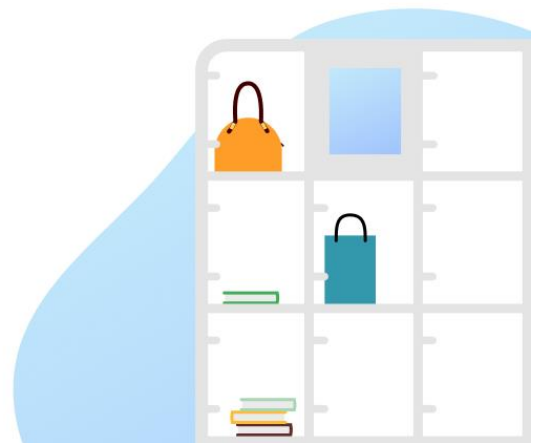
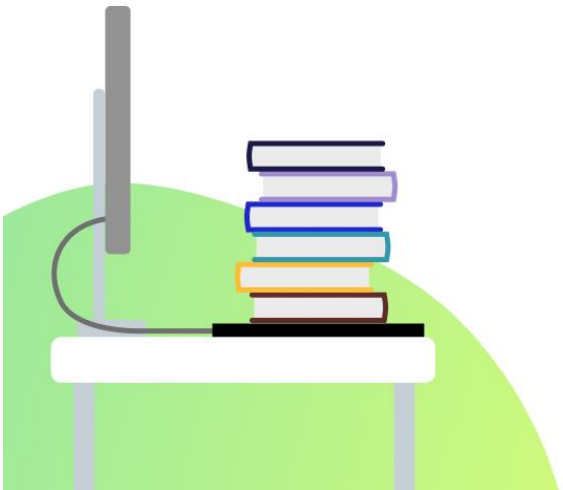




AP Admin Server^{Web}

Установка и запуск сервиса на Astra Linux 1.7 (или совместимая)

ООО «Автоматизированные решения»



1. Требования к системе

Параметр	Значение
ОС	Astra Linux 1.7 (или совместимая)
PostgreSQL	14
.NET / ASP.NET Core	Runtime 8
База данных	AdminServer_Cloud
Порт сервиса	8002

Примечание. Документ оформлен в виде пошаговой инструкции. Команды и конфигурационные блоки вынесены в отдельные блоки для быстрого копирования.

2. Подготовка файлов

2.1 Скачайте дистрибутив с сайта разработчика

```
https://ar-solutions.ru/uploads/dist/admin_server/adminserver-astra.zip
```

2.2 Распакуйте содержимое в папку на сервере

```
/opt/AdminServer
```

Команды:

```
sudo apt update
sudo apt install -y wget unzip

ARCHIVE_URL="https://ar-solutions.ru/uploads/dist/admin_server/adminserver-astra.zip"
TMP_ZIP="/tmp/AdminServer.zip"

sudo mkdir -p /opt
wget -O "$TMP_ZIP" "$ARCHIVE_URL"
sudo unzip -o "$TMP_ZIP" -d /opt/

BASE="/opt/AdminServer"
```

3. Установка PostgreSQL и развёртывание БД

3.1. Входные данные

- ОС: Astra Linux 1.7 (или совместимая).
- Нужны права пользователя с sudo.
- На диске есть папка /opt/AdminServer (см. шаг 0).
- Имя базы данных: AdminServer_Cloud.
- Пользователь БД: postgres.
- Пароль БД задаётся при установке (обозначим как <DB_PASSWORD> и используем в дальнейшем).

3.2. Установка PostgreSQL

Обновить список пакетов и установить PostgreSQL 14:

```
sudo nano /etc/apt/sources.list
```

Удаляем строку, указывающую на локальный CD-ROM

Затем нам необходимо подключить расширенный репозиторий:

```
sudo sed -i 's|^deb .*uu/last/repository-update.*|# &|' /etc/apt/sources.list /etc/apt/sources.list.d/*.list 2>/dev/null
printf '%s\n' \
'deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_x86-64 main contrib non-free' \
'deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_x86-64 astra-ce' \
```

```
sudo tee /etc/apt/sources.list.d/astra-extended.list
```

После этого можно устанавливать сам postgres

```
sudo apt-get update
sudo apt-get install -y postgresql-14 postgresql-client-14
```

Проверить, что сервер запущен:

```
sudo systemctl status postgresql
```

В статусе должно быть: active (running).

3.3. Задать пароль пользователю postgres

Пароль понадобится сервису и админ-инструментам (psql, DBeaver и т.д.).

```
sudo -u postgres psql -c "ALTER USER postgres WITH PASSWORD '<DB_PASSWORD>';"
```

Где <DB PASSWORD> — фактический пароль, например: Strong Pass_2025!.

Этот же пароль далее нужно будет указать в конфигурации сервиса.

3.4. Проверка подключения по паролю

Проверить, что можно подключиться к БД по TCP так же, как это будет делать сервис:

```
psql -h localhost -U postgres -d AdminServer_Cloud
```

PostgreSQL спросит пароль — ввести <DB PASSWORD>.

Если видно приглашение вида:

```
AdminServer_Cloud=#
```

- БД успешно развёрнута и готова к использованию сервисом.

```
sudo -u postgres psql -d AdminServer_Cloud -f /opt/AdminServer/PostgreSQL/SQL/StartPack.sql
```

Также можно использовать скрипт `deploy.sh`

Инструкция в `README.txt`

4. Установка и запуск сервиса AP Admin Server

4.1. Входные данные

- PostgreSQL установлен и настроен, база IDLogic AdminServer Cloud развёрнута.
- Пользователь БД: postgres.
- Пароль БД: <DB_PASSWORD> (тот же, что был задан командой ALTER USER postgres ...).
- На диске есть папка /opt/AdminServer
- Сервис поставляется в виде файла Service.dll по пути:

```
/opt/AdminServer/Services/AdminServer/Service/bin/Debug/Service.dll
```

- Порт сервиса по умолчанию: 8002.
- ОС: Astra Linux 1.7 (или совместимая).
- Требуется права пользователя с sudo.

4.2. Установка .NET Runtime

Для запуска `Service.dll` нужен **.NET Runtime 8:**

```
cd /tmp
sudo wget https://dot.net/v1/dotnet-install.sh -O dotnet-install.sh
sudo chmod +x dotnet-install.sh
sudo mkdir -p /usr/share/dotnet
sudo ./dotnet-install.sh --channel 8.0 --runtime aspnetcore --install-dir /usr/share/dotnet
```

Проверить наличие:

```
dotnet --info
```

В списке `runtimes` должна быть строка вида: `Microsoft.AspNetCore.App 8.0.x`.

4.3. Рабочая директория сервиса

Рабочая директория сервиса (там лежит `Service.dll` и `appsettings.json`):

```
/opt/AdminServer/Services/AdminServer/Service/bin/Debug
```

Можно убедиться:

```
ls /opt/AdminServer/Services/AdminServer/Service/bin/Debug
```

Ожидается увидеть файлы:

- `Service.dll`
- `appsettings.json`
- и другие `.dll`, необходимые для работы.

4.4. Настройка строки подключения в `appsettings.json`

Открыть конфигурационный файл сервиса:

```
sudo nano /opt/AdminServer/Services/AdminServer/Service/bin/Debug/appsettings.json
```

Найти блок:

```
"ConnectionStrings": {
  "AppDbContext": "..."
}
```

Заменить на:

```
"ConnectionStrings": {
  "AppDbContext":
  "Host=localhost;Port=5432;Database=AdminServer_Cloud;Username=postgres;Password=<DB_PASSWORD>;Include Error
  Detail=true"
}
```

Где `<DB PASSWORD>` — тот же пароль, который был задан пользователю `postgres` при установке `PostgreSQL`.

Сохранить файл (в `nano`: `Ctrl+O`, `Enter`, `Ctrl+X`).

4.5. Пробный запуск вручную (по желанию)

4.5.1 Этот шаг не обязателен, но полезен для проверки, что сервис стартует и корректно подключается к БД.

Перейти в папку сервиса:

```
cd /opt/AdminServer/Services/AdminServer/Service/bin/Debug
```

Запустить сервис на порту 8002:

```
ASPNETCORE_URLS="http://0.0.0.0:8002" /usr/share/dotnet/dotnet Service.dll
```

При успешном запуске в логах должны появиться строки вида:

```
Now listening on: http://0.0.0.0:8002
Application started. Press Ctrl+C to shut down.
[UserService] CheckDatabase: Database exists. No action applied
```

4.5.2 Не закрывая процесс, с рабочей станции в сети открыть в браузере:

```
http://<IP_сервера>:8002/swagger
```

Например, если IP сервера — 192.168.9.123:

```
http://192.168.9.123:8002/swagger
```

Если интерфейс/Swagger открывается — конфигурация верна.

Затем остановить процесс Ctrl+C.

4.6. Настройка systemd-сервиса

Чтобы сервис автоматически запускался при старте системы и работал в фоне, создаётся unit-файл systemd.

Создать файл:

```
sudo nano /etc/systemd/system/adminserver.service
```

Содержимое файла:

```
[Unit]
Description=AdminServer
After=network.target postgresql.service
[Service]
WorkingDirectory=/opt/AdminServer/Services/AdminServer/Service/bin/Debug
ExecStart=/usr/bin/dotnet Service.dll
Restart=always
RestartSec=5
Environment=ASPNETCORE_URLS=http://0.0.0.0:8002
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
[Install]
WantedBy=multi-user.target
```

(Отдельный пользователь для сервиса не создаётся, unit запускается от root, так как параметр User= не задан.)

4.7. Запуск и проверка systemd-сервиса

Загрузить новый unit и включить автозапуск:

```
sudo systemctl daemon-reload
sudo systemctl enable adminserver
```

Запустить сервис:

```
sudo systemctl start adminserver
```

Проверить статус:

```
sudo systemctl status adminserver
```

В статусе должно быть:

```
active (running)
```

При необходимости лог можно посмотреть командой:

```
journalctl -u adminserver -n 100 -f
```

4.8. Сетевые настройки и firewall (UFW)

4.8.1 Если на сервере используется UFW (стандартный фаервол Ubuntu), нужно убедиться, что доступ не блокируется.

Проверить состояние UFW:

```
sudo ufw status
```

- Если Status: inactive — фаервол выключен, дополнительных действий не требуется.
- Если Status: active — убедиться, что разрешены SSH и порт сервиса.

Минимально необходимый набор правил:

```
# Разрешить SSH-доступ (важно, чтобы не потерять доступ по ssh)
sudo ufw allow OpenSSH
# Разрешить доступ к веб-сервису Admin Server
sudo ufw allow 8002/tcp
```

4.8.2 Опционально, если требуется прямой доступ к PostgreSQL с других машин в сети:

```
sudo ufw allow 5432/tcp
```

Посмотреть текущие правила:

```
sudo ufw status
```

5. Итоговая проверка работы сервиса

5.1 Узнать IP сервера:

```
ip a
```

Например: 192.168.9.123.

5.2 С рабочей станции открыть в браузере:

```
http://192.168.9.123:8002/
http://192.168.9.123:8002/swagger
```

5.3 При успешной установке должен открываться веб-интерфейс/Swagger сервиса AP Admin Server.

На этом установка PostgreSQL, развёртывание базы и настройка сервиса AP Admin Server на Astra завершены.