



**ПРОГРАММНЫЙ КОМПЛЕКС
«ПРОСТОР»**

**Инструкция по установке экземпляра
программного обеспечения**



ВВЕДЕНИЕ

Настоящий документ описывает порядок установки экземпляра Программного комплекса «Простор» на целевом сервере под управлением операционной системы Astra Linux.



СОДЕРЖАНИЕ

1. Описание действий, обеспечивающих запуск ПК «Простор»	4
1.1. Общая схема развертывания ПК «Простор»	4
1.2. Требования к целевому серверу.....	4
1.3. Состав дистрибутива.....	5
1.4. Порядок действий при установке ПК «Простор»	6
1.5. Описание шагов развертывания	11

1. ОПИСАНИЕ ДЕЙСТВИЙ, ОБЕСПЕЧИВАЮЩИХ ЗАПУСК ПК «ПРОСТОР»

1.1. Общая схема развертывания ПК «Простор»

- Развертывание ПК «Простор» выполняется одним оркестратором `run_prostor.sh`, который последовательно запускает пять фаз:
- `destroy_services` — остановка и удаление ранее установленных прикладных сервисов, и создание бэкапа;
- `deploy_prostor -d` — удаление ядра (NATS, PostgreSQL, БД `prostor`);
- `deploy_prostor` — установка ядра (NATS, PostgreSQL, восстановление БД `prostor`);
- `deploy_services` — раскладка файлов прикладных сервисов по целевым путям;
- `install_services` — запуск прикладных сервисов через `systemd`.

Перед запуском рекомендуется выполнить три подготовительных скрипта:

- `check_server.sh` → собрать «паспорт» целевого сервера (железо, диски, каталоги, сервисы, порты, репозитории);
- `checkrisk_prostor.sh` → выявить риски неудачного развёртывания и проверить целостность комплекта развёртывания;
- `check_archives.sh` → проверить дистрибутивы прикладных сервисов на безопасность и полноту.

Все скрипты находятся в одном каталоге с `run_prostor.sh` (пример: `/tmp/run_prostor`).

1.2. Требования к целевому серверу

- ОС: Astra Linux SE 1.8.1.* (поддерживаются все версии ветки 1.8.1);
- права: `root` или `sudo`;
- CPU: минимум 8 ядер;
- RAM: минимум 32 GB;
- диск: минимум 1 TB (с учётом видеоархива и `tiles`);
- разделы: под `/var/Archives` (видеоархив) и `/opt/prostor_pp/Tiles` (`tiles`)

рекомендуется выделять отдельные ФС;

- порты: 4222 (NATS client), 8222 (NATS monitor), 5432 (PostgreSQL) — должны быть свободны;
- репозитории: доступные deb-источники (локальные или сетевые);
- синхронизация времени: обязательна. Сервис должен быть синхронизирован по времени любой доступной утилитой (например, NTP);
- кодировка для БД: ru_RU.UTF-8;
- часовой пояс: Europe/Moscow.

1.3. Состав дистрибутива

Скрипт развёртывания в качестве единственного параметра принимает на вход путь до каталога `hostinfo<C>-<D>` — версию данных для развёртывания (пример имени: `hostinfo140-220`).

Внутри каталога размещены:

- `extract/` — папка с дистрибутивами прикладных сервисов (для каждого сервиса — бинарник, `.cfg` и `.service`);
- `nats-amd64.tar.gz` — дистрибутив NATS;
- `prostor_backup_*.tar.gz` — архив структуры данных БД;
- `services.tsv` — список устанавливаемых прикладных сервисов.

В том же каталоге, что и `run_prostor.sh` должны находиться скрипты:

- `commfunc.sh`
- `check_server.sh`
- `checkrisk_prostor.sh`
- `check_archives.sh`
- `deploy_services.sh`
- `destroy_services.sh`
- `install_services.sh`
- `run_prostor.sh`



1.4. Порядок действий при установке ПК «Простор»

1.4.1. Шаг 1. Проверка целевого сервера

Скрипт: `check_server.sh`

Команда: `sudo ./check_server.sh`

Назначение - Автоматизирует проверку целевого сервера, которую инженер и так должен был бы провести вручную: собирает «паспорт» хоста (общая информация, диски, каталоги `/opt/prostor_pp`, состояние PostgreSQL и Tiles, статусы сервисов, занятость портов 4222/8222/5432, доступность репозитория). Никакие проверки не блокирует — только выводит данные для анализа инженером.

Что выводится:

1. General info — hostname, IP, дата, пользователь, ядро, сборка Astra;
2. Disks — `lsblk + df -h` по ключевым точкам;
3. HomeDir = `/opt/prostor_pp` — перечень подкаталогов сервисов с владельцем и размером;
4. Databases info — таблица по PostgreSQL main, tablespace video, каталогу Tiles (владелец, права, размер);
5. State: services — таблица по каждому сервису: наличие unit, is-active, is-enabled, `/etc/<service>`, версия (для PG и NATS);
6. порты — 4222, 8222, 5432 (LISTEN / нет).
7. sources.list — содержимое и построчная проверка доступности репозитория.

Пример корректного вывода (фрагмент):

Hostname: <hostname>

IP: <IP-адрес>

Kernel: <версия ядра>

Astra build: <сборка Astra>

sdb <объём> → `/var/Archives, /opt/prostor_pp/Tiles`

postgres `/var/lib/postgresql/15/main` postgres:postgres 700 <объём>

tablespace `/mnt/data/video` postgres:postgres 700 <объём>

tiles `/opt/prostor_pp/Tiles` root:root 777 skipped



4222 LISTEN nats

8222 LISTEN nats-monitor

5432 LISTEN postgresql

Как трактовать:

- все ключевые разделы, сервисы и порты на месте — переходим к шагу 2;
- любые inactive/FAIL/- в разделах 3–6 — повод разобраться до запуска развёртывания;
- сервисы, отсутствующие в поставке (например, ARM, NetPing, DrPelcoDe), могут не иметь unit — это не ошибка.

1.4.2. Шаг 2. Оценка рисков неудачного развёртывания

Скрипт: checkrisk_prostor.sh

Команда: sudo ./checkrisk_prostor.sh

Назначение - Выявляет обстоятельства на целевом сервере, которые могут привести к сбою установки PostgreSQL или базы данных «Простор» либо к неудачному развёртыванию в целом: наличие и права каталога tablespace video, отсутствие остатков PG_, наличие записи в /etc/fstab, свободное место, доступность репозитория, отсутствие ранее установленных PostgreSQL и NATS, а также целостность комплекта run_prostor (наличие и +x для скриптов, наличие deploy_prostor, nats-amd64.tar.gz, prostor_backup_.tar.gz). Скрипт только предупреждает о рисках неудачного развёртывания (WARN / FAIL); решение данных проблем должно производиться до запуска установки.

Производимые проверки:

1. Хранилище видео: Tablespace video — существование /mnt/data и /mnt/data/video, тип (symlink / mountpoint / каталог), права (postgres:postgres 700), остатки PG_*, свободное место (порог 10 GB), запись /mnt/data в /etc/fstab.
2. Репозиторий дистрибутивов Astra: Repository availability — все deb-источники и их доступность.
3. Предыдущие установки ПО — установлены ли postgresql-15 / nats-server.



4. Целостность пакета дистрибутива — наличие и +x для 7 скриптов, синтаксис каждого, наличие `deploy_prostor`, `nats-amd64.tar.gz`, `prostor_backup_*.tar.gz`.

Пример вывода:

1. Tablespace video

/mnt/data OK exists

tablespace mount WARN

RISK: Это директория, а не точка монтирования. Данные лежат в корневой ФС, при переустановке системы потеряются.

SOLUTION: Смонтировать отдельный раздел в /mnt/data/video

tablespace perms OK postgres:postgres 700

PG_* leftovers OK clean

free space OK <объём>

fstab /mnt/data WARN

RISK: Нет записи в /etc/fstab. После перезагрузки раздел не смонтируется, PostgreSQL упадёт при старте.

SOLUTION: Добавить строку в /etc/fstab с UUID раздела

2. Repository availability — все репозитории OK

3. Existing installations — clean

4. Package integrity — 7/7, `deploy_prostor`, `nats`, `backup` OK

Найдено рисков: N

Действия при возникновении извещений по степени важности:

– FAIL — устранить обязательно;

– WARN — не блокирует развёртывание сейчас, но означает риск потери данных/падения после перезагрузки. Решение — смонтировать отдельный раздел и добавить его в /etc/fstab;

Запускать `run_prostor.sh` можно с предупреждениями WARN, но их требуется зафиксировать в отчёте.



1.4.3. Шаг 3. Запуск развертывания

Скрипт: run_prostor.sh

Команда: sudo ./run_prostor.sh hostinfo<C>-<D>

Назначение - оркестратор развёртывания: запускает скрипты ключевых фаз, проводит предварительные проверки параметров сценария развёртывания, ведёт общий лог logs/run_prostor_<timestamp>.log и прерывает работу при первой же ошибке.

Требования:

- запуск от root/sudo;
- путь к распакованной папке hostinfo<C>-<D> — первый аргумент.

Что делает оркестратор:

1. открывает лог logs/run_prostor_<timestamp>.log и дублирует вывод на экран (tee);
2. проверяет входные данные (hostinfo, services.tsv, extract/, deploy_prostor, nats-amd64.tar.gz, prostor_backup_*.tar.gz);
3. последовательно выполняет 5 фаз;
4. при ошибке любой фазы производится остановка развертывания;
5. в конце выводится отчет об установке в пункте SUMMARY.

Критерий успеха:

SUMMARY

destroy_services OK

deploy_prostor -d OK

deploy_prostor OK

deploy_services OK

install_services OK

и в фазе install_services:

Итог: OK=17, FAIL=0

Если фаза завершилась FAIL — развёртывание не считается завершённым.

Необходимо:

1. открыть лог соответствующей фазы;
2. вызвать команду состояния сервиса: `systemctl status <Service>` и `journalctl -u <Service> -n 200 --no-pager`;
3. устранить причину и запустить `run_prostor.sh` заново.

1.4.4. Шаг.4 Контроль результатов

После успешного установки рекомендуется:

1. повторно запустить `check_server.sh` и убедиться, что:
 - порты 4222/8222/5432 — LISTEN;
 - все сервисы active/enabled;
 - `/etc/<Service>` присутствует у каждого сервиса.

2. проверить БД `prostor`:

```
sudo -u postgres psql -h 127.0.0.1 -U postgres -d prostor -c "SELECT count(*)  
FROM users;"
```

```
sudo -u postgres psql -h 127.0.0.1 -U postgres -d prostor -c "SELECT spcname  
FROM pg_tablespace WHERE spcname='video';"
```

3. зафиксировать путь к логу `run_prostor_<timestamp>.log` и к MANIFEST-отчёту `/root/backups/prostor/<timestamp>_MANIFEST.txt`.

1.4.5. Процедура отмены инсталляции

Полное удаление инсталляции «Простор» выполняется командой:

```
sudo ./run_prostor.sh -d
```

Эта команда удаляет прикладные сервисы, NATS, PostgreSQL и базу данных «Простор». Применяется в том числе при неудачном завершении фазы `install_services`, если принято решение не восстанавливать текущую инсталляцию, а переустановить её заново.



1.4.6. Логирование

Основной лог установки — в каталоге logs, расположенном в том же каталоге, что и run_prostor.sh, имя файла: run_prostor_<timestamp>.log.

Логи прикладных сервисов — стандартные для systemd (journalctl -u <Service>).

1.5. Описание шагов развертывания

Анализ по эталонному логу успешной установки.

При успешной установки отчет в SUMMARY выводит (итог OK=17, FAIL=0).

1.5.1. Фаза проверки и удаления предыдущих развертываний ПК «Простор»

Назначение - удалить ранее установленные прикладные сервисы и сделать бэкап.

Порядок удаления сервисов:

AuthService, CmdCtrlMgmt, DbAsync, DbClient, DbOuroboros,
DbVideoArchive, DrControlHors, DrCtrlNetping, DrONVIFController,
DrOpuPelcoDe, DrRlsZavantXml, LimanZ, meteomodbus,
OnvifCameraDriver, RTSP_proxy, Tiles_Server, VideoManager

Артефакты:

- бэкап: /root/backups/prostor
- отчёт: /root/backups/prostor/<timestamp>_MANIFEST.txt
- статус: ОК

1.5.2. Фаза удаления ядра (deploy_prostor -d)

1.5.2.1. Удаление сервиса nats_server

- Остановка сервиса.
- Удаление пакетов nats-server, natscli.
- Удаление файлов и пользователя nats.

1.5.2.2. Удаление сервиса drpo_postgres

- Удаление кластера 15/main.
- Удаление пакетов postgresql-15, postgresql-client-15, postgresql-common, postgresql-client-common.
- Очистка конфигурационных файлов (dpkg может предупредить, что /etc/postgresql-common и /etc/postgresql не пусты — это штатно).
- Проверка: пакеты удалены, /etc/postgresql удалён, пользователь postgres удалён, процессов нет.

1.5.2.3. Удаление сервиса prostordb

- Отдельного удаления нет: БД prostor удаляется вместе с PostgreSQL. Статус фазы: ОК.

1.5.3. Фаза развертывания ядра (deploy_prostor)

1.5.3.1. Развертывание сервиса nats_server

- Источник: nats-amd64.tar.gz.
- Установка nats-server 2.10.18 и natscli 0.0.35.
- Конфиг: /etc/nats-server/nats-server.conf.
- Unit: /etc/systemd/system/nats-server.service → enable → запуск.
- Проверки: ОК: 4222 слушается, ОК: monitoring 8222.

1.5.3.2. Развертывание сервиса drpo_postgres

Установить конфигурационный профиль развертывания: «prostor».

- запустить pg_hba.conf из conf/drpo_postgres/prostor/pg_hba.conf. с паролем postgres из conf/drpo_postgres/install.cfg (неинтерактивно).

Установка PostgreSQL 15, создание кластера 15/main:

Установить параметры для создания кластера:

- кодировка ru_RU.UTF-8, UTF8, text search russian, TZ Europe/Moscow;
- max_connections=100, shared_buffers=128MB;
- scram-sha-256 для host-аутентификации.

Запустить сервис, установить пароль, проверить установку протокола TCP.

Должно выдать сообщение:

– Проверка wal_level: logical — OK.

Итоговые параметры:

Host: 127.0.0.1 Port: 5432 User: postgres

1.5.3.3. Развертывание сервиса prostordb

Источник: prostor_backup_*.tar.gz (архив структуры данных БД).

Шаги развертывания производятся скриптом в следующей последовательности:

1. Создание БД prostor.
2. Создание ролей.
3. Создание tablespace video до restore.
4. pg_restore.
5. Установка прав (GRANT × 85).
6. Проверка: users=20, video_archive=0, tablespace video создан, video_archive в tablespace video.
7. Сообщение о завершении.

При ошибках в pg_restore, связанных с партиционированными таблицами и FK они помечаются в логге. Это ожидаемое поведение, оно не критично для успешной установки

Успешный статус фазы: OK.

1.5.4. Фаза развертывания сервисов (deploy_services)

Назначение - разложить файлы сервисов из extract/ по целевым путям.

Для каждого сервиса из services.tsv:

- Запустить бинарник → /opt/prostor_pp/<Service>/<Service>
- Разместить конфигурационный файл по пути →
/etc/<Service>/<Service>.cfg
- Разместить unit-файл сервиса → /lib/systemd/system/<Service>.service

В конце установки запустить команду для перезагрузки конфигурационных файлов сервиса — systemctl daemon-reload.



Статус: ОК.

1.5.5. Фаза установки сервисов (install_services)

Назначение - запустить каждый сервис через systemd.

Эталонный результат (успешная установка):

- AuthService — OK
- CmdCtrlMgmt — OK
- DbAsync — OK
- DbClient — OK
- DbOuroboros — OK
- DbVideoArchive — OK
- DrControlHors — OK
- DrCtrlNetping — OK
- DrONVIFController — OK
- DrOpuPelcoDe — OK
- DrRlsZavantXml — OK
- LimanZ — OK
- meteomodbus — OK
- OnvifCameraDriver — OK
- RTSP_proxy — OK
- Tiles_Server — OK
- VideoManager — OK

Итог: OK=17, FAIL=0, фаза ОК.

SUMMARY

destroy_services OK

deploy_prostor -d OK

deploy_prostor OK

deploy_services OK

install_services OK

Log: logs/run_prostor_<timestamp>.log

Программный комплекс «Простор» установлен успешно.