически вернуть исходную текущую версию; при начатом restore - страховочную backup	Если возврат неуспешен, оставить обслуживание и эскалировать.

Приложение Г - Состав автономного пакета обновления

Элемент	Проверка
Описание канала/выпуска	Файл и подпись Ed25519.
manifest.json	Файл, подпись и SHA-256.
runtime-bundle.tar.gz	Файл и SHA-256.
compose.runtime.yml	Соответствует манифесту и содержит только immutable refs.
Контейнеры НАЙС СОФТ	Все digest из манифеста присутствуют.
Сторонние контейнеры	Все upstream digest из манифеста присутствуют.
changelog	Описание изменений конкретной версии.
SBOM	Состав программных компонентов выпуска.
THIRD_PARTY / лицензии	Условия сторонних компонентов.
Договор миграции	Класс изменения данных и политика отката.
Проверка носителя	Носитель прошёл процедуру контроля организации.

Приложение Д - Структура манифеста выпуска

Ниже приведён логический, а не буквальный формат. Конкретная схема может расширяться без изменения принципов безопасного обновления.

```
release:
  version: 1.0.x
  state: final
compatibility:
  architectures: [amd64]
requirements:
  minimum_free_disk_bytes: ...
  health_timeout_seconds: ...
policy:
  host_build_allowed: false
  source_compilation_allowed: false
runtime_bundle:
  url: ...
  sha256: ...
images:
  - service: ...
    runtime_ref: repository@sha256:...
migration_contract:
  data_change_class: none | backward_compatible | restore_required
  automatic_failed_update_rollback: runtime_only | restore_pre_update_backup
  manual_rollback: runtime_only | restore_pre_update_backup
```

Приложение Е - Форма протокола обновления

Поле	Значение
Организация	_____
Экземпляр/площадка	_____
Дата и время начала	_____
Исходная версия	_____
Целевая версия	_____
Канал	stable / candidate / development
Хэш манифеста	_____
Проверенный ключ подписи	_____
Pre-update backup	_____
Результат предварительной проверки	Успешно / замечания: _____
Результат health gate	Успешно / неуспешно
Результат самопроверки	Успешно / неуспешно
Откат выполнялся	Нет / автоматический / ручной
Контроль модели	_____
Контроль интеграций	_____
Отклонения	_____
Ответственный специалист	_____
Подпись/ЭП	_____

Приложение Ж - Диагностические команды и точки контроля

Штатное обновление выполняется через административный интерфейс. Команды ниже предназначены для диагностики, резервирования и работы доверенного специалиста; они не заменяют управляемую кнопку обновления.

```
# Проверка контейнеров
docker compose ps

# Журналы служб (пример)
docker compose logs --tail=200 updater

# Создание штатной резервной копии через CLI
./nicesoft.sh backup

# Аварийное восстановление из .nsbackup
./nicesoft.sh restore /путь/к/архиву.nsbackup

# Для build/release-инфраструктуры НАЙС СОФТ, не для сервера заказчика:
./nicesoft.sh release-publisher-status
./nicesoft.sh release-publish --channel stable --confirm-channel stable
```

Запрет

На продуктивном сервере заказчика не следует использовать build/rebuild как способ штатного обновления. Управляемый клиентский механизм получает подписанный final-выпуск и выполняет переключение без docker build.

Лист регистрации изменений

Изм.	Листов изменено	Листов заменено	Листов новых	Всего листов	№ документа	Подпись	Дата