



РУКОВОДСТВО АДМИНИСТРАТОРА

KEYVIRT-BACKUP-UI-PLUGIN

Резервное копирование и восстановление виртуальных машин
KeyVirt / oVirt 4.5+

Версия плагина	1.0.0
Дата документа	04.07.2026
Платформа	KeyVirt / oVirt Engine 4.5+, EL9 x86_64

Оглавление

1. Назначение программы	3
2. Системные требования	3
3. Порядок установки и первоначальной настройки	4
3.1. Установка	4
3.2. Проверка установки	4
3.3. Задание учётных данных	4
3.4. Создание первого задания и заполнение параметров	4
3.4.1. Ручная правка конфига задания (альтернатива UI)	5
3.5.1. Настройка почтовых уведомлений	6
3.6. Требования к NFS ресурсу	7
3.7. Бэкап через Export Domain (backup_type = ovirt_export)	7
4. Режимы работы (интерфейс)	9
4.1. Графический интерфейс (UI)	9
4.2. Командная строка (CLI) — для администратора	10
5. Инструкция по эксплуатации (ежедневные действия)	10
5.1. Мониторинг (проверка успешности)	10
5.2. Остановка, отмена и разблокировка зависших заданий	11
6. Ротация (хранение) и очистка диска	13
7. Порядок восстановления данных	14
8. Правила безопасности	14
9. Типовые неисправности и способы решения	15
Приложение. Расположение ключевых файлов	16

1. Назначение программы

Программа «keyvirt-backup-ui-plugin» предназначена для автоматического и ручного создания резервных копий виртуальных машин (ВМ) платформы KeyVirt / oVirt 4.5+ с целью их последующего восстановления в случае сбоев, атак вирусов-шифровальщиков или случайного удаления данных.

Основные функции:

- Создание полных (Full) и инкрементальных (Incremental) резервных копий дисков ВМ.
- Бэкап через штатный механизм KeyVirt Export Domain (клонирование снапшота).
- Прямой NFS-бэкап: копирование дисков с гипервизора на NFS-сервер без посредничества Engine (экономия ресурсов и трафика).
- Автоматическая ротация: хранится N инкрементальных копий на цикл, полная копия хранится до создания следующей полной.
- Шифрование учётных данных (AES-256-GCM, ключ из `/etc/machine-id`).
- Веб-интерфейс в составе KeyVirt Portal: 4 вкладки — Бэкап, Расписание, Восстановление, Учётные данные.
- Восстановление ВМ из конкретного снапшота или из произвольного файла образа (raw, qcow2, ova).

2. Системные требования

Параметр	Значение
ОС	SberOS 9 / Rocky Linux 9 / RHEL 9 / AlmaLinux 9 / CentOS Stream 9 (x86_64)
Платформа	KeyVirt / oVirt Engine 4.5 или новее
Права	root для установки; сервис работает от пользователя <code>ovirt</code>
SELinux	Enforcing (поддерживается)
Python	python3 (3.9) или python3.11 — определяется по установленному mod_wsgi
Apache	httpd с mod_wsgi (python3-mod_wsgi или python3.11-mod_wsgi)

Сеть:

- Доступ к NFS-серверу по порту 2049 (TCP/UDP)
- SSH-доступ Engine → гипервизор по порту 22 (ключ `engine_id_rsa`)
- Интернет при установке HE требуется (wheels упакованы в RPM)

Дисковое пространство на NFS:

- Минимум: объём дисков всех бэкапируемых ВМ × (nfs_keep_count + 1)
- Рекомендуется: удвоенный объём с запасом на полный бэкап при ротации

- Инкрементальные файлы хранятся как sparse raw (фактически занимают только изменённые блоки — десятки МБ вместо ГБ)

3. Порядок установки и первоначальной настройки

3.1. Установка

Выполнить от имени root на сервере KeyVirt Engine:

```
dnf install -y keyvirt-backup-ui-plugin-1.0.0-<дата>.el9.x86_64.rpm
```

Установочный скрипт (`%post`) автоматически:

- Создаёт Python virtualenv (`.venv`) с нужной версией Python
- Устанавливает Flask/requests из bundled wheels (офлайн)
- Добавляет trigger-runner в `/var/spool/cron/root`
- Устанавливает SELinux-модуль и включает boolean `httpd_use_nfs`
- Настраивает `/etc/sudoers.d/keyvirt-backup` (mount/umount для ovirt)
- Создаёт директорию `/mnt/keyvirt-nfs-backup`
- Перезагружает Apache

3.2. Проверка установки

```
systemctl status httpd  
curl -s http://localhost:8008/backup/jobs | python3 -m json.tool
```

Ожидаемый результат: JSON со списком заданий (пустой список `[]` при первой установке — это нормально).

3.3. Задание учётных данных

Через UI: KeyVirt Portal → Backup → вкладка «Учётные данные» → обновить.

Через CLI:

```
curl -s -X POST http://localhost:8008/backup/config/apply-bulk-update \  
-H 'Content-Type: application/json' \  
-d '{"username": "admin@ovirt@internal", "password": "<пароль>"}'
```

Учётные данные автоматически шифруются (AES-256-GCM) перед сохранением. Хранятся в конфигурационных файлах заданий с префиксом `ENC:`.

3.4. Создание первого задания и заполнение параметров

Через UI: вкладка «Бэкап» → кнопка «Создать задание» → ввести имя задания.

После создания задание появится в таблице «Все задания». Нажать на имя задания для перехода к настройке параметров. Обязательно заполнить:

Обязательные параметры NFS-бэкапа:

Параметр	Назначение
nfs_host	адрес NFS-сервера (например: 10.0.0.10)
nfs_path	путь экспорта (например: /backups/vms)
nfs_mount_point	точка монтирования (например: /mnt/keyvirt-nfs-backup)
vm_names	список ВМ для бэкапа (выбирается из выпадающего списка)

Важно

Учётные данные KeyVirt (логин/пароль) заполняются один раз для всех заданий через вкладку «Учётные данные» (см. п. 3.3). Не нужно вводить их отдельно в каждом задании.

3.4.1. Ручная правка конфига задания (альтернатива UI)

Конфиг каждого задания хранится в файле:

```
/usr/share/keyvirt-backup-ui-plugin/keyvirt-backup-resources/jobs/<job_id>.conf
```

Шаблон создаётся автоматически при создании задания через UI. Для ручной правки открыть файл текстовым редактором от root:

```
vi /usr/share/keyvirt-backup-ui-plugin/keyvirt-backup-resources/jobs/<job_id>.conf
```

Пример минимального конфига (nfs_direct + incremental):

```
[config]
username = admin@ovirt@internal
password = <пароль> ; будет зашифрован при следующем сохранении через UI
server = https://engine.example.com/ovirt-engine/api

backup_type = nfs_direct
backup_mode = incremental
vm_names = ["vm1", "vm2"]

nfs_host = 10.0.0.10
nfs_path = /backups
nfs_mount_point = /mnt/keyvirt-nfs-backup
nfs_mount_options = nfsvers=4
nfs_keep_count = 3

hypervisor_ssh_user = root
hypervisor_ssh_key = /etc/pki/ovirt-engine/keys/engine_id_rsa
```

```
logger_file_path = /var/log/keyvirt-backup-ui-plugin/job-<job_id>.log
logger_log_level = info
```

После ручной правки рекомендуется зашифровать учётные данные через UI: вкладка «Учётные данные» → кнопка «Зашифровать все».

3.5.1. Настройка почтовых уведомлений

Уведомления на e-mail отправляются по завершении каждого запуска задания (успех или ошибка). Письма отправляет сервер KeyVirt Engine — тот, на котором установлен плагин. SMTP-сервер должен быть доступен с Engine по сети.

Настраиваются в параметрах задания — вкладка «Бэкап» → «Параметры задания» → раздел «Уведомления», или вручную в `.conf`-файле:

```
email_notify      = yes                ; включить уведомления (yes / no)
email_to          = admin@example.com   ; адрес получателя
email_from        = keyvirt-backup@engine ; адрес отправителя (отображается в письме)
email_smtp_host   = mail.example.com    ; адрес SMTP-сервера
email_smtp_port   = 587                 ; порт (587 — STARTTLS, 465 — SSL, 25 — без шифро-
ваная)
email_smtp_tls    = yes                 ; STARTTLS (yes / no)
email_smtp_user   = backup@example.com  ; логин SMTP (если требует авторизации)
email_smtp_password = <пароль SMTP>    ; будет зашифрован автоматически
```

Пример для корпоративного сервера без авторизации (внутренняя сеть):

```
email_notify      = yes
email_to          = sysadmin@corp.local
email_from        = keyvirt-backup@engine.corp.local
email_smtp_host   = mail.corp.local
email_smtp_port   = 25
email_smtp_tls    = no
email_smtp_user   =
email_smtp_password =
```

Пример для внешнего SMTP с авторизацией:

```
email_notify      = yes
email_to          = sysadmin@example.com
email_from        = backup@example.com
email_smtp_host   = smtp.example.com
email_smtp_port   = 587
email_smtp_tls    = yes
email_smtp_user   = backup@example.com
email_smtp_password = ENC:...           ; после шифрования через UI
```

Проверка отправки вручную: запустить задание через UI → убедиться в получении письма. Если письмо не пришло — проверить доступность SMTP-сервера и правильность параметров в лог-файле задания.

3.6. Требования к NFS ресурсу

NFS-сервер должен быть доступен с двух узлов:

а) Сервер KeyVirt Engine (тот, на котором установлен плагин):

- монтирует NFS локально в `/mnt/keyvirt-nfs-backup`
- читает метаданные бэкапов (`metadata.json`, `checkpoint.json`) через API
- пишет trigger-файлы и запускает восстановление через root-cron
- необходим доступ к NFS-серверу по порту 2049 (TCP/UDP)

б) Гипервизоры (KVM-хосты кластера):

- монтируют NFS непосредственно на гипервизоре во время бэкапа
- скачивают данные дисков VM напрямую с хранилища и пишут на NFS
- монтирование выполняется автоматически командой mount через SSH
- необходим доступ к NFS-серверу по порту 2049 с каждого гипервизора

Проверка доступности NFS с гипервизора (выполнять от root на гипервизоре):

```
showmount -e <nfs_host>                # посмотреть экспорты
mount -t nfs <nfs_host>:<nfs_path> /mnt/test # тест монтирования
touch /mnt/test/.write-test && echo OK    # тест записи
umount /mnt/test
```

Настройка экспорта на NFS-сервере (`/etc/exports`). Указать разрешённые адреса: Engine и все гипервизоры кластера.

```
/backups 10.0.0.10(rw,sync,no_root_squash) # Engine
/backups 10.0.0.11(rw,sync,no_root_squash) # гипервизор 1
/backups 10.0.0.12(rw,sync,no_root_squash) # гипервизор 2
```

Обязательно

`no_root_squash` обязателен: бэкап пишется от root на Engine и от root на гипервизоре. Без этой опции записи будут запрещены.

Монтирование на Engine через `/etc/fstab` (рекомендуется для ФСТЭК-сред, где PAM может блокировать sudo mount для пользователя ovirt):

```
echo "<nfs_host>:<nfs_path> /mnt/keyvirt-nfs-backup nfs nfsvers=4,defaults 0 0" \
>> /etc/fstab
mount -a
```

3.7. Бэкап через Export Domain (backup_type = ovirt_export)

Этот режим использует стандартный механизм KeyVirt: создаёт снапшот VM, клонирует его во временную VM и экспортирует в Export Domain (NFS-шара, подключённая к KeyVirt как домен типа Export). Подходит для полных бэкапов без инкрементальных возможностей.

Предварительные требования:

- В KeyVirt должен быть добавлен Storage Domain типа «Export»
- Export Domain должен быть активирован (статус Active) в нужном Datacenter
- NFS-сервер Export Domain должен быть доступен с гипервизоров

Шаг 1. Добавить Export Domain в KeyVirt (если ещё не добавлен): KeyVirt Portal → Storage → Domains → New Domain

- Type: Export
- Storage Type: NFS
- Export Path: `<nfs_host>:<nfs_path_for_export_domain>`

Подключить домен к нужному Datacenter.

Шаг 2. Создать задание в плагине: вкладка «Бэкап» → «Создать задание» → выбрать тип «oVirt Export».

Шаг 3. Заполнить параметры задания:

Параметр	Значение
<code>server</code>	URL Engine: <code>https://<engine_host>/ovirt-engine/api</code>
<code>username</code>	логин KeyVirt (обычно <code>admin@ovirt@internal</code>)
<code>password</code>	пароль (будет зашифрован)
<code>vm_names</code>	список ВМ для бэкапа
<code>export_domain</code>	имя Export Domain в KeyVirt (например: <code>keyvirt-export</code>)

Или в `.conf`-файле:

```
[config]
backup_type = ovirt_export
server      = https://engine.example.com/ovirt-engine/api
username    = admin@ovirt@internal
password    = ENC:...
vm_names    = ["Debian_11_tst", "prod-vm"]
export_domain = keyvirt-export

logger_file_path = /var/log/keyvirt-backup-ui-plugin/job-<job_id>.log
logger_log_level = info
```

Шаг 4. Настроить расписание (вкладка «Расписание») или запустить вручную.

Как работает процесс (отображается в лог-файле задания):

1. Проверка состояния Export Domain (должен быть Active)
2. Удаление старого клона этой ВМ (если остался от предыдущего запуска)
3. Создание снимка ВМ (~2 минуты)
4. Клонирование снимка во временную ВМ `<vm_name>_BACKUP_<дата>`

5. Удаление снапшота
6. Экспорт клона в Export Domain (~2–5 минут на 10 ГБ диска)
7. Удаление временного клона

Признак успешного завершения в логе:

```
INFO: VM exported as <vm_name>_BACKUP_<дата>
INFO: Backup done for: <vm_name>
INFO: All backups done
```

Количество хранимых бэкапов в Export Domain определяется параметром `keep_exports` (по умолчанию: 1 — хранится только последний бэкап). Старые экспорты удаляются автоматически перед созданием нового.

4. Режимы работы (интерфейс)

4.1. Графический интерфейс (UI)

Доступен в KeyVirt Portal через кнопку «Backup» в боковом меню. Состоит из четырёх вкладок.

Вкладка «Бэкап»:

- «Создать задание» — настройка нового задания бэкапа
- «▶ Запустить» — ручной немедленный запуск (см. ниже)
- «■ Остановить» — прервать выполняющийся бэкап
- «Обновить» (Refresh) — обновить статус всех заданий
- Таблица «Все задания» — список всех заданий с последним статусом

Кнопка «▶ Запустить» (вкладка «Бэкап») работает в зависимости от типа выбранного задания:

Тип задания	Что происходит при нажатии «▶ Запустить»
oVirt Export Domain	Скрипт запускается немедленно в фоновом потоке веб-сервера (от пользователя ovirt). Cron не участвует. Ответ от кнопки — мгновенный.
NFS Direct / Локальный каталог	Создаётся trigger-файл <code>.run-now</code> . Скрипт запускается root-cron'ом в течение ≤1 минуты. Необходим root: бэкап пишет данные на NFS.

Вкладка «Расписание»:

- Показывает все задания, у которых задано расписание (любой тип, включая KeyVirt Export Domain)
- Кнопка «▶ Сейчас» в строке каждого задания — всегда создаёт trigger-файл `.run-now` и запускает через root-cron за ≤1 минуту, независимо от типа задания. При нажатии выводится подтверждение: «Запустить задание '...' немедленно через cron?»

Вкладка «Восстановление»:

- «Из бэкапа по заданию» — выбор задания, VM, метки времени
- «Из файла» — выбор произвольного файла (raw/qcow2/ova) на NFS-шаре или в локальном каталоге Engine

Вкладка «Учётные данные»:

- Массовое обновление логина/пароля KeyVirt во всех заданиях

4.2. Командная строка (CLI) — для администратора

Запустить бэкап немедленно (блокирующий вызов):

```
/usr/share/keyvirt-backup-ui-plugin/.venv/bin/python3 \  
/usr/share/keyvirt-backup-ui-plugin/run.py \  
-c /usr/share/keyvirt-backup-ui-plugin/keyvirt-backup-resources/jobs/<job_id>.conf
```

Создать trigger для запуска через ≤1 минуту:

```
touch /usr/share/keyvirt-backup-ui-plugin/keyvirt-backup-resources/jobs/<job_id>.run-now
```

Проверить запущенные задания:

```
ls /usr/share/keyvirt-backup-ui-plugin/keyvirt-backup-resources/jobs/*.pid 2>/dev/null
```

Посмотреть расписание:

```
cat /var/spool/cron/root
```

5. Инструкция по эксплуатации (ежедневные действия)

5.1. Мониторинг (проверка успешности)

Рекомендуется ежедневно проверять результаты ночных бэкапов.

Шаг 1. Проверить наличие ошибок в логах заданий:

```
grep -h "ERROR\|Backup failure" \  
/var/log/keyvirt-backup-ui-plugin/job-*.log | \  
grep "$(date +%Y-%m-%d)"
```

Если вывод пустой — все бэкапы за сегодня завершились без ошибок.

Шаг 2. Убедиться в наличии свежих бэкапов на NFS:

```
ls -lt /mnt/keyvirt-nfs-backup/<vm_name>/ | head -5
```

Последний каталог должен иметь дату сегодня или вчера (в зависимости от расписания).

Шаг 3. Проверить свободное место на NFS:

```
df -h /mnt/keyvirt-nfs-backup
```

Критично

Если заполнено более 90% — очистить вручную или расширить хранилище.

Признаки **успешного** бэкапа в лог-файле:

```
INFO: [incremental] Backup done for: <vm_name> (incremental)
INFO: ≡≡ NFS Direct Backup finished ≡≡
```

или для полного:

```
INFO: [incremental] Backup done for: <vm_name> (full)
INFO: ≡≡ NFS Direct Backup finished ≡≡
```

Признаки **ошибки** в лог-файле:

```
ERROR: Incremental backup failed for VM <vm_name>: ...
ERROR: Some errors occurred during the backup, please check the log file
```

5.2. Остановка, отмена и разблокировка зависших заданий

Типичное время выполнения (ориентир для обнаружения зависания):

- Инкрементальный бэкап NFS (≤100 MB изменений): 3–5 минут
- Полный бэкап NFS (диск 20 ГБ, ~2 ГБ данных): 5–10 минут
- KeyVirt Export Domain (51 ГБ VM): 10–15 минут

Сводная таблица состояния всех заданий

Вкладка «Учётные данные» → секция «Управление активными заданиями». Отображается без выбора задания; загружается автоматически при открытии вкладки.

Задание	ВМ	Заблокированные ВМ	Статус бэкапа	Действия
job-01	vm1	vm1 (transferring)	▶ Запущен	[Отменить] [Разблок...] [🔄]
job-02	vm2	—	■ Простой	[🔄]

Кнопки в строке:

- **[🔄]** — обновить строку (повторный запрос статуса)
- **[Отменить]** — послать SIGTERM процессу бэкапа (создаёт `.cancel` trigger)
- **[Разблокировать <ВМ>]** — отменить зависший бэкап в oVirt API и снять блокировку ВМ; появляется только если oVirt API сообщает об активном бэкапе для данной ВМ (фазы: ready / starting / transferring)

Глобальная кнопка «🔄 Обновить» — обновляет все строки параллельно.

Остановка процесса бэкапа

А. Через UI (для всех типов заданий): вкладка «Учётные данные» → «Управление активными заданиями» → [Отменить], или вкладка «Бэкап» → секция «Все задания» → [Отменить] (только если running).

Б. Через CLI (от root на KeyVirt Engine):

```
JOBS=/usr/share/keyvirt-backup-ui-plugin/keyvirt-backup-resources/jobs
cat $JOBS/<job_id>.pid          # получить PID
kill <PID>                      # SIGTERM — корректное завершение
ls $JOBS/*.pid 2>/dev/null      # убедиться, что PID-файл исчез
```

Или через trigger (обрабатывается root-cron'ом в течение ≤1 мин):

```
touch $JOBS/<job_id>.cancel
```

Зависшие бэкапы KeyVirt Export (ВМ осталась заблокированной)

Если процесс бэкапа типа «oVirt Export» завис или был убит «снаружи», KeyVirt может сохранить ВМ в состоянии «backup in progress» (ImageTransfer или клон ВМ с именем `<vm_name>_BACKUP_*`).

- В UI: кнопка [Разблокировать <VM>] в таблице «Управление активными заданиями» — вызывает API oVirt для отмены ImageTransfer и удаляет клон. Появляется автоматически, если API oVirt сообщает об активном бэкапе (фаза: ready / starting / initializing / transferring).
- Через CLI — найти и удалить застрявший клон:

```
# В UI oVirt: Compute → Virtual Machines → найти <vm_name>_BACKUP_*
# Удалить через oVirt UI или:
curl -sk -u 'admin@ovirt@internal:ПАРОЛЬ' \
  'https://localhost/ovirt-engine/api/vms?search=name%3D*_BACKUP_*' \
  -H 'Accept: application/json' | python3 -m json.tool | grep '"name"'
```

- Перезапустить задание — система автоматически удалит старый клон перед новым бэкапом (строка в логе: «Delete cloned VM (...) started»).

Поиск заданий oVirt Export в потоке Flask (httpd)

Задания типа «oVirt Export», запущенные кнопкой «▶ Запустить» на вкладке «Бэкап», выполняются в фоновом потоке Flask/httpd, а не через root-cron. Такие задания:

- НЕ создают `.pid`-файл в стандартном месте
- Не управляются через trigger-файлы (`.cancel`)
- Видны через API: `GET /jobs/<id>/run-status` → `{running: true, pids: "..."}`

Для принудительной остановки:

```
# Найти процессы backup_tool
ps aux | grep "backup_tool\|run.py" | grep -v grep

# Остановить (SIGTERM)
kill <PID>
```

```
# Если процесс не реагирует на SIGTERM – SIGKILL:  
kill -9 <PID>
```

После принудительной остановки oVirt Export — проверить наличие зависшего бэкапа через таблицу «Управление активными заданиями» (кнопка  для строки) и при необходимости нажать [Разблокировать <VM>].

6. Ротация (хранение) и очистка диска

Параметр `nfs_keep_count` задаёт количество инкрементальных копий в одном цикле. Полный (full) бэкап хранится до создания следующего полного.

Пример при `nfs_keep_count = 3`:

```
Цикл 1: full1 → inc1 → inc2 → inc3 → (сброс checkpoint)  
Цикл 2: full2 → inc1 → inc2 → inc3 → ...  
      ↑ при создании full2 вся цепочка full1+inc1+inc2+inc3 удаляется
```

Структура каталогов на NFS:

```
/mnt/keyvirt-nfs-backup/  
└─ <vm_name>/  
   ├── checkpoint.json           ← текущий checkpoint инкремента  
   ├── 2026-07-06_20-00-01_full/ ← полный бэкап  
   │   ├── metadata.json  
   │   └─ disk_<alias>.raw       ← sparse raw (apparent = 20 ГБ, real = 2 ГБ)  
   ├── 2026-07-07_00-00-01_incremental/ ← инкрементальный бэкап  
   │   ├── metadata.json  
   │   └─ disk_<alias>.raw       ← sparse raw (real = 10-100 МБ)  
   └─ ...
```

Важно

`ls -lh` показывает provisioned size диска (например, 20 ГБ), реальный занятый объём показывает `du -sh`. Это нормальное поведение sparse-файлов.

Сброс цикла вручную (принудить следующий бэкап быть полным):

```
rm /mnt/keyvirt-nfs-backup/<vm_name>/checkpoint.json
```

Внимание!

Если диск NFS заполнен на 95% и более — бэкап завершится ошибкой. Раз в неделю проверяйте свободное место: `df -h /mnt/keyvirt-nfs-backup`

7. Порядок восстановления данных

Важно

Восстановление выполняется только уполномоченным администратором. Не восстанавливайте данные поверх работающей ВМ.

Алгоритм «Быстрое восстановление ВМ»:

1. Открыть KeyVirt Portal → раздел Backup → вкладка «Восстановление».
2. Выбрать режим «Из бэкапа по заданию».
3. Выбрать задание → ВМ → нужную метку времени (timestamp).
4. Выбрать целевой Storage Domain для размещения восстановленных дисков.
5. Нажать «Восстановить». Статус отображается на той же вкладке.
6. После завершения — подключить диски к ВМ в интерфейсе KeyVirt и запустить ВМ.

Алгоритм «Восстановление из файла» (raw / qcow2 / ova):

1. Вкладка «Восстановление» → режим «Из файла».
2. Выбрать источник: «NFS шара» или «Локальный каталог».
3. Указать путь к файлу образа (browse по каталогам).
4. Задать имя нового диска и целевой Storage Domain.
5. Нажать «Восстановить».

Проверка хода восстановления через CLI:

```
tail -f /var/log/keyvirt-backup-ui-plugin/job-<job_id>.log
# или для восстановления из файла:
ls /var/log/keyvirt-backup-ui-plugin/restore-*.log
```

Просмотр доступных снапшотов для ВМ:

```
ls -lt /mnt/keyvirt-nfs-backup/<vm_name>/
cat /mnt/keyvirt-nfs-backup/<vm_name>/<timestamp>_full/metadata.json
```

8. Правила безопасности

- **Запрещено** вручную изменять или удалять файлы в каталогах бэкапов на NFS (каталоги `*_full`, `*_incremental`, `checkpoint.json`) — это нарушает цепочку инкрементов и сделает восстановление невозможным.
- **Запрещено** останавливать httpd (Apache) во время выполнения бэкапа — API-лог и статусы могут оказаться в неконсистентном состоянии. Перед остановкой httpd убедиться, что нет активных заданий (`*.pid` файлов).

- **Запрещено** изменять `/var/spool/cron/root` вручную — содержимое перезаписывается автоматически `trigger-runner`’ом при каждом изменении расписания через UI. Для управления расписанием использовать только UI или `.schedule` -файлы.
- **Запрещено** хранить пароли в открытом виде в `.conf` -файлах заданий. Использовать вкладку «Учётные данные» для шифрования (AES-256-GCM). Зашифрованный конфиг с одного сервера нельзя перенести на другой (ключ привязан к `/etc/machine-id`).
- **Запрещено** копировать `.conf` -файлы заданий между серверами без повторного шифрования учётных данных.
- **Запрещено** выключать сервер KeyVirt Engine во время активного бэкапа — прерванный ImageTransfer оставит диск в статусе «locked» в KeyVirt. При необходимости сначала остановить задание через UI.

9. Типовые неисправности и способы решения

Ошибка в логe	Вероятная причина	Действие оператора
<code>Permission denied: .../ <vm_name></code>	NFS не смонтирован или PAM блокирует <code>sudo mount</code> для KeyVirt	Смонтировать вручную от root: <code>mount -t nfs <host>:<path> /mnt/keyvirt-nfs-backup</code> . Добавить в <code>/etc/fstab</code> для постоянного монтирования.
<code>backup finalize wait error: ('Connection aborted.', RemoteDisconnected(...))</code>	KeyVirt Engine закрыл HTTP-соединение по таймауту. Бэкап завершается успешно.	Не требует действий. Это предупреждение, не ошибка. Убедиться, что следующая строка в логe: <code>INFO: checkpoint saved:</code>
<code>incremental count (N) reached nfs_keep_count (N) — checkpoint cleared, next run will do a full backup</code>	Штатная ротация: цикл инкрементов завершён.	Не требует действий. Следующий запуск по расписанию выполнит полный бэкап.
<code>Fault reason "Operation Failed", Fault detail "General Exception", HTTP response code 400</code>	Застрявший клон VM из предыдущего неудачного запуска.	Перезапустить задание. При повторном запуске система автоматически удалит старый клон и создаст новый.
<code>sudo: PAM account management error: Permission denied</code>	PAM полностью блокирует <code>sudo</code> для пользователя <code>ovirt</code> (ФСТЭК).	Примонтировать NFS через <code>/etc/fstab</code> (постоянное монтирование). См. п. 3.6.
<code>409 CONFLICT</code> при нажатии «▶ Сейчас» или «▶ Запустить»	Завис <code>.run-now</code> файл от предыдущей попытки.	Удалить файл: <code>rm /usr/share/.../jobs/<job_id>.run-now</code> . Повторить запуск.
API не отвечает (нет ответа от Flask API)	httpd остановлен или WSGI завершился с ошибкой.	<code>systemctl restart httpd</code> ; <code>tail -50 /var/log/httpd/error_log</code>
SELinux: httpd не читает файлы <code>metadata.json</code> с NFS	Boolean <code>httpd_use_nfs</code> выключен.	<code>setsebool -P httpd_use_nfs on</code> (устанавливается автоматически при установке RPM, но может сброситься при переустановке SELinux-политики)

A. Приложение. Расположение ключевых файлов

Путь	Назначение
<code>/usr/share/keyvirt-backup-ui-plugin/</code>	корень плагина
<code>.../keyvirt-backup-resources/jobs/</code>	задания и trigger-файлы
<code>.../keyvirt-backup-resources/jobs/<id>.conf</code>	конфиг задания
<code>.../keyvirt-backup-resources/jobs/<id>.schedule</code>	расписание задания (cron-выражение)
<code>.../keyvirt-backup-resources/jobs/<id>.pid</code>	PID активного бэкапа
<code>.../run.py</code>	trigger-runner (cron → root)
<code>.../cron-manager.sh</code>	управление crontab root через API
<code>/var/log/keyvirt-backup-ui-plugin/</code>	все логи
<code>.../keyvirt-backup-api.log</code>	лог Flask API
<code>.../job-<id>.log</code>	лог задания бэкапа
<code>.../restore-<name>.log</code>	лог восстановления из файла
<code>/var/spool/cron/root</code>	расписания (crontab root)
<code>/etc/sudoers.d/keyvirt-backup</code>	права ovirt на mount/cron-manager
<code>/mnt/keyvirt-nfs-backup/</code>	точка монтирования NFS
<code>/mnt/keyvirt-nfs-backup/<vm_name>/checkpoint.json</code>	текущий checkpoint инкремента