Btrfs с атомарными обновлениями и мгновенным откатом через GRUB

В НАЙС.ОС используется файловая система Btrfs с полной поддержкой атомарных обновлений и мгновенного отката через меню GRUB. Все обновления устанавливаются в отдельный подтом и активируются только после перезагрузки — если что-то пойдёт не так, вы всегда можете откатиться к предыдущему состоянию всего за пару секунд. Технология базируется на возможностях Btrfs (snapshots, subvolumes), менеджере пакетов tdnf и интеграции с загрузчиком GRUB. Всё это работает автоматически и прозрачно для пользователя. Благодаря этому обновления в НАЙС.ОС безопасны, воспроизводимы и соответствуют требованиям по надёжности. Решение идеально для рабочих станций, серверов, инфраструктуры CI/CD и систем с повышенными требованиями к отказоустойчивости.

Атомарные обновления и откат состояния системы в НАЙС.ОС на базе Btrfs

Документ описывает модель обновлений, контрольные точки (snapshots) и порядок отката в НАЙС.ОС при использовании Btrfs в качестве корневой ФС.

1. Область применения

Настоящий документ устанавливает порядок выполнения обновлений и отката состояния системы в НАЙС.ОС, если корневая файловая система размещена на Btrfs и используется механизм снапшотов (snapshots) для формирования контрольных

точек перед и после транзакций обновления.

Документ ориентирован на системных администраторов, инженеров эксплуатации и разработчиков инфраструктуры. Термины и команды приведены для консоли root.

Примечание
Снапшоты Btrfs не заменяют резервное копирование. Снапшот фиксирует состояние данных внутри той же файловой системы и не защищает от отказа носителя.

2. Термины и сокращения

Термин	Определение
COW	Copy-on-Write: механизм записи «через копирование», при котором изменяемые блоки записываются в новое место, а не перезаписываются поверх.
Subvolume	Субтом Btrfs: логически независимое дерево каталогов внутри одной Btrfs, имеющее собственный идентификатор и пригодное для снапшотов/монтажа.
Snapshot	Снапшот: фиксация состояния субтома на момент времени. Может быть read-only (неизменяемым) и использоваться как контрольная точка отката.
Default subvolume	Субтом по умолчанию, из которого система загружается при монтировании Btrfs без указания параметра `subvol=` / `subvolid=`.
Атомарность обновления	Свойство процесса обновления: новая версия системы вводится в эксплуатацию целиком или не вводится вовсе, с возможностью возврата к предыдущему состоянию.

3. Проблемы классической модели обновлений

В классической модели обновления (установка пакетов непосредственно в активную файловую систему) транзакция обновления изменяет критические компоненты (библиотеки, службы, конфигурации, ядро, загрузочные артефакты) в рабочем окружении. При сбое транзакции возможны неконсистентные состояния.

3.1. Типовые сценарии отказа

- Обрыв питания или аварийная перезагрузка во время установки пакетов.
- Неполная транзакция менеджера пакетов (ошибка зависимостей, нехватка места, ошибка ввода-вывода).
- Несовместимость обновлённых библиотек с приложениями, требующая немедленного отката.
- Ошибки в конфигурации служб после обновления (включая некорректные миграции форматов конфигов).

3.2. Эксплуатационные требования

Для серверов, узлов безопасности и инфраструктуры, подлежащей регламентам эксплуатации, требуется механизм восстановления «до рабочего состояния» без загрузки из внешних носителей и без ручного восстановления пакетов. В НАЙС.ОС указанная задача решается использованием Btrfs-снапшотов в качестве контрольных точек с возможностью загрузки предыдущего состояния.

4. Btrfs как базовая технология атомарности

Btrfs является COW-файловой системой, предоставляющей встроенные механизмы субтомов и снапшотов. Снапшот создаётся быстро, поскольку фиксирует метаданные и ссылки на существующие блоки. Фактическое копирование блоков происходит только при последующих изменениях (COW).

4.1. Свойства Btrfs, используемые в НАЙС.ОС

- **Снапшоты субтомов** до и после транзакций обновления.
- **Read-only снапшоты** как неизменяемые контрольные точки (для отката и аудита).
- **Сжатие** (например, `compress=zstd`) для уменьшения дискового следа снапшотов и снижения I/O.
- **Интеграция с загрузчиком** посредством генерации пунктов меню для загрузки из снапшотов (при соответствующей конфигурации).

Примечание по модели данных
Снапшот фиксирует состояние в пределах конкретного субтома. Если критические данные размещены вне субтома (например, отдельные разделы `/boot`, `/var` или

внешние тома), откат корня не гарантирует откат этих данных.

5. Архитектура субтомов и контрольных точек в НАЙС.ОС

Для реализации отката требуется предсказуемая разметка Btrfs. Типовая схема использует отдельный корневой субтом (например, `@`), а также отдельные субтомы для изменяемых данных. Это снижает вероятность побочных эффектов при загрузке из read-only снапшота.

5.1. Рекомендуемая структура субтомов

```
# Пример (логическая модель)
@           -> /
@home       -> /home
@var        -> /var
@log        -> /var/log
@snapshots  -> /.snapshots
```

5.2. Контрольные точки обновлений

Перед обновлением создаётся снапшот «PRE». После выполнения транзакции создаётся снапшот «POST». Эксплуатационная модель предполагает: (1) обновление, (2) перезагрузку, (3) верификацию работоспособности, (4) при необходимости — загрузку из снапшота PRE.

```
# Проверка субтомов и снапшотов
btrfs subvolume list -p /
btrfs subvolume list -p /.snapshots
```

Примечание
Конкретные имена субтомов и каталог снапшотов зависят от профиля установки НАЙС.ОС. Рекомендуется использовать единую схему именования и хранить снапшоты в выделенном субтоме (например, `/.snapshots`).

6. Интеграция с загрузчиком GRUB

GRUB сам по себе не обязан отображать снапшоты как отдельные пункты меню. Публикация снапшотов в меню загрузки выполняется через генерацию конфигурации GRUB с учётом снапшотов и/или с использованием специализированного компонента, формирующего пункты меню по снапшотам.

6.1. Требования к загрузке из снапшота

- Корневой раздел Btrfs доступен загрузчику и ядру на этапе загрузки.
- Для пункта меню задан корректный `rootflags=subvol=...` или эквивалентный параметр, указывающий целевой снапшот/субтом.
- Конфигурация `fstab` и initramfs не препятствуют монтированию корня из выбранного субтома.

6.2. Диагностика пунктов меню

```
# Генерация конфигурации GRUB (команда зависит от сборки GRUB в системе)
grub-mkconfig -o /boot/grub/grub.cfg

# Просмотр параметров корня для текущей загрузки
cat /proc/cmdline
```

Примечание
Если `/boot` расположен на отдельной файловой системе (например, ext4), то снапшот корня не откатывает содержимое `/boot`. В этом случае откат «системного состояния» может требовать дополнительных процедур (например, удержания версии ядра/инициатора).

7. Порядок выполнения обновления и контрольных точек

7.1. Рекомендуемая последовательность

1. Выполнить проверку свободного места на Btrfs.
2. Создать read-only снапшот текущего состояния (PRE).
3. Выполнить транзакцию обновления (`tdnf update` или `tdnf distro-sync`).

4. Создать снапшот после обновления (POST) и зафиксировать его как контрольную точку.
5. Перезагрузить систему и выполнить проверку сервисов.

7.2. Пример команд

```
# 1) Оценка использования пространства
btrfs filesystem df /
btrfs filesystem usage /

# 2) Создание read-only снапшота (пример)
SNAP_DIR="/.snapshots"
TS="$(date -u +%Y%m%dT%H%M%SZ)"
btrfs subvolume snapshot -r / "${SNAP_DIR}/pre-${TS}"

# 3) Обновление (пример)
tdnf makecache
tdnf distro-sync -y

# 4) Создание контрольной точки после обновления
btrfs subvolume snapshot -r / "${SNAP_DIR}/post-${TS}"

# 5) Перезагрузка
reboot
```

Замечание по согласованности
Для систем с активно изменяемыми данными (БД, очереди сообщений) рекомендуется размещать такие данные в отдельных субтомах (`/var/lib/...`) и определять правила их отката отдельно. Откат корня не должен использоваться как механизм восстановления данных прикладного уровня.

8. Откат состояния

8.1. Откат через меню загрузки

При наличии пунктов меню для снапшотов оператор выбирает контрольную точку «PRE» и выполняет загрузку. После успешной загрузки выполняется верификация работоспособности. При подтверждении отката снапшот может быть назначен корнем по умолчанию.

8.2. Ручное назначение корня по умолчанию

При ручном управлении используется установка default subvolume на целевой снапшот/субтом. Параметры зависят от конкретной схемы монтирования.

```
# Пример: определить идентификатор субтома/снапшота
btrfs subvolume list -o /.snapshots

# Пример: назначить subvolid как default (укажите фактический ID)
btrfs subvolume set-default <SUBVOL_ID> /

# Перезагрузка для загрузки из назначенного default subvolume
reboot
```

8.3. Откат через snapper (если используется)

```
# Просмотр списка снапшотов
snapper -c root list

# Откат (типовая операция snapper rollback формирует новый снапшот и переключает корень)
snapper -c root rollback

# Перезагрузка
reboot
```

Примечание
Механика `snapper rollback` зависит от схемы субтомов и интеграции с загрузчиком. Рекомендуется сначала отработать процедуру на тестовом узле.

9. Контроль изменений и эксплуатационная безопасность

Снапшоты, создаваемые в режиме read-only, могут использоваться как контрольные точки для анализа изменений системы между версиями. Это обеспечивает трассируемость обновлений и снижает риск скрытой модификации системных файлов вне управляемых транзакций.

9.1. Сравнение состояния файловой системы

```
# Пример: монтирование снапшота read-only для анализа (путь примерный)
mkdir -p /mnt/snap-pre
mount -o ro,subvol=/.snapshots/pre-20250101T000000Z /dev/<BTRFS_DEV> /mnt/snap-pre

# Пример: сравнение каталога /etc
diff -ruN /mnt/snap-pre/etc /etc | less

umount /mnt/snap-pre
```

9.2. Контроль целостности (подход)

В зависимости от профиля безопасности может применяться контроль целостности (например, файловые хэши и база эталонного состояния) с возможностью сверки как активного корня, так и смонтированных снапшотов. Конкретные инструменты и политика зависят от требований эксплуатации.

10. Производительность и требования

10.1. Рекомендуемые параметры

- **Носитель**: SSD/NVMe (предпочтительно) для снижения латентности COW-операций и метаданных.
- **Память**: минимально 4 ГБ; для серверов и активных снапшотов рекомендуется 8 ГБ и более.
- **Сжатие**: `compress=zstd` для корня и типовых субтомов (при отсутствии противопоказаний профиля безопасности).

10.2. Обслуживание Btrfs

```
# Проверка состояния scrub (если запущен)
btrfs scrub status /

# Запуск scrub (пример)
btrfs scrub start -Bd /

# Просмотр использования пространства
```

```
btrfs filesystem usage /
```

Замечание
Большое количество снапшотов увеличивает объём метаданных и может усложнять управление свободным местом. Рекомендуется регламент хранения снапшотов (retention) и плановое удаление устаревших контрольных точек.

11. Ручное управление снапшотами

11.1. Создание снапшота

```
# Read-only снапшот корня
btrfs subvolume snapshot -r / /.snapshots/manual-pre-$(date -u +%Y%m%dT%H%M%SZ)

# Read-write снапшот (для временных задач)
btrfs subvolume snapshot / /.snapshots/manual-rw-$(date -u +%Y%m%dT%H%M%SZ)
```

11.2. Удаление снапшота

```
btrfs subvolume delete /.snapshots/manual-pre-20250101T000000Z
```

11.3. Контрольный перечень перед обновлением

- Подтверждено свободное место на Btrfs.
- Создан read-only снапшот «PRE».
- Зафиксирован список критичных сервисов для проверки после перезагрузки.
- Определена процедура отката (через GRUB или ручная).

12. Направления развития

Для повышения доверия к поставке и воспроизводимости восстановления в дорожной карте целесообразны следующие направления:

- Криптографическая фиксация состояния (подпись метаданных снимка/манифеста пакетов) и верификация при загрузке.
- Политики хранения снапшотов (retention) с учётом SLA и требований аудита.
- Формализация границ атомарности для узлов с отдельным `/boot` и внешними

томами.
- Интеграция «проверка после обновления» (health-check) с автоматическим возвратом к предыдущему состоянию при невыполнении критериев.

13. Заключение

Использование Btrfs-снапшотов в НАЙС.ОС позволяет формировать контрольные точки перед изменениями, выполнять проверку после обновления и, при необходимости, возвращаться к предыдущему состоянию. Для корректной эксплуатации требуется единая схема субтомов, регламент управления снапшотами и проверенная процедура загрузки/отката (в том числе с учётом размещения `/boot` и изменяемых данных).