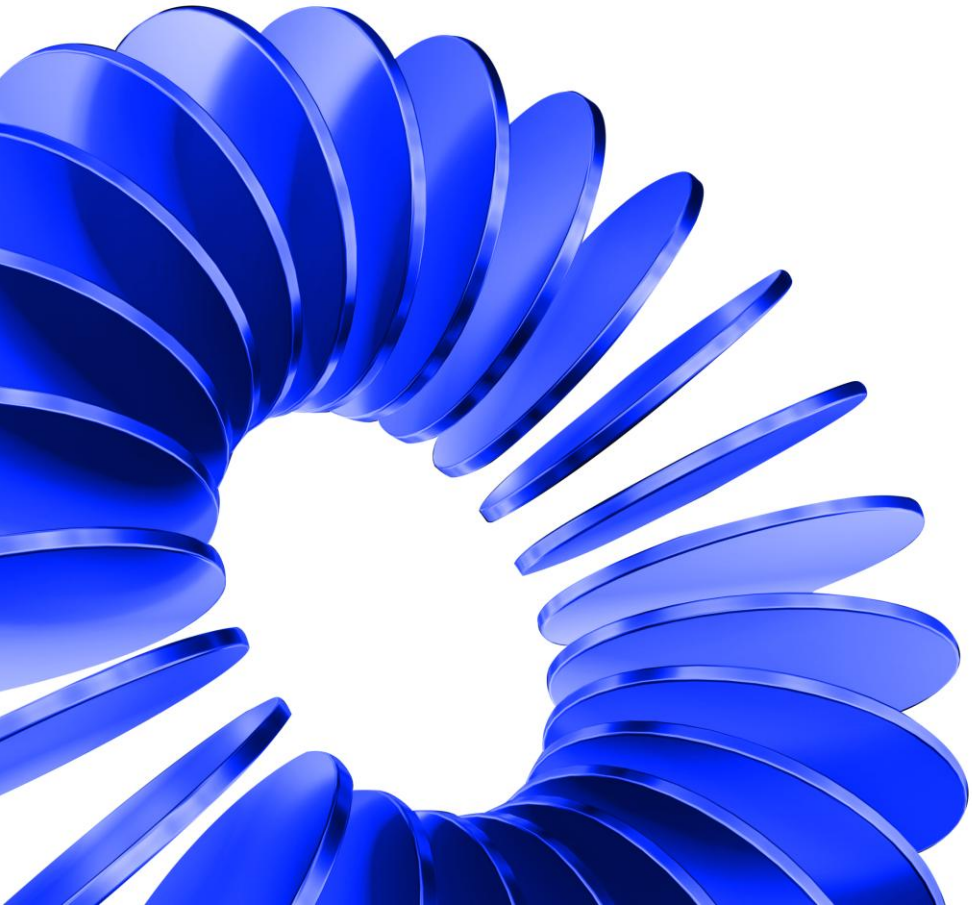
ОБЩЕСТВО С ОГРАНИЧЕННОЙ
ОТВЕТСТВЕННОСТЬЮ «РУСБИТЕХ-АСТРА»

СИСТЕМА ХРАНЕНИЯ ДАННЫХ TROK

РУКОВОДСТВО ПО УСТАНОВКЕ



Москва, 2026г.

СОДЕРЖАНИЕ

1. ОПИСАНИЕ ПРОЕКТА

ISO-образ включает программное обеспечение, предназначенное для установки и настройки компонентов TROK.

Программа, которая позволяет установить TROK, реализована на языке C++ с применением фреймворка Qt, стандартной библиотеки и библиотеки YAML-CPP для работы с конфигурационными файлами.

Основные компоненты программного обеспечения TROK разработаны на языке Go с использованием пакетов и основных библиотек protobuf, gRPC и yaml. Пользовательский интерфейс реализован с применением JavaScript и библиотеки React.

2. ОПИСАНИЕ КОМПОНЕНТОВ

Программа поддерживает установку следующих компонентов:

– Controller: центральный контроллер для управления кластером.

– Worker: системная утилита для управления хранилищем.

– WEBUI: веб-интерфейс для управления TROK.

– Harbor: компонент, необходимый для обеспечения возможности предоставления ресурсов по протоколам NFS, iSCSI или SMB.

Программа установки поддерживает три режима работы:

– Консольный режим: установка с использованием командной строки.

– Режим с ключами: запуск с указанием параметров через аргументы командной строки.

– Интерактивный режим: если программа установки запущена без ключей, пользователю будет предложено выбрать тип установки и дополнительные компоненты через графический интерфейс.

3. ТРЕБОВАНИЯ

Минимальная конфигурация аппаратных средств для корректной работы TROK представлена ниже:

– Операционная система специального назначения Astra Linux Special Edition. Конкретные поддерживаемые версии операционной системы указаны в релизных примечаниях к требуемой версии программного обеспечения. Допускается запуск программного обеспечения на операционной системе уровня защищённости Орел.

– Для установки ПО TROK пользователь должен обладать правами администратора (sudo).

Минимальные требования к аппаратному обеспечению (к серверам хранения данных, предназначенным для сохранения информации, и вычислительным узлам, которые используются для обработки этой информации):

Таблица 1 – Аппаратные требования по типам узлов

Компонент	CPU	RAM	Дисковое пространство	Примечания
Контроллер	1 чип с 4 ядрами и более Частота: от 2 ГГц	Минимальный объем: 8 ГБ Рекомендуемый объем: 64 ГБ и более Тип: DDR4, DDR5, ECC (защита от сбоев)	Минимум 2 накопителя: • Дисковое пространство для операционной системы и программного обеспечения емкостью 128 ГБ и более	2 сетевых порта с пропускной способностью 10 Гбит/с

			(SAS/SATA HDD/SSD); • Дисковое пространство для хранения данных TROK: 1 накопитель (и более) с емкостью не менее 128 ГБ (SAS/SATA HDD).	
Worker (с хранилищем)	1 чип с 4 ядрами и более Частота: от 2 ГГц	Минимальный объем: 8 ГБ Рекомендуемый объем: 64 ГБ и более Тип: DDR4, DDR5, ECC (защита от сбоев) Тип: DDR4, DDR5, ECC (защита от сбоев)	2 накопителя: • Дисковое пространство для операционной системы и программного обеспечения емкостью 128 ГБ и более (SAS/SATA HDD/SSD); • Дисковое пространство для хранения данных TROK	2 сетевых порта с пропускной способностью 10 Гбит/с

			(серверы хранения данных): 1 накопитель (и более) с емкостью не менее 1 ТБ (SAS/SATA HDD).	
Diskless-worker	1+ ядро	Минимум: 1 ГБ	Накопитель для операционной системы и программного обеспечения емкостью 128 ГБ и более (SAS/SATA HDD/SSD)	Используется для кворума или как initiator NVMe-oF
Клиент (CLI/GUI)	1 ядро	512 МБ	Накопитель для операционной системы и программного обеспечения емкостью 128 ГБ и более (SAS/SATA HDD/SSD)	Только для управления

Таблица 2 – Список используемых портов

Служба	Порт	Протокол	Назначение	Критичность

Controller	50000	TCP	REST API и CLI-клиенты	Высокая
Worker	50002	TCP	Внутренняя коммуникация с контроллером (3367 является резервным портом)	Высокая
DRBD Replication	7000 - 7999	TCP	Синхронная репликация данных	Максимальная
NVMe-oF	4420	TCP	Экспорт томов через NVMe/TCP	Опционально

Важно. Диапазон портов DRBD по умолчанию – 7000-7999.

При необходимости проверить, какие порты заняты, введите команду:

```
sudo ss -tulnp
```

- -t: TCP-порты;
- -u: UDP-порты;
- -l: Только слушающие порты (LISTEN);
- -n: Выводить номера портов (без резолвинга имён);
- -p: Показать процессы (требует sudo).

Функция будет доступна только после установки компонентов.

4. ПОДГОТОВКА ОПЕРАЦИОННОЙ СИСТЕМЫ К УСТАНОВКЕ

Выполните стандартную процедуру установки операционной системы специального назначения Astra Linux Special Edition с соблюдением требований к аппаратному обеспечению и файловым системам. Поддерживаемые версии операционной системы указаны в примечаниях к выпуску (release notes SDS TROK).

Важно. В процессе установки операционной системы, если виртуальный или физический сервер будет использоваться в качестве узла хранения данных, обязательно произведите разметку дисков, заранее выделив необходимое пространство для хранения (отдельный диск для хранения данных).

В ходе установки необходимо выбрать уровень защищённости операционной системы «Орел».

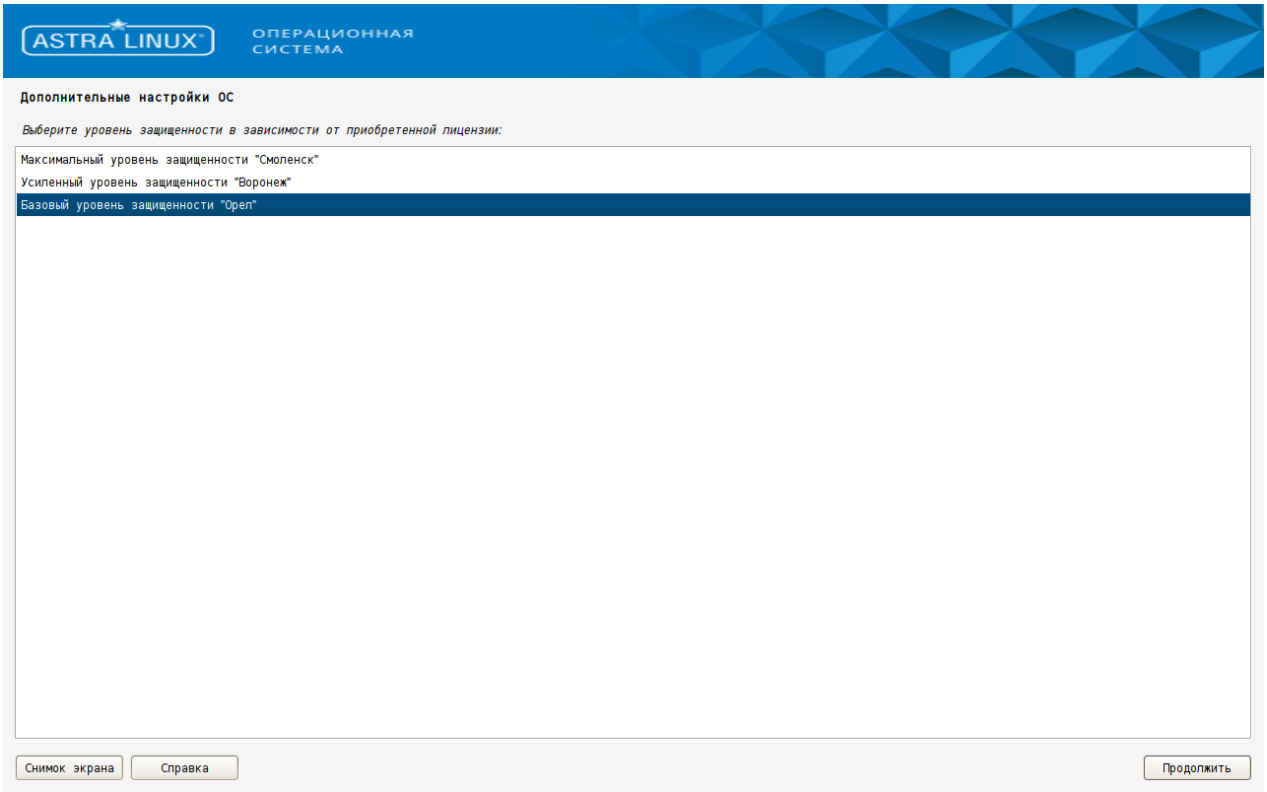


Рисунок 1 – Меню выбора уровня защищенности

В зависимости о типа iso образа, который вы используете, меню выбора выглядит немного иначе.

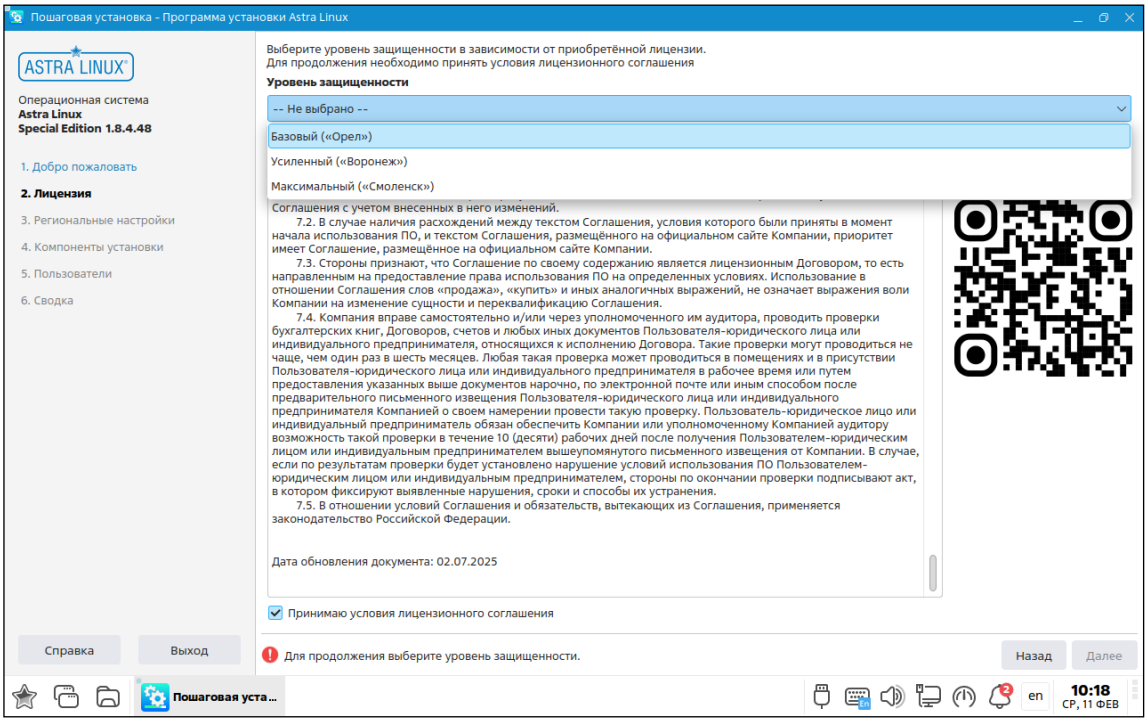


Рисунок 2 – Альтернативное меню выбора уровня защищенности

На последующих этапах установки, если вы устанавливаете Astra Linux SE версии 1.8.x система предложит выбрать одну из двух версий ядра Linux.

Рекомендуется использовать ядро linux-6.1-generic.

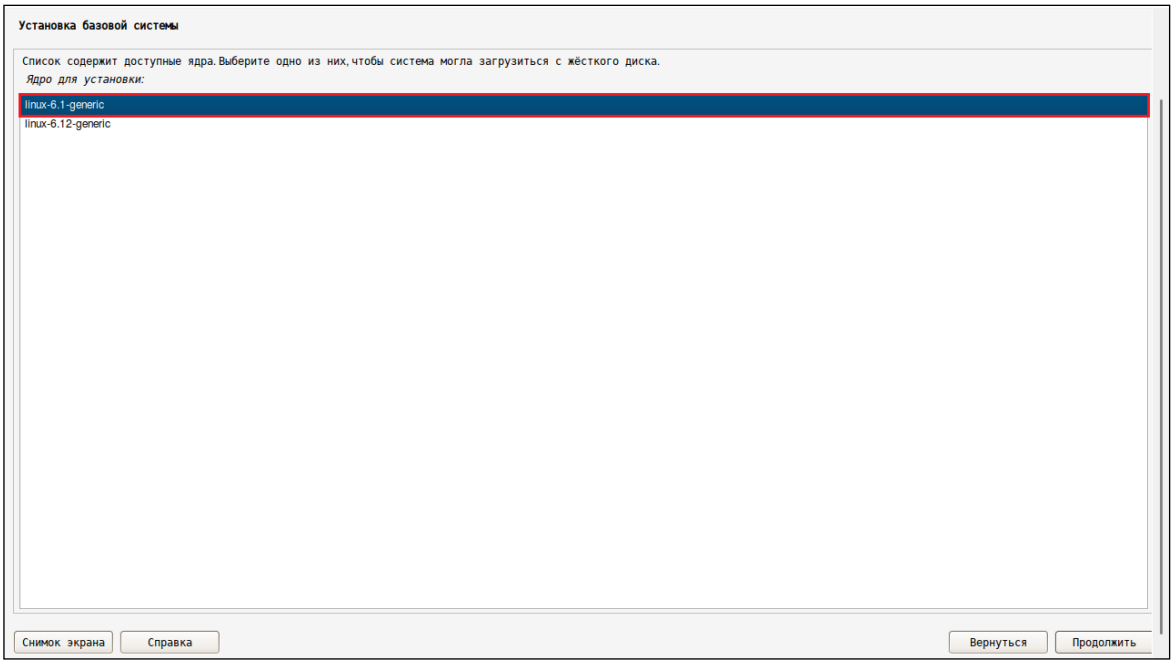


Рисунок 3 – Меню выбора ядра Linux в процессе установки Astra Linux SE 1.8.x

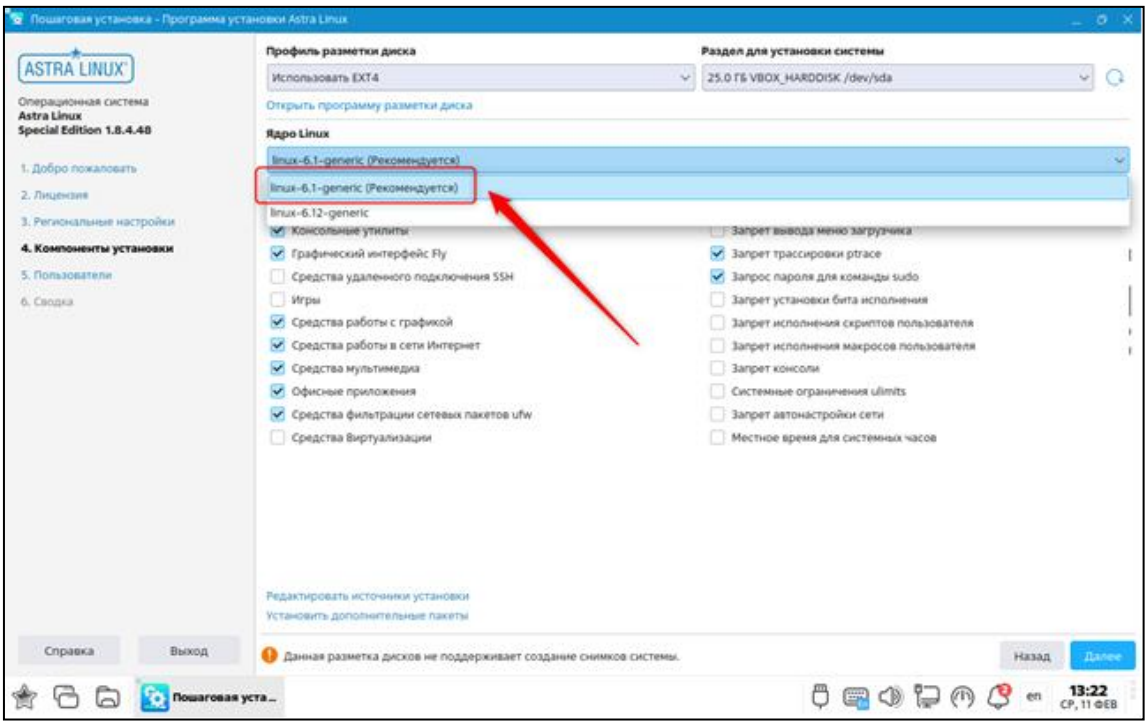


Рисунок 4 – Альтернативное меню выбора ядра Linux в процессе установки Astra Linux SE 1.8.x

Если вы устанавливаете Astra Linux SE версии 1.7.x система предложит выбрать альтернативные версии ядра Linux:

- linux-5.10-generic;
- linux-5.10-hardened;
- linux-5.15-generic;
- linux-5.15-hardened;
- linux-5.15-lowlatency;
- linux-5.4-generic;
- linux-5.4-hardened.

В этом случае допускается выбор любой версии ядра Linux, удовлетворяющей вашим задачам. Такой выбор не оказывает влияния на процедуру установки программного обеспечения.

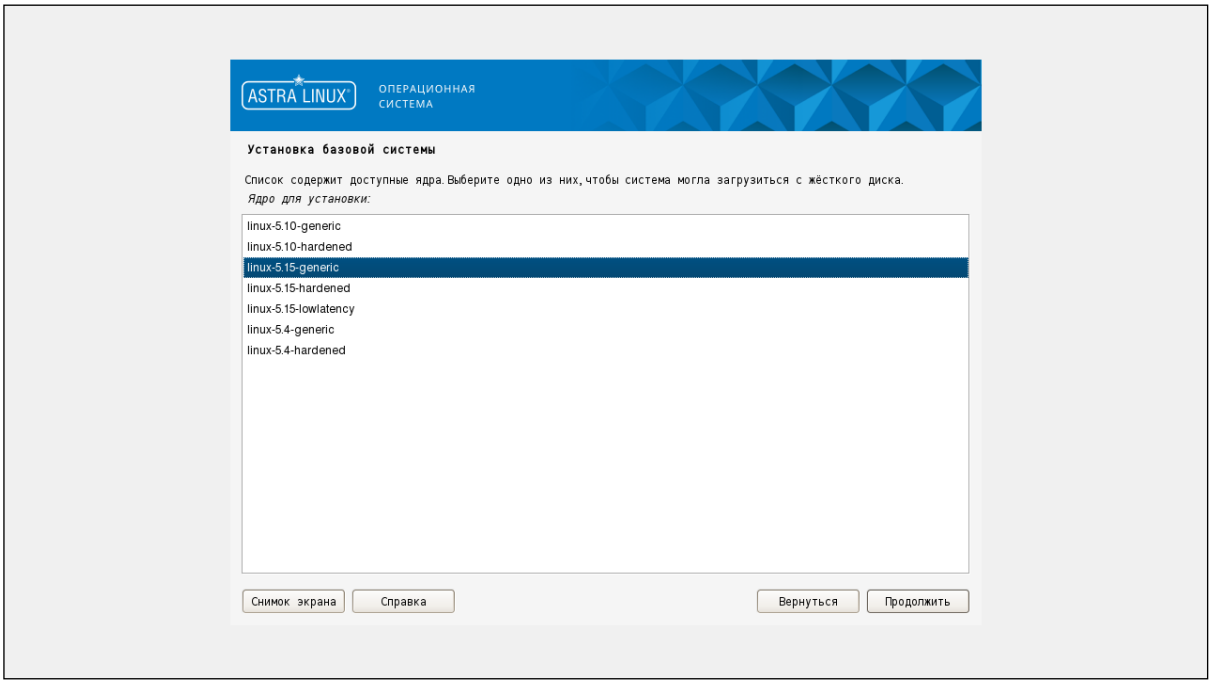


Рисунок 5 – Меню выбора ядра Linux в процессе установки Astra Linux SE 1.7.x

После завершения процесса установки операционной системы дождитесь её перезагрузки, затем выполните вход в систему, используя учётные данные администратора, указанные на этапе установки.

4.1. Подготовка операционной системы специального назначения Astra Linux Special Edition к установке ПО TROK

Откройте файл конфигурации репозиториев операционной системы (например, с помощью команды sudo vim /etc/apt/sources.list) и проанализируйте его содержимое согласно дальнейшей инструкции. Данный файл содержит перечень репозиториев программного обеспечения, используемых системой для установки и обновления пакетов. Состав и структура этого списка зависят от используемой версии операционной системы.

Для установки программного обеспечения TROK рекомендуется использовать операционные системы, настроенные на работу с репозиториями ветки «frozen» либо с репозиториями, размещёнными на установочном дистрибутиве Astra Linux Special Edition. Выбор и использование репозиториев должен осуществляться строго в соответствии

с версией установленной операционной системы, чтобы обеспечить корректную установку и функционирование программного обеспечения.

Использование репозиториев ветки «stable» нежелательно. При работе с данным типом репозиториев существует риск автоматической установки неподходящей версии ядра, что может привести к конфликтам зависимостей и некорректной работе системы.

В пунктах 4.1.1 и 4.1.2 представлены два возможных варианта настройки, которые следует выбирать в зависимости от уровня защищенности операционной системы, которую вы используете.

4.1.1. Настройка репозиториев для систем с доступом к сети Интернет (настройка для использования ветки репозиториев «frozen»)

Процесс изменения репозиториев сводится к редактированию основного файла /etc/apt/sources.list.

Перед началом изменений узнайте номер нужного обновления (например, 1.8.4) из официального бюллетеня безопасности. Также можно проверить версию с помощью ввода команды:

```
cat /etc/astra_version
```

В файле измените строки, начинающиеся с deb https://dl.astralinux.ru/astra/stable/..., заменив stable на frozen и добавив номер обновления.

Для версии 1.8.4 список репозиториев выглядит следующим образом:

Было (ветка stable):

```
# Основной репозиторий, включающий актуальное оперативное или срочное обновление

deb            https://dl.astralinux.ru/astra/stable/1.8_x86-64/main-repository/ 1.8_x86-64 main contrib non-free non-free-firmware

# Расширенный репозиторий, соответствующий актуальному оперативному обновлению

deb        https://dl.astralinux.ru/astra/stable/1.8_x86-64/extended-repository/ 1.8_x86-64 main contrib non-free non-free-firmware
```

Стало (ветка frozen, обновление 1.8.4):

```
   deb      https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.4/main-repository/
1.8_x86-64 main contrib non-free non-free-firmware

   deb  https://dl.astralinux.ru/astra/frozen/1.8_x86-64/1.8.4/extended-repository/
1.8_x86-64 main contrib non-free non-free-firmware
```

Если строка закомментирована (в начале строки ссылки на репозиторий присутствует знак «#»), то ее необходимо раскомментировать (удалите знак «#» для активации доступа к репозиториям и обеспечения обновлений).

Важно. При переходе на новое оперативное или срочное обновление номер обновления в ссылке необходимо будет также изменить вручную.

Поддерево «frozen»: репозитории Astra Linux x.7 выглядят немного иначе:

```
   # Базовый репозиторий

   deb      https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/repository-base/
1.7_x86-64 main contrib non-free

   # Расширенный репозиторий

   deb  https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/repository-extended/
1.7_x86-64 main contrib non-free

   # Базовый репозиторий срочные обновления

   deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/uu/2/repository-base/
1.7_x86-64 main contrib non-free

   # Расширенный репозиторий срочные обновления

   deb      https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.3/uu/2/repository-
extended/ 1.7_x86-64 main contrib non-free
```

Выполните обновление кэша пакетов, используемого системой для отслеживания актуальных версий доступного программного обеспечения, с помощью команды:

```
   sudo apt update
```

Команда инициирует повторное считывание и актуализацию индексов пакетов из подключённых репозиториев.

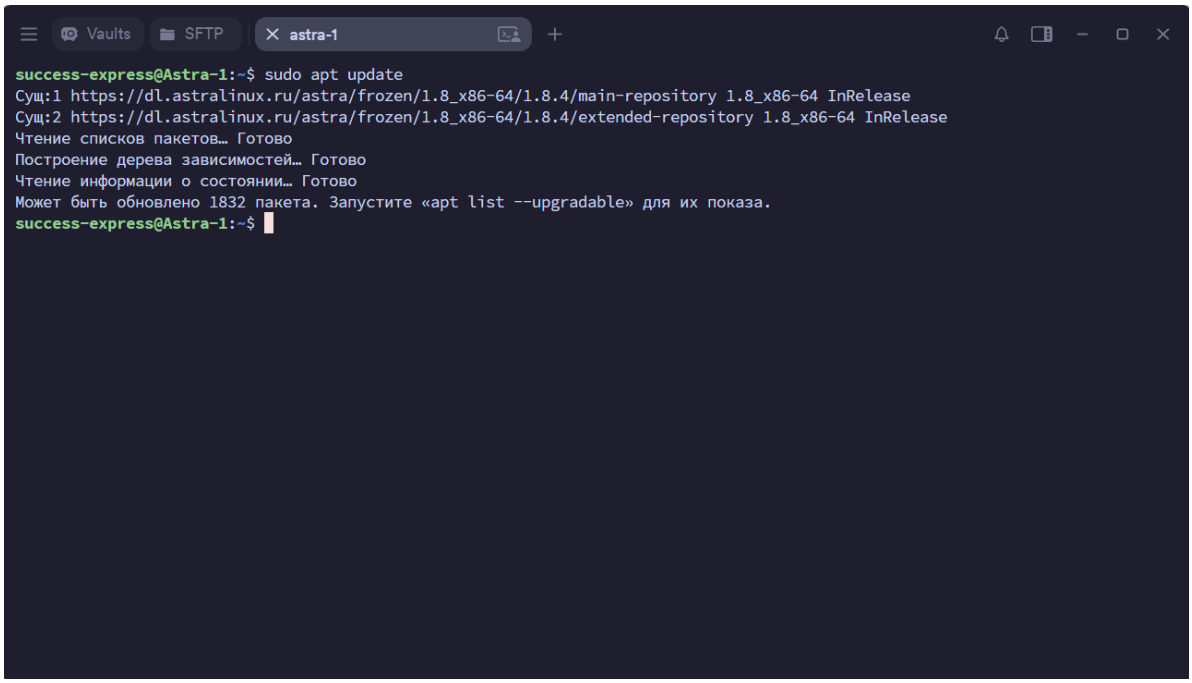


Рисунок 6 – Пример результата выполнения команды `sudo apt update`

После этого перейдите к процедуре монтирования ISO-образа программы TROK (п. 4.2).

4.1.2. Перенастройка репозиториев для автономных систем (локальные репозитории iso образа ОС)

Для получения данных из локальных репозиториев также требуется произвести редактирование файла /etc/apt/sources.list, однако в случае с локальными репозиториями необходимо внести другие изменения.

Закомментируйте строки (добавьте знак «#» перед началом строки), начинающиеся с deb https://dl.astralinux.ru/astra/stable/... или deb https://dl.astralinux.ru/astra/frozen/...

Для установки в закрытом контуре укажите источник cdrom:

Для версии 1.8.1 Astra Linux Special Edition строка со ссылкой на локальные репозитории выглядит следующим образом:

```
   deb cdrom:[OS Astra Linux 1.8.1.6  DVD]/ 1.8_x86-64 contrib main non-free non-free-firmware
```

Для версии Astra Linux Special Edition 1.7.5 следующий пример:

```
deb cdrom:[OS Astra Linux 1.7.5 1.7_x86-64 DVD]/ 1.7_x86-64 contrib main non-free
```

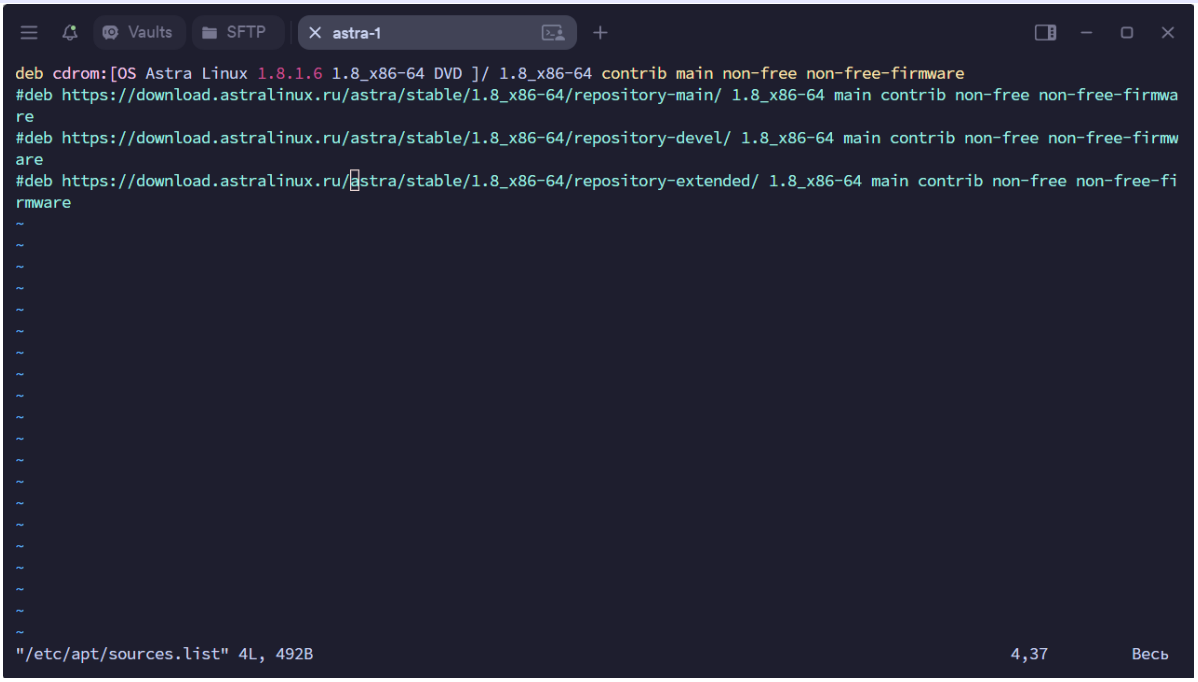


Рисунок 7 – Файл sources.list с внесенным адресом локальных репозиториев

В случае ручного монтирования установочного образа, строку в файле /etc/apt/sources.list необходимо заменить на соответствующую альтернативу. В качестве примера рассмотрим конфигурацию для версии 1.7.1 Astra Linux Special Edition:

```
deb file:///media/cdrom/ 1.7_x86-64 contrib main non-free
```

, где:

file:///media/cdrom/ – путь к каталогу, в который был смонтирован установочный образ,

1.7_x86-64 – соответствует версии и архитектуре операционной системы,

contrib, main, non-free — обозначают секции репозитория, включающие основной, дополнительный и несвободный программный продукт соответственно.

После сохранения файла /etc/apt/sources.list следует также выполнить обновление кэша пакетов, используемого системой для отслеживания актуальных версий доступного программного обеспечения, с помощью команды:

```
sudo apt update
```

4.2. Монтирование образа TROK

Перед установкой ПО TROK необходимо убедиться в наличии необходимой версии библиотеки libyaml-cpp для чтения и создания YAML-файлов. Требуемая версия этой библиотеки зависит от используемой архитектуры и версии операционной системы.

Версию библиотеки для вашей конкретной платформы можно узнать с помощью команды:

```
apt-cache search libyaml-cpp
```

Команда покажет все доступные в репозиториях пакеты, содержащие libyaml-cpp.

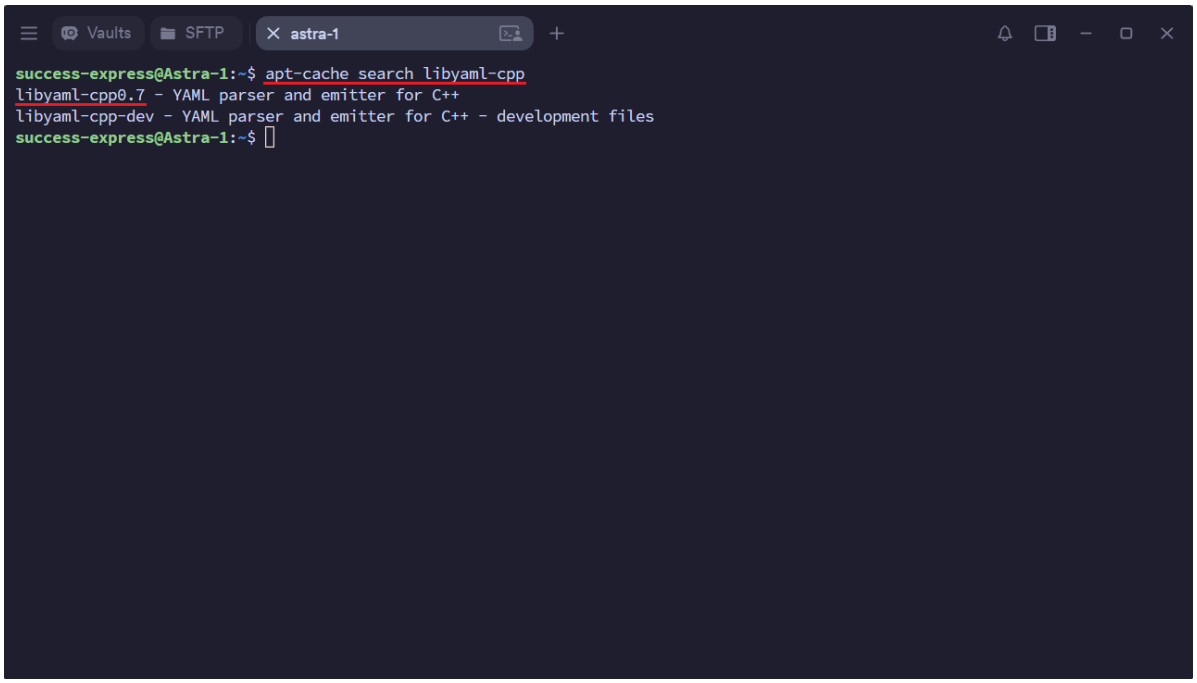


Рисунок 8 – Пример вывода команды apt-cache search libyaml-cpp

В качестве примера ниже приведена попытка установки пакета устаревшей версии. Система сообщает, что указанный пакет не найден в доступных репозиториях.

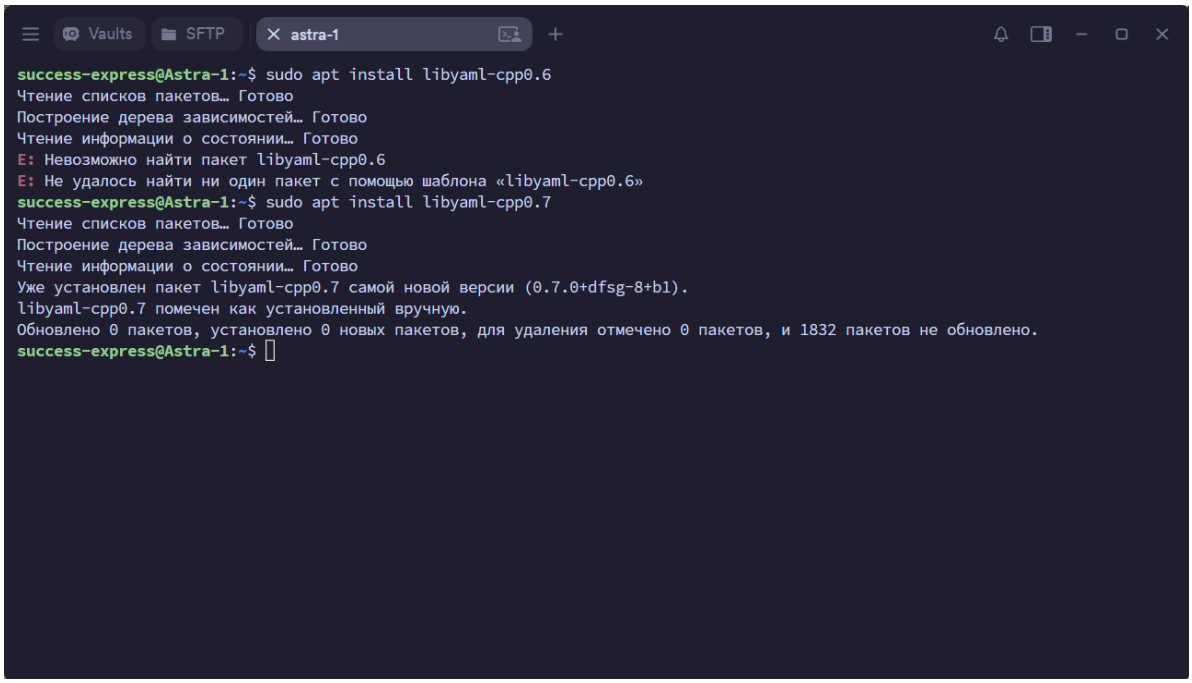


Рисунок 9 – Пример реакции системы на попытку установки неподходящего пакета libyaml-cpp

Для специальной версии TROK, предназначенной для архитектуры ARM, чаще требуется предварительная установка пакета libyaml-cpp0.6. Установку можно выполнить командой:

```
sudo apt install libyaml-cpp0.6
```

В ряде версий Astra Linux Special Edition (например, CH ALSE 1.8.3uu1 для архитектуры amd64 (x86_64)), установка или работа программного обеспечения может требовать наличие более новой версии библиотеки – libyaml-cpp0.7.

Для установки воспользуйтесь командой:

```
sudo apt install libyaml-cpp0.7
```

Для установки программного обеспечения TROK используется ISO-образ. Перед началом установки необходимо выполнить монтирование образа, чтобы получить доступ к его содержимому.

Скопируйте образ диска в удобное для вас место в операционной системе.

Создайте каталог /mnt/trok_installer используя команду:

```
sudo mkdir /mnt/trok_installer
```

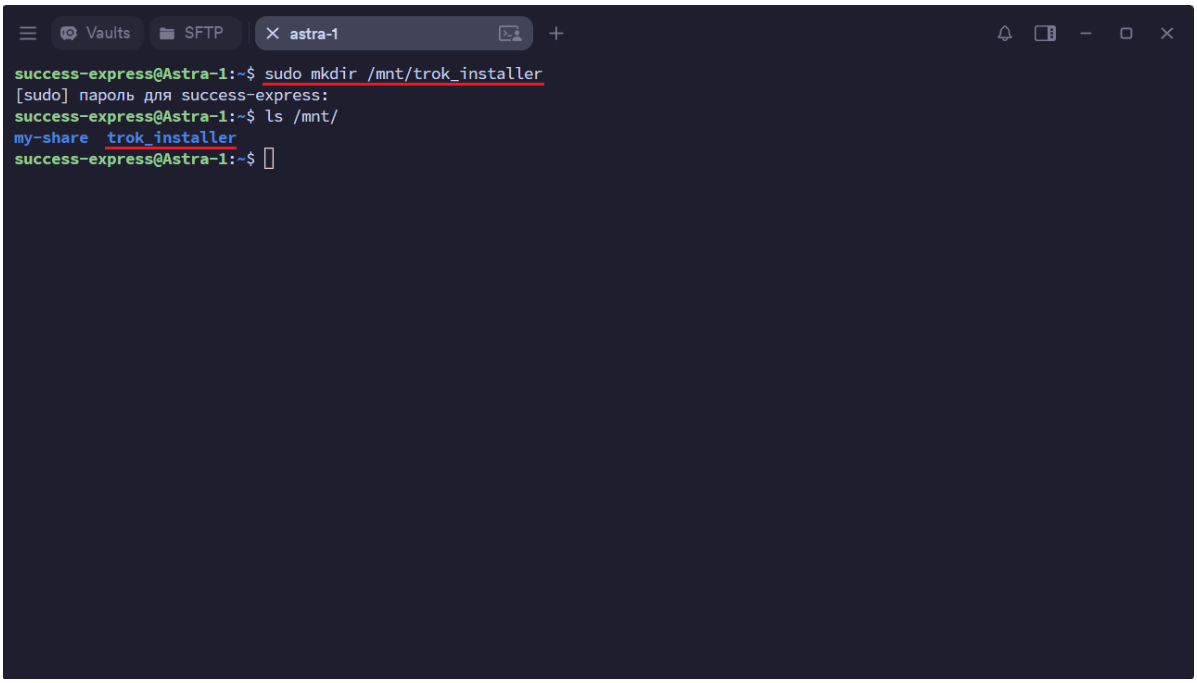


Рисунок 10 – Результат выполнения команды sudo mkdir /mnt/trok_installer

С помощью команды:

```
sudo mount -o loop <место_расположения_ISO-образа> /mnt/trok_installer
```

производится монтирование ISO-файла в созданный каталог с использованием опции loop, позволяющей обращаться с образом как с виртуальным блочным устройством.

(например, `sudo mount -o loop /home/admin/ТРОК-1.0-Астра_amd64.iso /mnt/trok_installer`)

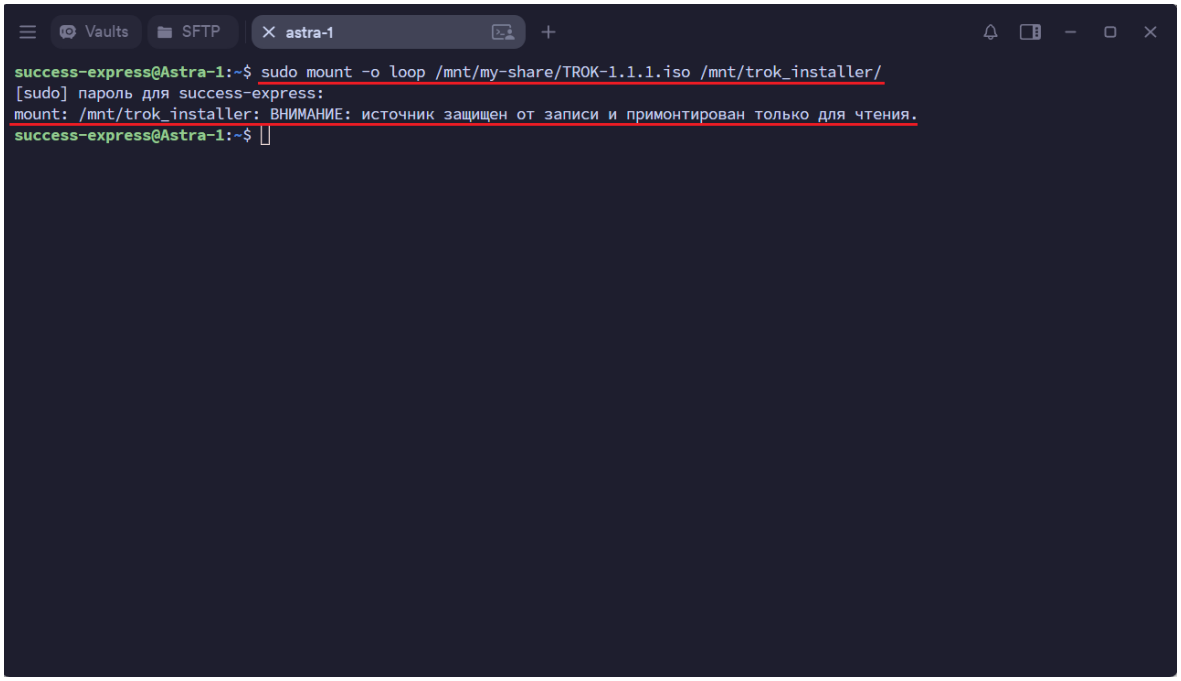


Рисунок 11 – Уведомление системы об успешном завершении процедуры монтирования образа

Для проверки успешного монтирования и просмотра содержимого образа используйте команду:

```
ls /mnt/trok_installer
```

Важно! Перед началом процедуры установки перейдите в каталог, в который был смонтирован образ:

```
cd /mnt/trok_installer/installer
```

Далее все команды установки предполагают, что текущая рабочая директория – /mnt/trok_installer/installer.

Если по каким–то причинам вы выполняете команды из другого каталога, замените префикс «./» в командах вида sudo ./trok_installer на соответствующий абсолютный или относительный путь к исполняемому файлу, например: sudo /mnt/trok_installer/installer/trok_installer.

5. КОНФИГУРАЦИЯ

Программа для установки использует YAML-конфигурацию настройки репозиториев и пакетов. Файл расположен в iso образе TROK и доступен после выполнения процедуры монтирования образа.

Пример файла config.yaml:

```
   options:
     archive: TROK-1.1.tar.gz
     repository_entry: deb [arch=amd64 signed-by=/usr/share/keyrings/trok-archive-
keyring.gpg] file:///opt/TROK/SDS/apt-repo stable main
     repository_config: /etc/apt/sources.list.d/trok_repo.list
     keyring_file: trok-archive-keyring.gpg
     eula_file: EULA_SDS_Trok.txt
     eula_path: /usr/share/doc/trok
     log: /var/log/trok-installer
     autoinstall_path: /usr/sbin/trok-drbd-autoinstall
   packages:
     worker:
     - name: trok-worker-meta
       version: ''
     controller:
     - name: trok-controller-meta
       version: ''
     harbor:
     - name: trok-harbor-meta
       version: ''
     webui:
     - name: nginx
       version: ''
     - name: trok-webui
       version: ''
```

При необходимости Вы можете изменить путь к директории для сохранения лог-файлов заранее, используя указанный файл конфигурации, указав в параметре log нужные значения.

В описании устанавливаемых пакетов в случае, если значение version остается пустым, программа установки автоматически осуществляет загрузку самой последней версии пакета. В случае явного указания конкретного значения параметра version будет загружена и установлена именно эта версия.

6. УСТАНОВКА С ИСПОЛЬЗОВАНИЕМ КОМАНДНОЙ СТРОКИ

Для установки программного обеспечения TROK в папке installer предоставляется 2 программы установки на выбор:

– установка с использованием командной строки;

– установка с применением графического интерфейса пользователя.

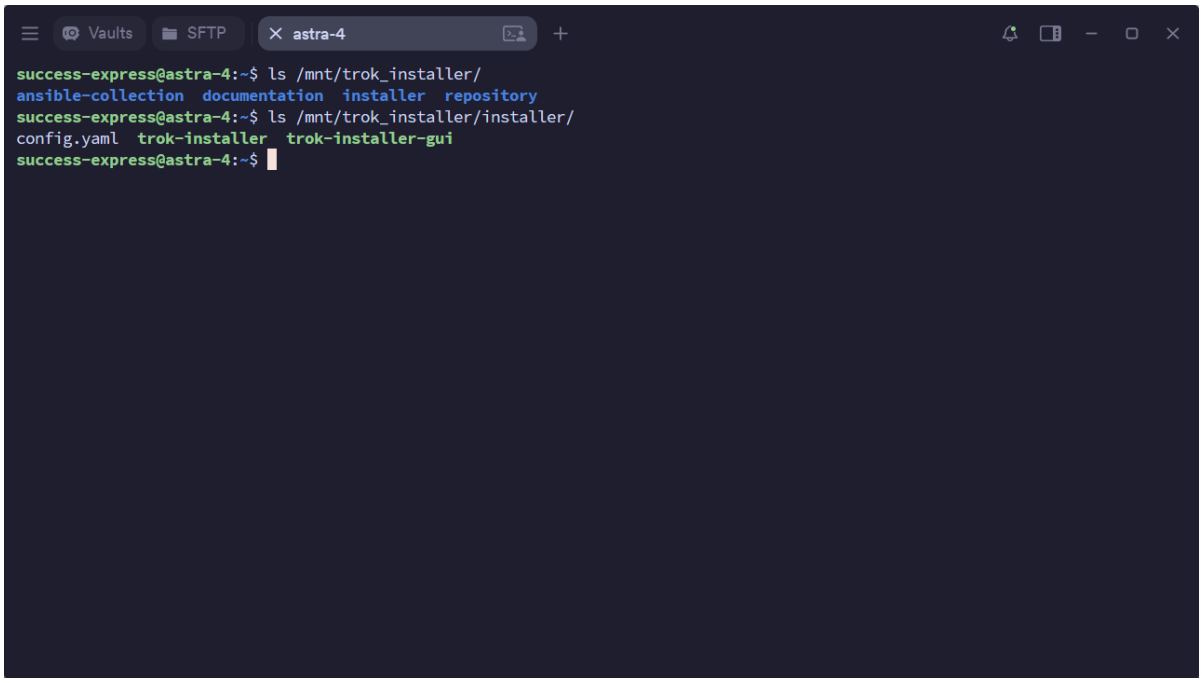


Рисунок 12 – Пример размещения установочных файлов в структуре iso образа

Для установки с использованием консоли введите команду:

```
sudo ./trok-installer
```

Далее в руководстве все команды в примерах будут указаны на примере версии Astra Linux Special Edition 1.8.

Программа поддерживает следующие аргументы командной строки:

Таблица 3 – Аргументы, используемые при установке в режиме с ключами

Аргумент	Описание
--worker	Установка узла типа Worker
--controller	Установка узла типа Controller
--combined	Установка узла типа Combined (Worker + Controller)

--webui	Установка WEBUI и nginx
--log <директория>	Указание пути к директории для сохранения лог-файлов (по умолчанию: /var/log/trok-installer)
--console	Запуск с консольным интерфейсом
--harbor	Установка Harbor

Пример использования:

– Установка с указанием пути к директории для сохранения лог-файлов:

```
./trok-installer --log <директория>
```

Важно. В процессе ввода пути к директории для сохранения логов вы можете использовать как абсолютный путь (например, /home/user/my_logs/trok), так и относительный (например, ../trok-installer/logs).

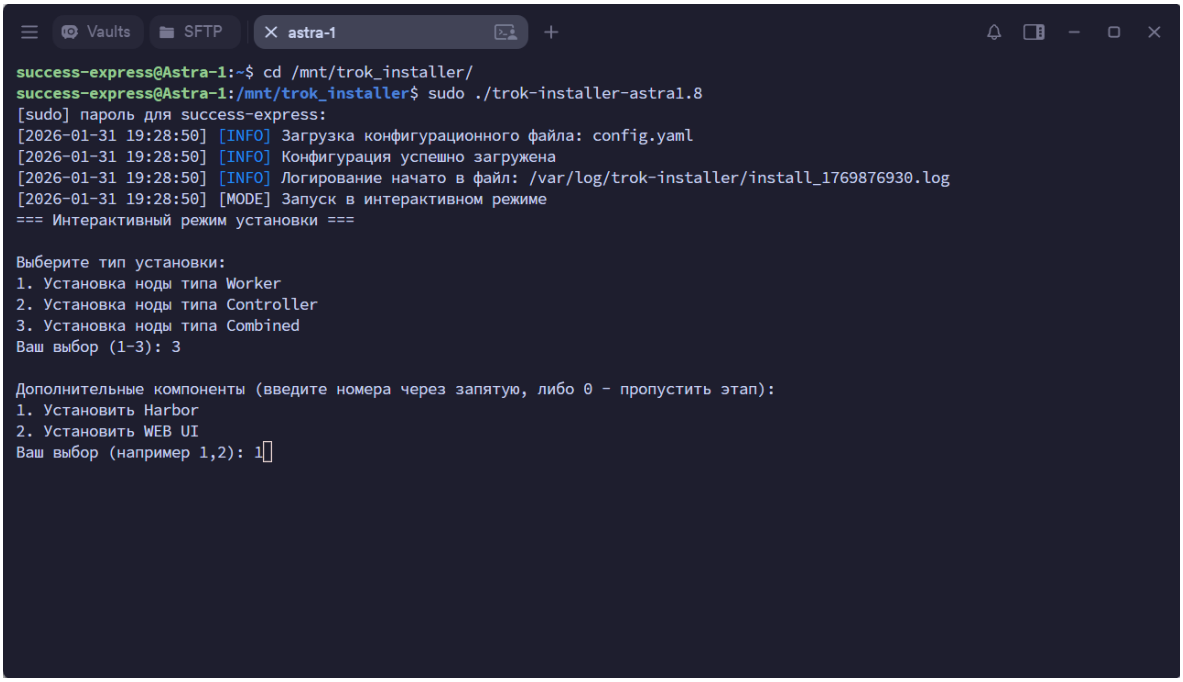


Рисунок 13 – Начало установки программного обеспечения в режиме командной строки

После выполнения установки рекомендуется перезагрузить систему.

В случае возникновения проблем с установкой DRBD смотрите пункт 8.

7. УСТАНОВКА В ИНТЕРАКТИВНОМ РЕЖИМЕ (GUI)

В случае использования графического интерфейса в операционной системе и запуска установочного файла trok-installer-gui посредством данного интерфейса автоматически инициируется графический режим конфигурирования.

Важно. Для запуска необходимо находиться в директории, в которой располагается установочный файл.

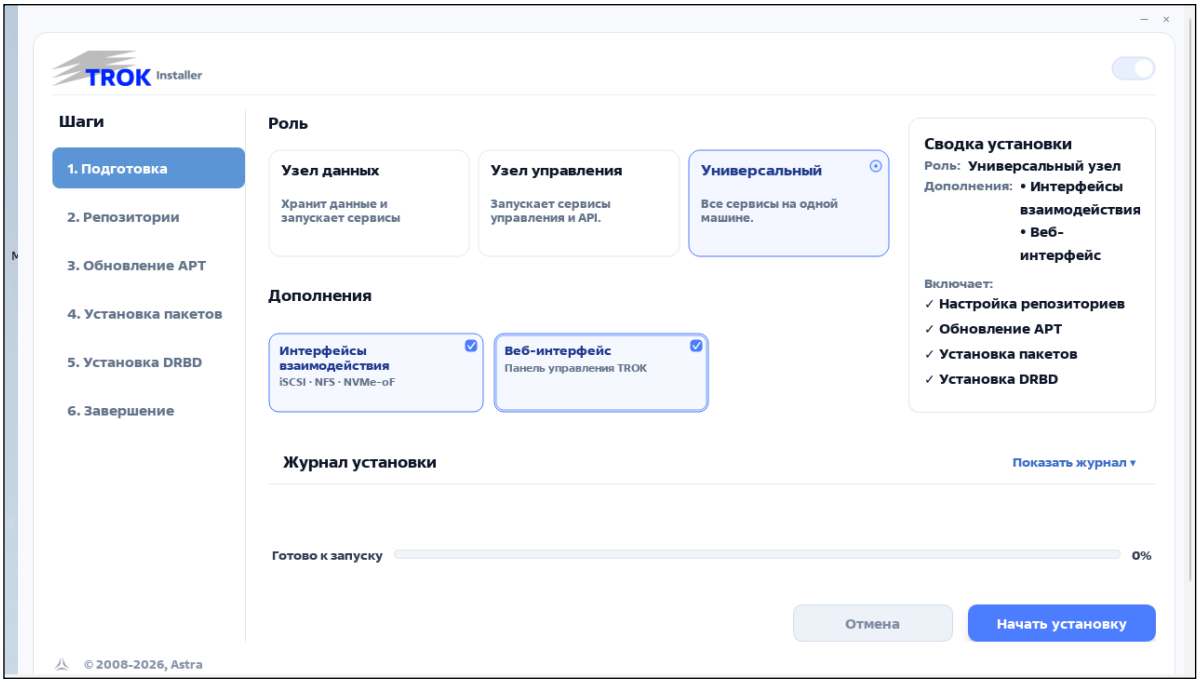


Рисунок 14 – Установка в интерактивном режиме

После запуска программы установки пользователю предоставляется выбор между следующими архитектурными конфигурациями:

– Определение топологии узла:

– worker: Установка узла категории worker (исполнительный компонент).

– combined: Комбинированная установка (worker + controller).

– controller: Развертывание управляющего узла (control plane).

– Компоненты, автоматически устанавливающиеся при установке trok-controller:

– trok-auth: обеспечивает централизованную аутентификацию и авторизацию для всех компонентов TROK через gRPC API, управляя OAuth2

клиентами (приложениями) и выдачей JSON Web Tokens (JWT) в соответствии со спецификацией OpenID.

– trok-cp-endpoint: обеспечивает REST API TROK для управления и мониторинга DRBD - ресурсов. Сервис выступает единой точкой входа для всех операций управления кластером, обеспечивая взаимодействие между пользователями/клиентами и бэкендом контроллера.

– Технические особенности реализации:

– harbor: готовый программный модуль, объединяющий службу автоматизации DRBD (drbd-reactor), набор сценариев управления ресурсами (resource-agents) и поддержку сервисов доступа (iSCSI, NFS, SMB), который поставляется в виде отдельного мета-пакета для установки на узлах хранения, а также входит в состав контроллера.

– WEBUI: Веб-интерфейс управления (включая сервис nginx).

Перед началом установки ознакомьтесь с лицензионным соглашением и нажмите «Я принимаю условия соглашения», если готовы продолжать установку программы.

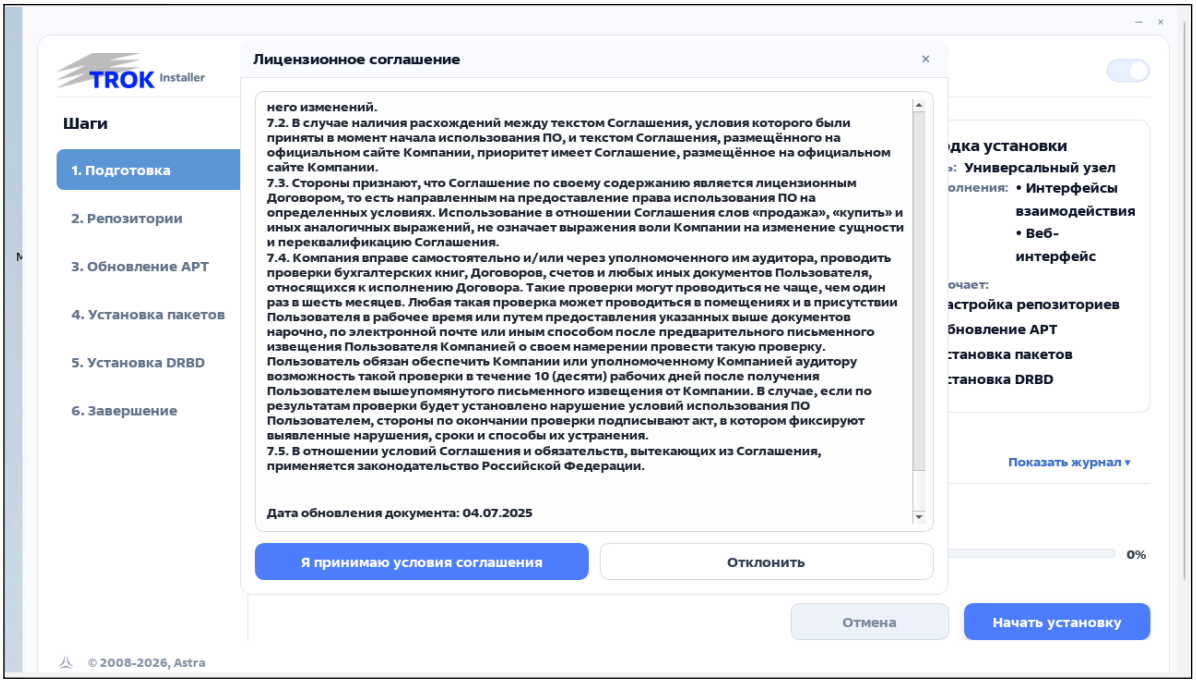


Рисунок 15 – Лицензионное соглашение

Все дальнейшие шаги установки производятся автоматически. По завершении установки отображается сообщение об успешном завершении процесса.

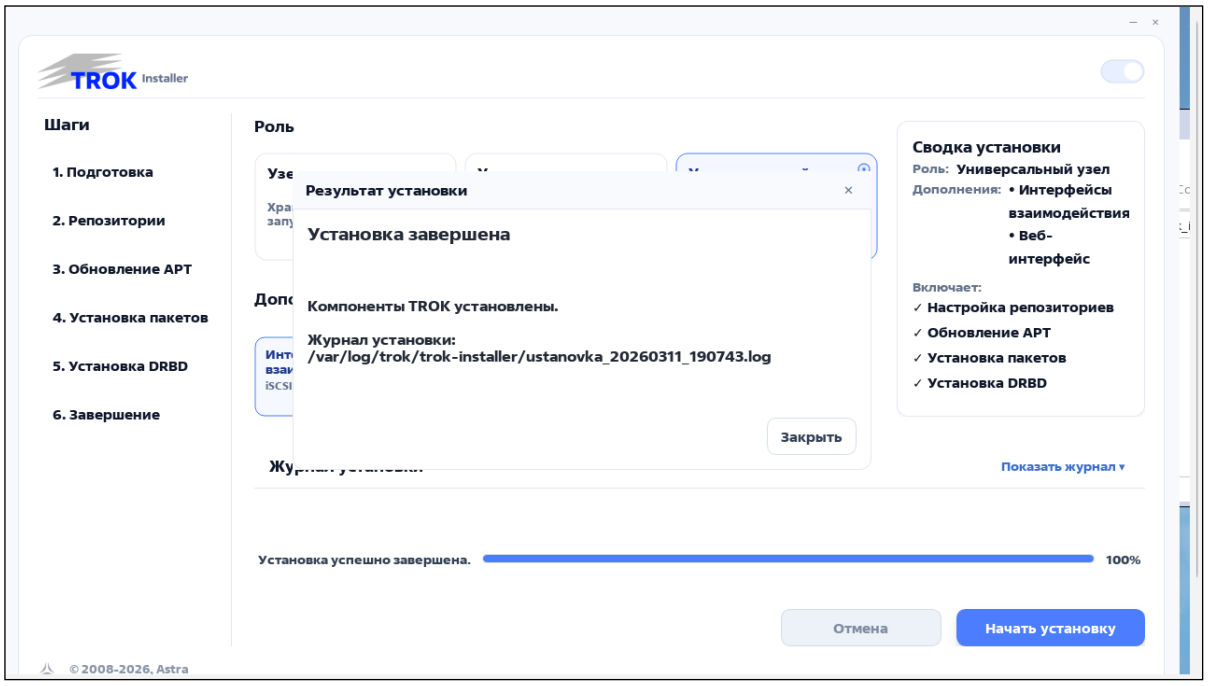


Рисунок 16 – Уведомление о завершении установки

После выполнения установки рекомендуется перезагрузить систему.

В случае возникновения проблем с установкой DRBD смотрите пункт 8.

8. УСТАНОВКА С ИСПОЛЬЗОВАНИЕМ ПАКЕТОВ

В зависимости от типа узла может быть установлены различные категории пакетов программ. Ниже приведен вписок доступных пакетов.

Таблица 4 – Пакеты и их назначение

Пакет	Назначение	На каких узлах устанавливать
trok-controller-meta	Установка модуля, предназначенного для реализации функций управления данными и координации узлов в составе кластера хранения.	На узлах, выполняющих роль controller.
trok-worker-meta	Установка модуля обеспечивающего непосредственное хранение данных, локальное управление и обработку команд, поступающих от управляющего узла (controller).	На узлах, выполняющих роль worker.
trok-combined-meta	Установка двух модулей единовременно: управляющего (controller) и модуля хранения данных (worker).	На совмещённых (комбинированных) узлах с функциями controller и worker.
trok-webui (+ nginx)	Установка компонента, предоставляющего веб-интерфейс управления с	На комбинированных узлах или узлах controller.

	использованием веб-сервера nginx.	
trok-harbor-meta	Установка модуля, реализующего API и функцию шлюза для интеграции с внешними системами.	На комбинированных узлах или узлах controller.

Перед началом установки программного обеспечения с использованием пакетов уточните, какую роль будет играть узел (controller, worker, combined).

Найдите архив с пакетами для установки. В смонтированной директории /mnt/trok_installer в папке repository находится файл TROK-<версия>.tar.gz.

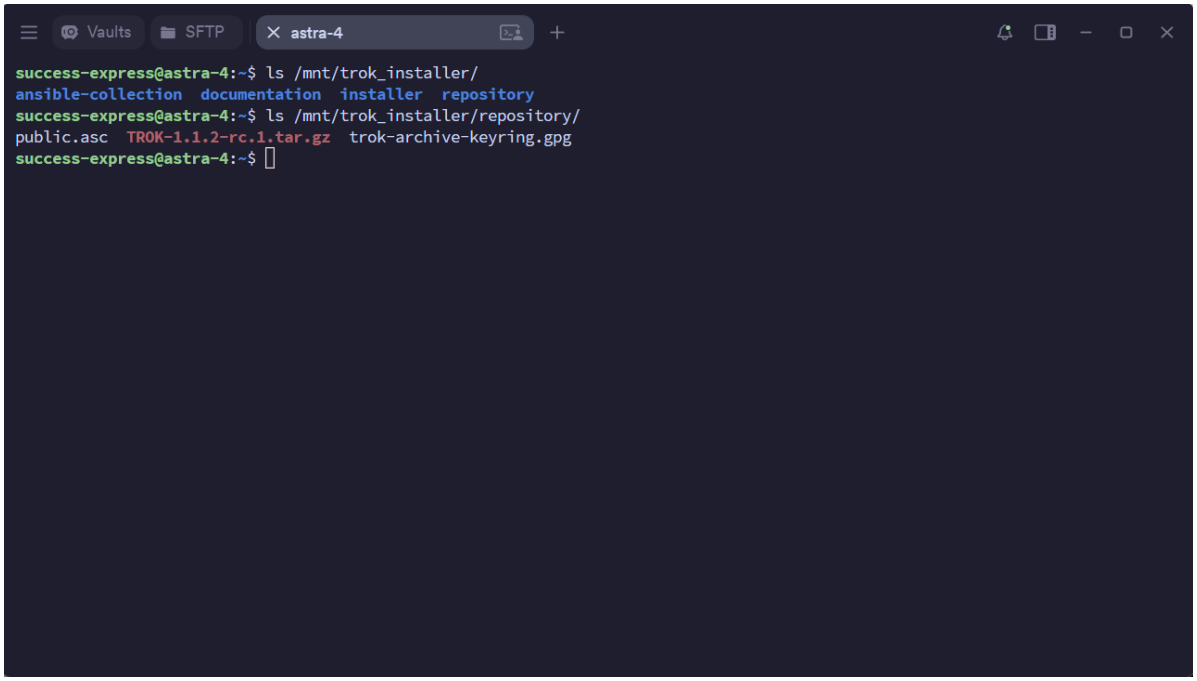


Рисунок 17 – Расположение архива, содержащего пакеты для установки TROK

Выберите место для распаковки архива и создайте в нем папку, например trok-pack. Из-за того, что в смонтированную директорию нельзя вносить изменения, выберите другую папку, например /opt (используется для сторонних программ).

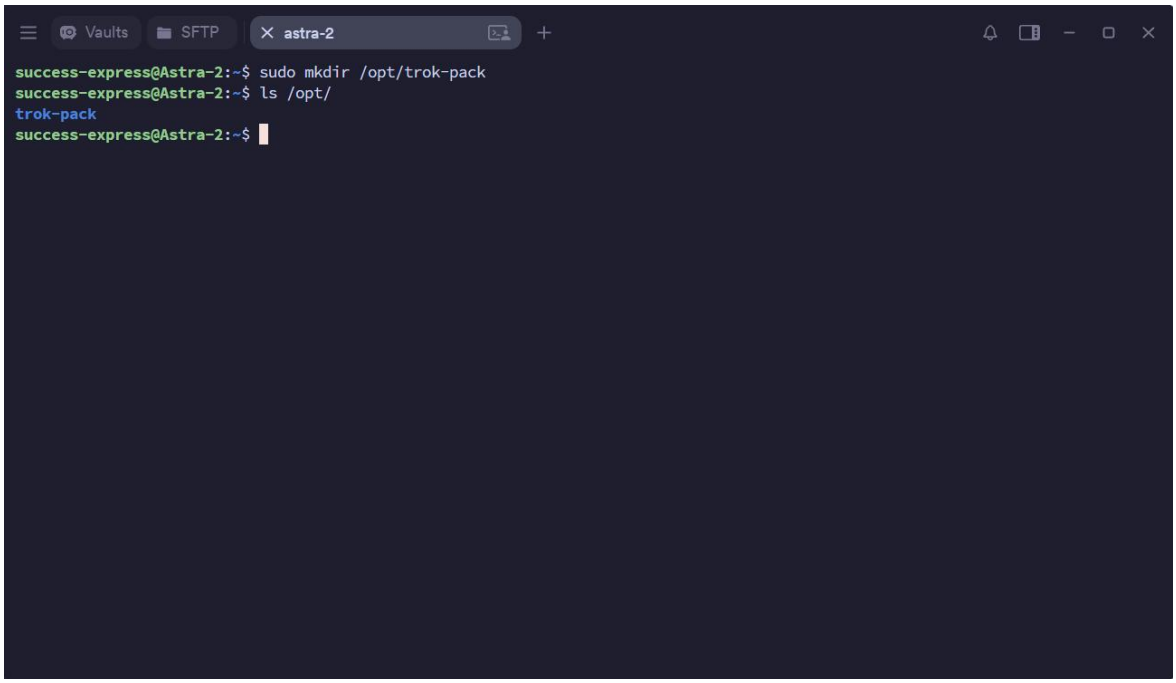


Рисунок 18 – Создание папки для последующего размещения пакетов TROK

Выполните команду для распаковки архива в созданную папку:

```
sudo tar -xzf /mnt/trok_installer/repository/<имя_архива> --overwrite -C /opt/trok-pack/
```

Ключ --overwrite не всегда обязателен, но допустим, если файл уже существует.

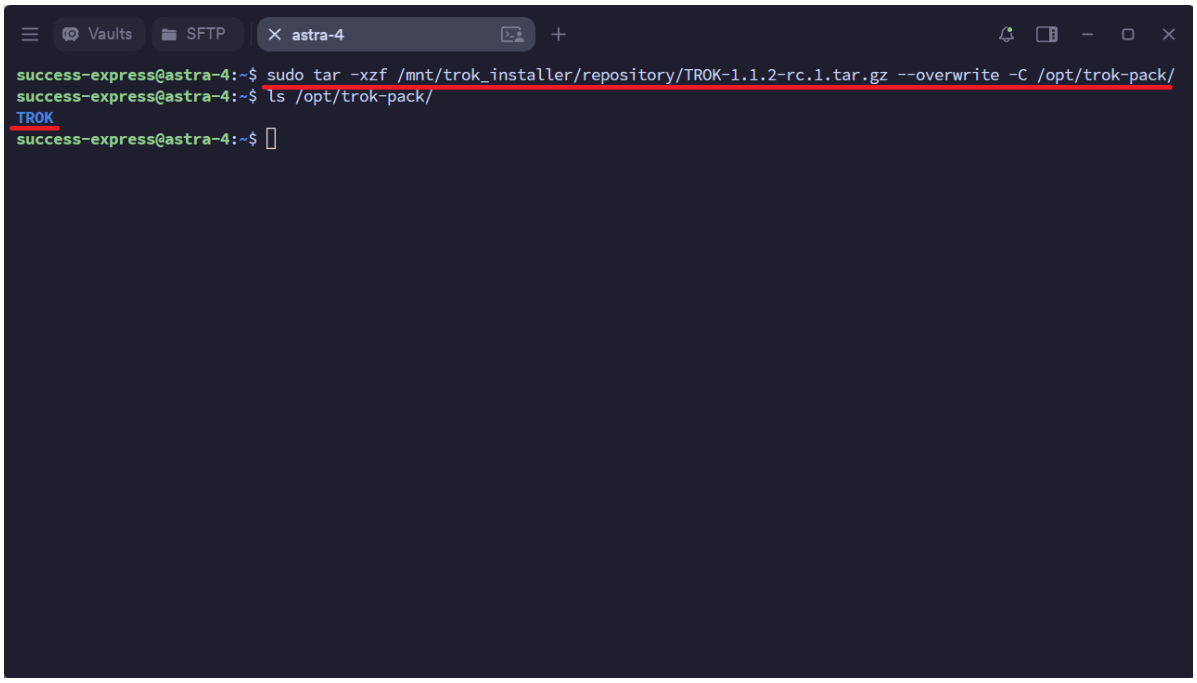


Рисунок 19 – Результат выполнения команды распаковки

Создайте отдельный файл репозитория с помощью команды:

```
sudo vim /etc/apt/sources.list.d/trok_repo.list
```

Добавьте в созданный файл строку репозитория:

```
deb [arch=amd64 signed-by=/usr/share/keyrings/trok-archive-keyring.gpg] file:///mnt/trok_installer/SDS/apt-repo stable main
```

Рисунок 20 – Файл репозиториев trok_repo.list после внесения изменений

В файлах смонтированного образа найдите файл trok-archive-keyring.gpg и копируйте его в систему.

Для начала убедитесь, что в директории /usr/share/ есть папка, keyrings. Если ее нет, создайте вручную:

```
sudo mkdir -p /usr/share/keyrings
```

Флаг -p позволяет создать вложенные директории, если их ещё нет, и не выдаёт ошибку, если директория уже существует.

Затем выполните команду копирования:

```
sudo cp /mnt/trok_installer/trok-archive-keyring.gpg /usr/share/keyrings/
```

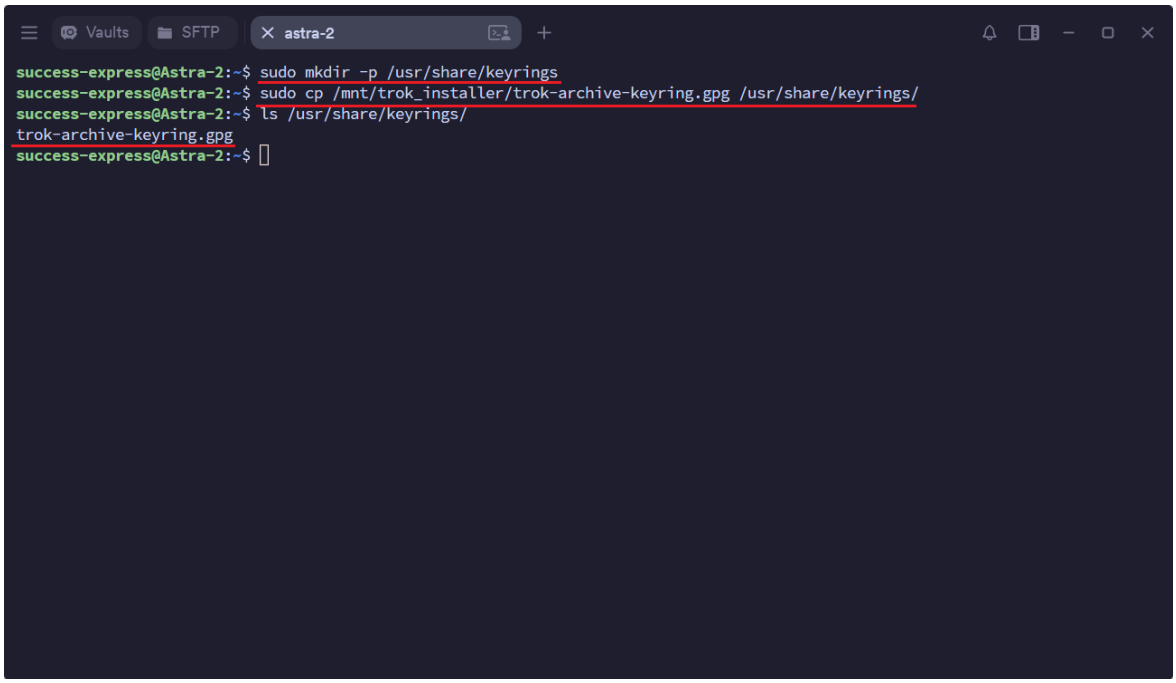


Рисунок 21 – Процесс копирования ключа

Установите корректные права для ключа. Это обеспечит системам безопасности надлежащий доступ к ключу.

```
sudo chmod 644 /usr/share/keyrings/trok-archive-keyring.gpg
```

Создайте файл приоритета:

```
sudo vim /etc/apt/preferences.d/99-trok.pref
```

(Обратите внимание: между 99- и trok.pref не должно быть пробелов).

Добавьте в файл следующее содержимое:

```
Package: *
Pin: release o=TROK
Pin-Priority: 1001
```

Такая запись гарантирует приоритет использования пакетов из вашего локального репозитория при установке.

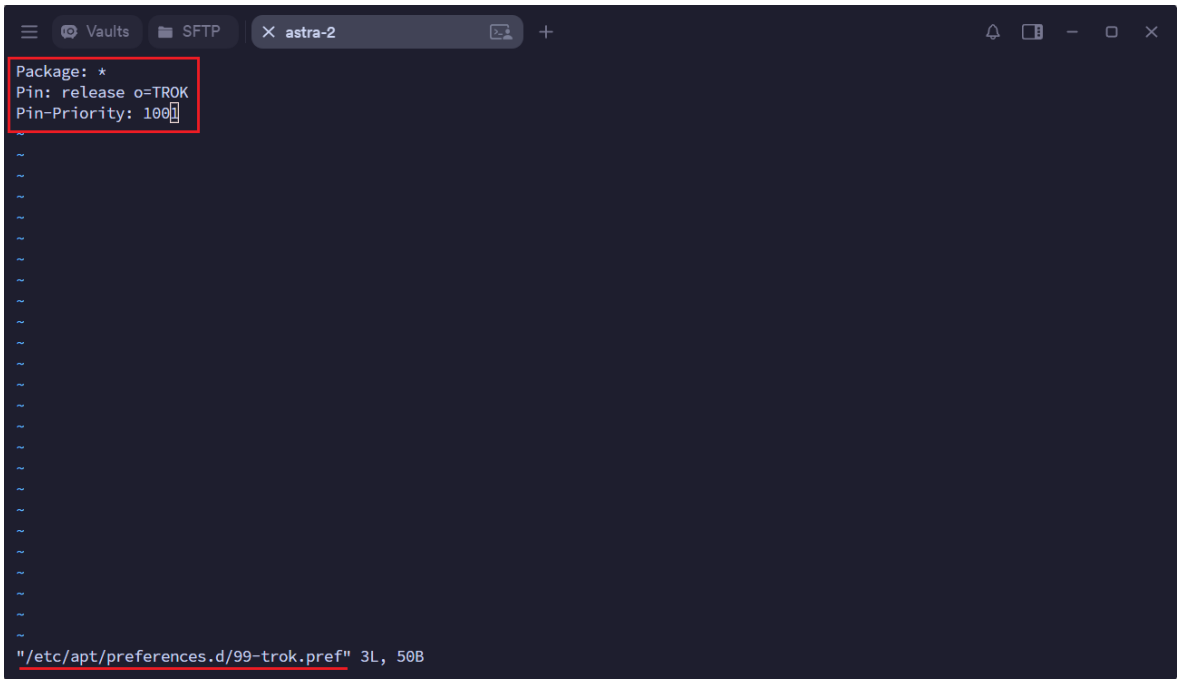


Рисунок 22 – Файл 99-trok.pref после внесения изменений

После выполнения всех подготовительных шагов обновите индекс пакетов. Для этого выполните:

```
sudo apt update
```

Теперь возможно устанавливать программное обеспечение с использованием пакетов.

Если предопределенная роль – выделенный контроллер (controller), начните с установки пакета trok-webui, поскольку управление контроллером осуществляется с его помощью. Обратите внимание, что установка пакетов осуществляется в строгой последовательности. Используйте команду:

```
sudo apt install trok-webui
```

Веб-сервер nginx устанавливается автоматически во время установки пакета trok-webui.

Рисунок 23 – Уведомление о настройке nginx

Если этого не произошло, воспользуйтесь дополнительной командой:

```
sudo apt install nginx
```

Установите компонент контроллер:

```
sudo apt install trok-controller-meta
```

В качестве завершающего этапа выполните установку пакета trok-harbor-meta:

```
sudo apt install trok-harbor-meta
```

Если предопределенная роль – узел хранения (worker), используйте команду:

```
sudo apt install trok-worker-meta
```

В качестве завершающего этапа выполните установку пакета trok-harbor-meta:

```
sudo apt install trok-harbor-meta
```

После установки обязательно выполните команду:

```
sudo trok-drbd-autoinstall
```

В качестве последнего альтернативного варианта установки выступает комбинированный узел. Для того, чтобы не вводить много однотипных команд, установку пакетов и зависимостей можно произвести одной командой:

```
sudo apt install trok-webui nginx trok-combined-meta trok-harbor-meta
```

После установки обязательно выполните команду:

```
sudo trok-drbd-autoinstall
```

Рекомендуется перезагрузить систему, когда все требуемые пакеты будут установлены на узле.

Важно. После завершения установки компонента worker, всегда необходимо выполнить команду для запуска программы инсталляции DRBD от имени суперпользователя (root): sudo trok-drbd-autoinstall.

Важно. В случае, если в вашей операционной системе возникают проблемы с установкой DRBD, например, как в случае на изображении ниже, установите пакет drbd-dkms.

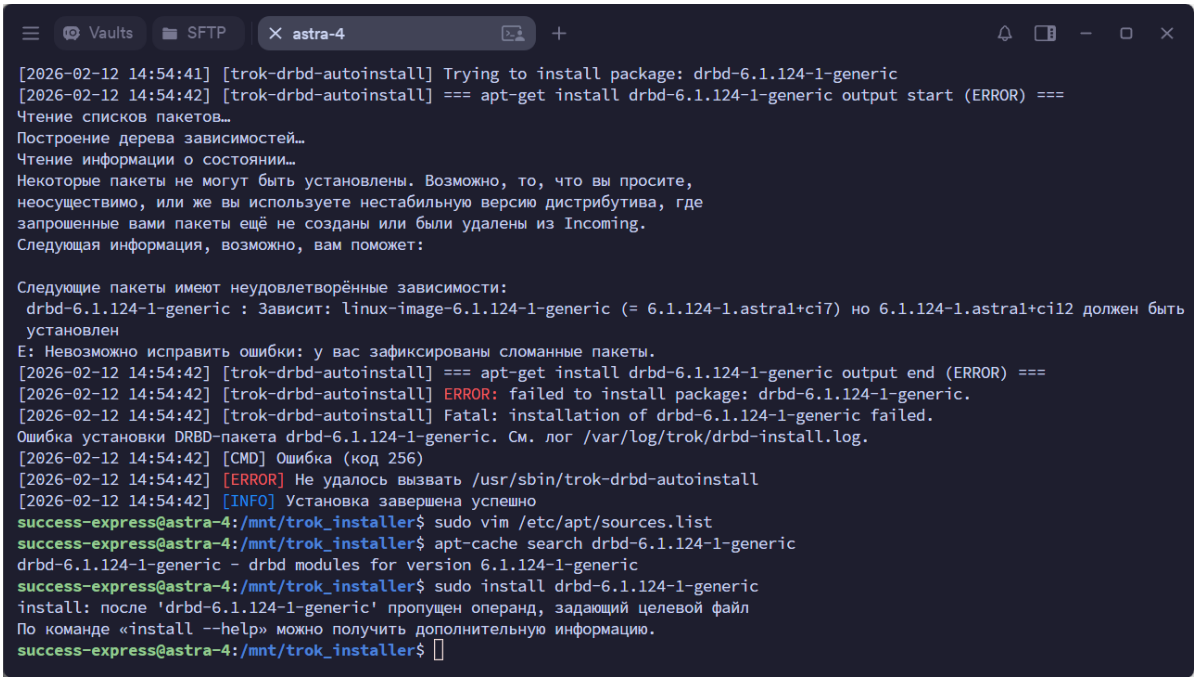


Рисунок 24 – Ошибка совместимости DRBD-пакета с ядром Linux

Пакет drbd-dkms также устанавливается из архива с пакетами TROK после выполнения всех необходимых действий по добавлению локальных пакетов в список репозиториев.

Команда для установки:

```
sudo apt install drbd-dkms
```

Рисунок 25 – Процесс установки пакета drbd-dkms

9. ДЕПЛОЙ КЛАСТЕРА

9.1. Требования к управляющему хосту (узел управления Ansible)

Минимальные требования к программному обеспечению:

– Ansible версии 2.15.0 или выше – инструмент для автоматизации конфигурации и управления инфраструктурой, обеспечивающий выполнение задач на удаленных узлах.

– Python 3 и пакет python3-netaddr – python 3 требуется как базовая среда выполнения для Ansible; пакет python3-netaddr предоставляет дополнительные функции для работы с IP-адресами и сетевыми диапазонами, необходимые для сценариев автоматизации.

– Коллекция community.general – набор дополнительных модулей Ansible из официального репозитория Ansible Galaxy, расширяющий функциональность для выполнения общих задач, таких как управление сетевыми интерфейсами или файловыми системами.

– SSH-доступ к узлам кластера – обеспечивает безопасное соединение с целевыми узлами (controller и worker), позволяя Ansible выполнять команды удаленно без необходимости физического доступа.

Для подготовки узла управления к работе с Ansible выполните следующие команды в терминале (команды выполняются с правами суперпользователя (sudo)):

Обновите локальный индекс пакетов системы управления пакетами APT (Advanced Package Tool):

```
sudo apt update
```

Установите пакеты Ansible и python3-netaddr с помощью APT:

```
sudo apt install -y ansible python3-netaddr
```

Введите команду, которая использует инструмент Ansible Galaxy (часть Ansible) для установки коллекции модулей community.general из официального репозитория:

```
ansible-galaxy collection install community.general
```

9.2. Установка коллекции TROK

Если коллекция astra.trok подготовлена в виде архивного файла (например, astra-trok-1.0.0.tar.gz), произведите ее установку перед деплоем. Коллекция представляет собой набор модулей и ролей Ansible, предназначенный для автоматизации задач развертывания (деплоя) в инфраструктуре.

Рисунок 26 – Коллекция astra-trok (размещается в каталоге ansible-collection, который находится внутри смонтированного ISO-образа TROK)

Выполните команду, которая устанавливает коллекцию astra-trok из локального файла архива astra-trok-<версия>.tar.gz (относительный путь ./ указывает на текущую директорию):

```
ansible-galaxy collection install ./astra-trok-<версия>.tar.gz
```

В случае повторной установки (например, если коллекция была обновлена или требуется перезапись существующей версии) используйте флаг --force для принудительного выполнения операции, игнорируя возможные конфликты:

```
ansible-galaxy collection install ./astra-trok-<версия>.tar.gz --force
```

После установки рекомендуется выполнить проверку, чтобы подтвердить, что коллекция успешно интегрирована в систему и готова к использованию:

```
ansible-galaxy collection install ./astra-trok-<версия>.tar.gz --force
```

В случае проблем проверьте логи Ansible или запустите команды с повышенными правами (sudo), если требуется.

9.3. Подготовка структуры проекта

Для подготовки среды развертывания (деплоя) системы с использованием Ansible рекомендуется создать организованную структуру каталогов и файлов. Структура рабочего каталога trok-deploy/ должна выглядеть следующим образом:

```
trok-deploy/
├─ inventory.ini
├─ group_vars/
│  └─ all.yaml
└─ site.yml
```

inventory.ini – файл инвентаря, который определяет список целевых хостов (серверов), их группы и параметры подключения (например, IP-адреса, SSH-ключи).

group_vars/all.yaml – файл переменных в формате YAML, расположенный в подкаталоге group_vars/. Он содержит глобальные переменные, доступные всем хостам, такие как пути к репозиториям, версии компонентов или ключи шифрования.

site.yml – основной плейбук Ansible в формате YAML, расположенный в корневом каталоге. Файл описывает последовательность задач развертывания, включая установку ПО, конфигурацию служб и проверку состояния. Он служит точкой входа для запуска всего процесса автоматизации.

9.4. Настройка Inventory

Файл inventory.ini служит статическим инвентарем в формате INI, где перечислены серверы, разделенные на группы, с указанием необходимых атрибутов для успешного выполнения плейбуков. Этот файл обеспечивает

Ansible информацией о том, как и где выполнять задачи, минимизируя риски ошибок в сетевом взаимодействии и аутентификации.

Важные рекомендации по конфигурации:

Для корректной работы плейбуков коллекции строго рекомендуется указывать IP-адрес каждого узла в параметре ansible_host, даже если на целевых хостах настроено разрешение DNS (Domain Name System) или FQDN (Fully Qualified Domain Name). Логика плейбуков напрямую зависит от этого параметра, поскольку он гарантирует точное определение сетевых адресов без reliance на внешние службы разрешения имен.

Указание только имени хоста без ansible_host может привести к сбоям в обработке инвентаря, подключении к узлам и выполнении задач, что нарушит процесс развертывания кластера.

Файл inventory.ini состоит из секций (групп хостов), где каждая группа объединяет узлы с общими характеристиками. Для развертывания TROK типичными группами являются [controllers] (контроллеры, управляющие компоненты кластера) и [satellites] (воркеры, дополнительные узлы). Каждый хост в группе описывается строкой с именем (alias), за которым следуют параметры подключения. Параметры включают:

– ansible_host: IP-адрес или FQDN узла для прямого доступа;

– ansible_user: Имя пользователя для SSH-подключения (например, ansible);

– ansible_port: Порт SSH (стандартно 22);

– ansible_connection: Тип соединения (обычно ssh для безопасного протокола);

– ansible_ssh_private_key_file: Путь к приватному SSH-ключу на узле управления (например, ~/.ssh/testinfra_key);

– ansible_ssh_common_args: Дополнительные аргументы SSH, такие как -o StrictHostKeyChecking=no (отключение проверки хоста) и -o UserKnownHostsFile=/dev/null (игнорирование файла известных хостов), для упрощения автоматизации в тестовых или изолированных средах.

```
[controllers]
cmbnd ansible_host=10.177.161.211 ansible_user=ansible ansible_port=22
ansible_connection=ssh \
ansible_ssh_private_key_file=~/.ssh/testinfra_key \
ansible_ssh_common_args="-o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null"

[satellites]
sat1 ansible_host=10.177.161.212 ansible_user=ansible ansible_port=22
ansible_connection=ssh \
ansible_ssh_private_key_file=~/.ssh/testinfra_key \
ansible_ssh_common_args="-o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null"

sat2 ansible_host=10.177.161.213 ansible_user=ansible ansible_port=22
ansible_connection=ssh \
ansible_ssh_private_key_file=~/.ssh/testinfra_key \
ansible_ssh_common_args="-o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null"

[all:vars]
ansible_become=true
```

9.5. Настройка переменных (group_vars/all.yaml)

Файл group_vars/all.yaml представляет собой критический элемент автоматизированной настройки инфраструктуры на основе Ansible, предназначенный для определения глобальных параметров, применяемых ко всем узлам кластера TROK. YAML-файл (Yet Another Markup Language) обеспечивает централизованное управление конфигурациями.

Файл разделен на логические блоки, каждый из которых фокусируется на определенном аспекте конфигурации. Ниже приведен пример с объяснениями каждой переменной:

```
########################################################################
# Источник пакетов TROK: локальный ISO-образ на управляющем хосте
# URL-адрес не используется (оставляем пустым для предотвращения сетевых
зависимостей)
trok_iso_url: ""
# Локальный путь к ISO-файлу на управляющем хосте (адаптируйте под вашу файловую
систему)
trok_iso_local_src: "/home/ansible/iso/ trok-0.1.1-Astra1.7_amd64.iso"
########################################################################

########################################################################
# Сетевая конфигурация кластера TROK
# Список IP-адресов контроллерных узлов (управляющих компонентов кластера)
controllers:
```

```
    - 10.177.161.211
  # Список IP-адресов узлов типа worker (дополнительных компонентов для расширения)
  satellites:
    - 10.177.161.212
    - 10.177.161.213
  # Подсеть для репликации данных через DRBD (Distributed Replicated Block Device)
  drbd_replication_network: 10.177.161.0/24
  # Активация развертывания TROK Gateway для протоколов iSCSI, NFS и NVMe (обеспечивает доступ к хранилищам)
  enable_gateway: true
  # Конфигурация брандмауэра UFW (Uncomplicated Firewall): отключение для упрощения в тестовых средах
  ufw_enable: false
  ######################################################################
```

Описание переменных

– trok_iso_url: Пустая строка, указывающая на отсутствие удаленного источника пакетов. Это предотвращает зависимость от внешних сетевых ресурсов, обеспечивая автономность развертывания.

– trok_iso_local_src: Абсолютный путь к локальному ISO-образу на управляющем хосте. Этот файл содержит все необходимые пакеты для установки TROK, и его корректный путь критичен для успешного выполнения плейбуков.

– controllers: Массив IP-адресов узлов-контроллеров, отвечающих за координацию и управление кластером. Эти узлы формируют ядро системы.

– satellites: Массив IP-адресов узлов, предназначенных для хранения, расширяющих возможности кластера путем распределения нагрузки и хранения данных.

– drbd_replication_network: CIDR-нотация подсети (Classless Inter-Domain Routing), выделенной для репликации блоков данных между узлами с использованием DRBD.

– enable_gateway: Булево значение (true/false), активирующее развертывание шлюза TROK для поддержки протоколов хранения (iSCSI для блочных устройств, NFS для файловых систем, NVMe для высокопроизводительного доступа).

– ufw_enable: Булево значение для управления брандмауэром UFW. Отключение (false) рекомендуется в контролируемых средах для избежания

блокировки необходимых портов, но в продакшене следует активировать для безопасности.

9.6. Что делает плейбук

Плейбук Ansible представляет собой автоматизированный сценарий, предназначенный для последовательного выполнения задач по установке и настройке компонентов кластера TROK на целевых узлах. Ниже приведено описание ключевых операций, выполняемых плейбуком, с пояснениями.

– Монтирование образа ISO на целевых узлах:

Плейбук монтирует образ ISO на файловую систему целевых узлов. ISO-файл содержит все необходимые установочные файлы, драйверы и репозитории. Монтирование делает их доступными для чтения, обеспечивая локальный доступ к ресурсам без необходимости сетевых загрузок.

– Извлечение локального APT-репозитория из ISO в указанный путь:

Из образа ISO извлекается локальный репозиторий пакетов APT и размещается в директории, заданной переменной trok_repo_path.

– Подключение репозитория к системе управления пакетами:

Плейбук интегрирует извлеченный репозиторий в конфигурацию APT системы.

– Установка необходимых пакетов TROK:

Выполняется установка пакетов, требуемых для функционирования кластера TROK. Пакеты включают ядро системы хранения данных (SDS), инструменты для репликации и управления.

– Настройка сетевых параметров и системных служб:

Плейбук конфигурирует сетевые интерфейсы, маршрутизацию и связанные службы, что включает в себя настройку IP-адресов, firewall-правил и параметров сетевой безопасности.

– Условное развертывание TROK Gateway при включенной опции:

Если переменная enable_gateway установлена в true, плейбук разворачивает компонент TROK Gateway. Gateway выступает в роли шлюза для внешнего доступа к кластеру, обеспечивая интеграцию с протоколами вроде HTTP/HTTPS.

– Активация служб и их включение в автозагрузку:

Плейбук запускает системные службы TROK и настраивает их автоматический запуск при перезагрузке системы.

9.7. Плейбук для деплоя (site.yml)

Плейбук site.yml представляет собой автоматизированный сценарий на основе Ansible, предназначенный для последовательного развертывания компонентов кластера TROK. Этот файл YAML (Yet Another Markup Language) структурирован как набор задач (plays), каждая из которых определяет целевые узлы, параметры выполнения и роли для применения.

```
---
# Плейбук для развертывания кластера TROK

- name: Controller deploy
  hosts: controllers
  gather_facts: true
  become: true
  become_user: root
  roles:
    - { role: astra.trok.controller }

- name: Satellite deploy
  hosts: satellites
  gather_facts: true
  become: true
  become_user: root
  roles:
    - { role: astra.trok.satellite }
```

Описание структуры плейбука:

– Задачи (Plays): Каждая задача (play) – это независимый блок, выполняемый последовательно. Первая задача фокусируется на контроллерах (ядро кластера), вторая – на узлах хранения (worker, расширяющие узлы для распределения нагрузки).

– hosts: Указывает группу узлов из инвентаря Ansible (inventory), определенную в group_vars/all.yaml.

– gather_facts: Параметр, активирующий сбор системных данных (например, ОС, IP-адреса, аппаратные ресурсы). Это критично для условного выполнения задач, основанного на характеристиках узла.

– become и become_user: Директивы обеспечивают эскалацию привилегий до root, необходимую для установки пакетов, изменения конфигураций и управления службами.

roles: Ссылка на роли Ansible – модульные компоненты, содержащие задачи, шаблоны и переменные. Роль astra.trok.controller управляет установкой контроллера (центрального узла для координации), а astra.trok.satellite – воркера (для хранения и репликации данных). Роли должны быть предварительно установлены или загружены.

9.8. Запуск деплоя

Для инициации процесса развертывания (деплоя) кластера TROK выполните следующие шаги на управляющем хосте.

Перейдите в директорию проекта, содержащую файлы развертывания:

```
cd trok-deploy
```

Запустите полный плейбук для развертывания всех компонентов кластера (контроллеров и воркеров):

```
ansible-playbook -i inventory.ini site.yml
```

Команда ansible-playbook исполняет плейбук site.yml, используя инвентарь inventory.ini для определения целевых узлов. Параметр -i указывает путь к файлу инвентаря, который содержит группы узлов (например, controllers и workers).

Опции для частичного развертывания:

Для оптимизации процесса и тестирования можно ограничить выполнение плейбука конкретными группами узлов, используя флаг --limit. Это позволяет развернуть только контроллеры или воркеры.

Развертывание только контроллерных узлов:

```
ansible-playbook -i inventory.ini site.yml --limit controllers
```

Развертывание только узлов хранения (типа worker):

```
ansible-playbook -i inventory.ini site.yml --limit satellites
```

Аналогично, --limit satellites фокусирует процесс на группе satellites, обеспечивая развертывание расширяющих узлов для распределения нагрузки и хранения данных.

9.9. Проверка после деплоя

9.9.1. Контроллер

Выполните команду для проверки статуса службы контроллера:

```
sudo systemctl status trok-controller
```

Команда systemctl status (из systemd) отображает текущее состояние службы (активна/неактивна), журнал событий и возможные ошибки, позволяя оперативно выявить проблемы с запуском или конфигурацией.

9.9.2. Воркеры

Выполните команду для проверки статуса служб типа worker:

```
sudo systemctl status trok-worker || sudo systemctl status trok-controller
```

Оператор || обеспечивает альтернативную проверку (fallback) на случай, если основная служба недоступна.

9.9.3. Проверка взаимодействия

После успешного развертывания необходимо подтвердить корректную связь между контроллером и воркерами.

– Для проверки доступа к веб-интерфейсу TROK в веб-браузере перейдите по адресу контроллера:

http://<IP_контроллера>

Например: http://10.177.161.211

– Для аутентификации в системе войдите с использованием учетных данных по умолчанию:

Логин: admin

Пароль: admin

– Для навигации к разделу узлов перейдите во вкладку «Узлы».

– Для проверки статуса подключения убедитесь, что все узлы отображаются в списке со статусом «Online». Если какой-либо узел находится в состоянии «Offline», проверьте сетевое подключение и настройки репликационной сети (параметр drbd_replication_network). В случае проблем рекомендуется использовать инструменты вроде ping или ip route для диагностики.

10. ЗАВЕРШЕНИЕ УСТАНОВКИ

После завершения установки необходимо проверить, что компонент DRBD корректно интегрирован в систему. Ниже приведены ключевые команды проверки с краткими пояснениями для каждой строки или блока. Успешное выполнение команд указывает на готовность модуля DRBD к использованию.

Процедура верификации модуля DRBD применяется исключительно на узлах хранения данных (trok-worker, trok-combined). Если на узле размещается только trok-controller, наличие модуля DRBD не обязательно.

Выполните проверку версии ядра ОС (kernel release):

```
sudo uname -r
```

Пример вывода подтверждает загруженную версию ядра (здесь: 6.1.152-1-generic):

```
6.1.152-1-generic
```

DRBD – это модуль ядра Linux. Он должен быть скомпилирован и загружен именно для той версии ядра, которая сейчас активна. Несовпадение версии ядра и версии DRBD может привести к неработоспособности хранения, поэтому следующим шагом проверьте версию DRBD, чтобы сопоставить данные.

Для этого выполните команду, которая отобразит перечень установленных пакетов, применяя фильтр для вывода только тех, в наименовании которых присутствует «drbd»:

```
sudo apt list --installed | grep drbd
```

Пример вывода:

```
drbd-6.1.152-1-generic/stable,stable,now 9.2.12-4 amd64 [установлен] # Установлен связанный с ядром модуль DRBD версии 9.2.12.

drbd-reactor/stable,now 1.4.1-1 amd64 [установлен] # Установлен компонент реактора для управления ресурсами.

drbd-ueficert/now 9.2.11-1 all [установлен, локальный] # Установлен сертификат UEFI для DRBD.
```

```
   drbd-utils/stable,now 9.28.0-1 amd64 [установлен] # Установлены утилиты DRBD версии 9.28.0.
```

Анализ представленного вывода подтверждает, что все необходимые компоненты (ядро DRBD, утилиты, интеграционные модули, сертификаты) установлены корректно. В случае отсутствия какого-либо из указанных элементов в списке результатов, требуется их установка. Судя по префиксу «drbd-6.1.152-1-generic…», используется модуль DRBD версии 9.2.12. Он подходит к текущему ядру, потому что его версия совпадает с тем, что показывает команда sudo uname -r.

Выполните команду для отображения информации о версии и сборке DRBD Admin utility:

```
   sudo drbdadm --version
```

Пример вывода:

```
   DRBDADM_BUILDTAG=GIT-hash:\ ba2ce9037989b6141222c7901d1219cf852949f1configure.ac\ build\ by\ root@linux\,\ 2025-01-17\ 18:32:01 # Метка сборки с хэшем Git и датой компиляции.

   DRBDADM_API_VERSION=2 # Версия API drbdadm (2).

   DRBD_KERNEL_VERSION_CODE=0x09020c # Код версии ядра DRBD (9.2.12).

   DRBD_KERNEL_VERSION=9.2.12 # Человеко-читаемая версия ядра DRBD.

   DRBDADM_VERSION_CODE=0x091c00 # Код версии drbdadm (9.28.0).

   DRBDADM_VERSION=9.28.0 # Версия утилиты drbdadm.
```

Вывод показывает, что утилита для управления DRBD (drbdadm) установлена и работает. Любая рассинхронизация или отсутствие – свидетельство неправильной установки и источник будущих проблем кластера.

Если версии не совпадают или утилита не работает – требуется устранить рассогласования в установке.

Проверьте загруженные модули ядра, фильтруя по «drbd»:

```
   sudo lsmod | grep -i drbd
```

Пример вывода:

```
   drbd      819200   0   # Основной модуль DRBD загружен (размер 819200 байт, используется 0 процессами).

   lru_cache  16384  1 drbd # Зависимый модуль LRU-кэша, используемый DRBD.

   libcrc32c   16384  6 nf_conntrack,nf_nat,dm_persistent_data,nf_tables,drbd,sctp # Модуль CRC32C, включая DRBD как одного из клиентов.
```

DRBD должен быть не просто установлен, а также загружен в ядро ОС как модуль и готов к работе. Без этого система хранения работать не сможет – даже если все пакеты формально установлены.

Если строки drbd нет: DRBD не загружен – либо из-за ошибки, либо несовместимости. Нужно выполнить установочную команду trok-drbd-autoinstall от имени суперпользователя или разбираться с ошибкой ядра.

Для подтверждения успешной установки компонентов кластера TROK рекомендуется выполнить команду на узлах контроллера:

```
   sudo systemctl status trok-auth.service trok-controller.service trok-cp-endpoint.service trok-worker.service
```

Если на узле контроллера вы не устанавливали компонент trok-worker, удалите этот аргумент из запроса:

```
   sudo systemctl status trok-auth.service trok-controller.service trok-cp-endpoint.service
```

Для узлов хранения (типа worker) команда будет иметь меньшее количество аргументов:

```
   sudo systemctl status trok-worker.service
```

Все основные компоненты (аутентификация, контроллер, endpoint, worker) должны запускаться как сервисы. Если сервис не стартовал/имел ошибку на запуске – система хранения не будет работать, как ожидается.

Если сервисы имеют статус active (running), нет ошибок типа failed, inactive и т.п., результат удачный.

Отрицательный сценарий: ошибки загрузки, сервисы в состоянии failed. Требуется диагностика: изучите логи, проверьте шаги установки/настройки.

Рисунок 27 – Результат вывода команды проверки статуса установленных служб (комбинированный тип узла)

11. ОСОБЕННОСТИ ПЕРЕУСТАНОВКИ КОМПОНЕНТОВ

Если при переустановке пакета trok-webui возникают ошибки (например, конфликты конфигураций или зависимостей), следуйте этой последовательности команд для полной очистки и повторной установки:

Удалите пакет с полным очищением:

```
sudo apt remove --purge trok-webui
```

Команда remove --purge удаляет пакет и все его конфигурационные файлы, а не только исполняемые файлы. Такой способ удаления предотвращает конфликты при повторной установке.

Удалите ненужные зависимости:

```
sudo apt autoremove
```

Это полезная команда для регулярного обслуживания системы, так как она эффективно очищает неиспользуемые файлы любого программного обеспечения и поддерживает оптимальную производительность.

Переустановите пакет:

```
sudo apt install trok-webui
```

Команды следует выполнять с правами администратора (sudo).

12. ЛОГИРОВАНИЕ

Логи сохраняются в директорию /var/log/trok-installer (или в указанную через --log). Имя файла лога генерируется автоматически на основе текущей даты и времени.

Пример:

/var/log/trok-installer/install_1750759134.log

Система осуществляет автоматическую регистрацию событий в соответствии со следующей схемой:

- Хранилище журналов
  - Основной каталог: /var/log/trok-installer/
  - Альтернативный путь: определяется параметром --log при запуске (подробности в пункте 5)
- Генерация имен файлов

  Формируется по шаблону: install_<timestamp>.log

  где: <timestamp> – это количество секунд, прошедших с начала эпохи Unix (01.01.1970). Например, install_1750759134.log

13. КРИТИЧЕСКИЕ ПРЕДУПРЕЖДЕНИЯ

Перед началом удаления обратите внимание на следующие особенности:

– Порядок удаления: Нарушение последовательности (например, удаление пулов до ресурсов) приведет к ошибкам InUse.

– Тестирование: Перед выполнением в продакшене протестируйте процедуру на стенде.

– Служебные данные: Не удаляйте директории /srv/ha/internal/ до полной остановки сервисов — это может вызвать ошибки NFS 1.

14. УДАЛЕНИЕ СТРУКТУРНЫХ КОМПОНЕНТОВ SDS

Удаление структурных компонентов SDS производится с использованием web-интерфейса TROK.

Очередность удаления компонентов SDS:

– 1 шаг – ресурсы;

– 2 шаг – шаблоны томов;

– 3 шаг – шаблоны ресурсов;

– 4 шаг – группы ресурсов;

– 5 шаг – пулы хранения;

– 6 шаг – узлы;

Более подробная информация указана в документе «TROK Руководство по работе с графическим интерфейсом».

15. ПОЛНОЕ УДАЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

15.1. Удаление пакетов формата deb

Для удаления установленных пакетов выполните команду:

Внимание! Следующая команда полностью удалит компоненты TROK и веб-сервер **nginx**. Если **nginx** использовался для других целей, не включайте его в команду удаления.

```
sudo apt purge trok-controller-meta trok-worker-meta trok-harbor-meta trok-cp-endpoint trok-auth trok-webui nginx
```

Затем удалите неиспользуемые зависимости:

```
sudo apt autoremove
```

Если необходимо удалить пакеты по отдельности, используйте команду:

```
sudo apt purge <имя_пакета>
```

где <имя_пакета> — название конкретного пакета для удаления.

Если необходимо удалить deb-пакеты по отдельности, следует учитывать, какие пакеты устанавливаются вместе с каждым компонентом системы. Полный перечень устанавливаемых пакетов представлен в примечаниях к выпуску (Release Notes) для каждой конкретной версии программного обеспечения.

15.2. Очистка LVM-томов и блочных устройств

Перед удалением LVM-томов и блочных устройств необходимо остановить службу trok-controller командой:

```
sudo systemctl stop trok-controller
```

На узле типа worker удалите описания ресурсов:

```
sudo rm -r /etc/drbd.d/*.res
```

Перезагрузите сервер:

```
sudo reboot
```

Проверьте имеющиеся группы томов:

```
sudo lvs
```

Команда выводит текущее состояние логических томов (LV), их VG, размер и состояние.

Удалите все логические тома в вашей группе томов:

```
sudo lvremove -y /dev/<имя_группы_томов>/*
```

Команда без запроса подтверждения удаляет все логические тома внутри группы томов. Процесс полностью и безвозвратно стирает тома и связанные данные. Может потребоваться предварительно отмонтировать тома и убедиться, что данные не нужны.

Для полной очистки содержимого блочного устройства (например, диска) применяется команда:

```
sudo dd if=/dev/zero of=/dev/sd<буква_тома> bs=1G status=progress
```

где /dev/sd<буква_тома> – имя целевого устройства. Эта операция перезаписывает устройство нулями, что гарантирует удаление всех данных. Используйте с осторожностью.