

Общество с ограниченной ответственностью «ПСБ Лаб»

Программа для ЭВМ

BI Intellect

**РУКОВОДСТВО АДМИНИСТРАТОРА ПРИЛОЖЕНИЙ
(Руководство по установке)**

МОСКВА, 2025 г.

С О Д Е Р Ж А Н И Е

1.	ОСНОВНЫЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ	3
2.	ВВЕДЕНИЕ	3
3.	ИНСТАЛЛЯЦИЯ ПРЭВМ	3
4.	НАСТРОЙКА ПРЭВМ	8
5.	ДЕИНСТАЛЛЯЦИЯ ПРЭВМ	19
6.	ЭКСПЛУАТАЦИЯ ПРЭВМ	19

1. ОСНОВНЫЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ И СОКРАЩЕНИЯ

Для целей настоящего руководства администратора приложений (далее – Руководство, документ) используются следующие понятия, определения и сокращения:

БД – база данных.

Информационная система (ИС) – взаимосвязанная совокупность технических и программных средств, других объектов информационной инфраструктуры, содержащейся в базах данных информации и обеспечивающих ее обработку технологий.

ПО – программное обеспечение.

2. ВВЕДЕНИЕ

2.1. Назначение документа

Настоящее Руководство предназначено для администраторов программы для ЭВМ VI Intellect (далее - «ПрЭВМ»), которые будут устанавливать и (или) деинсталлировать, настраивать и сопровождать ПрЭВМ.

2.2. Требования к рабочему месту

Для входа в ПрЭВМ необходимо иметь один из перечисленных ниже браузеров:

- Chrome;
- Opera;
- Microsoft Edge;
- Firefox;
- Яндекс.Браузер;
- Прочие браузеры на основе Chromium.

3. ИНСТАЛЛЯЦИЯ ПрЭВМ

3.1. Пакетная установка ПО

Офлайн установка в ОС Astra Linux Orel с отдельно стоящим PostgreSQL

Перед началом установки должны быть подключены репозитории:

- Flask;
- RestrictedPython.

Требуемые версии пакетов:

- python – 3.7;
- pip – 23.2.1;
- Flask – 2.2.5;
- RestrictedPython – 6.2.

Для установки необходимо выполнить следующие действия:

1. Создать пользователя в PostgreSQL;
2. Создать БД для приложения и предоставить на них права для созданного пользователя;

```
CREATE DATABASE platform_manager_cache;  
  
GRANT ALL PRIVILEGES ON DATABASE
```

```
platform_manager_cache TO USER;
```

```
CREATE DATABASE platform_manager;
```

```
GRANT ALL PRIVILEGES ON DATABASE platform_manager TO  
USER;
```

```
CREATE DATABASE platform_notifications;
```

```
GRANT ALL PRIVILEGES ON DATABASE  
platform_notifications TO USER;
```

```
CREATE DATABASE platform_logstash;
```

```
GRANT ALL PRIVILEGES ON DATABASE platform_logstash TO  
USER;
```

```
CREATE DATABASE platform_storage;
```

```
GRANT ALL PRIVILEGES ON DATABASE platform_storage TO  
USER;
```

3. Настроить доступность PostgreSQL с сервера (BM), на котором будет установлено приложение;
4. Создать директорию для репозитория:

```
mkdir -p /srv/biintellect/repo/
```

5. Скачать архив репозитория BI_Intellect.zip в созданную директорию:

```
cd /srv/biintellect/repo/ && wget https://nexus-  
external/repository/biintellect.ru-proxy-  
raw/v1.26.0/BI_Intellect.zip
```

где v1.26.0 – версия продукта в формате x.y.z.

6. Разархивировать архив репозитория BI_Intellect.zip:

```
unzip BI_Intellect.zip
```

7. Добавить ключ для репозитория в список доверенных ключей:

```
apt-key add biintellect.key
```

8. Подключить репозиторий:

```
echo "deb file:/srv/biintellect/repo/astra/orel/ orel  
main" > /etc/apt/sources.list.d/biintellect.list
```

9. Выполнить обновление списка доступных пакетов:

```
apt update
```

10. Установить пакет для организации работы с PostgreSQL:

```
apt install biintellect-platform-external-postgresql
```

11. Включить конфигурирование БД и отредактировать параметры подключения (user, password, host) к PostgreSQL в файле postgresql.env:

```
sed -i 's/false/true'  
/etc/biintellect/platform/env.d/postgresql.env  
vim /etc/biintellect/platform/env.d/postgresql.env
```

12. Установить приложение:

```
apt install biintellect-platform -y
```

13. Проверить статус сервисов приложения:

```
for service in $(systemctl list-unit-files  
biintellect-platform* | grep .service | cut -d. -f1);  
do echo "$service is $(systemctl show -p SubState --  
value $service)"; done
```

14. Для первого входа в систему необходимо использовать стандартные данные:

Логин: **admin**
Пароль: **qwe123**

3.2. Установка клиентской части ПО

Не применимо.

3.3. Настройка Nginx

Для установки сертификата необходимо указать пути до сертификатов в файле конфигурации:

```
vim /etc/nginx/conf.d/biintellect.conf
```

```
ssl_certificate /path/to/cert/  
ssl_certificate_key /path/to/cert/
```

Для увеличения таймаутов запросов необходимо добавить параметр в файл конфигурации nginx для сервиса gateway:

```
vim /etc/nginx/biintellect/locations/gateway.conf
```

```
location /proxy/ {  
    ...  
    proxy_read_timeout <n>;  
}
```

<n> - количество секунд

3.4. Настройка Redis

1. Для корректной работы приложения необходимо произвести донастройку сервиса Manager, добавив настройки подключения к Redis в конфигурационный файл /etc/biintellect/platform/manager/config.json:

```
"redis": {  
    "address": [  
        "127.0.0.1:637",  
    ],  
    "username": "",  
    "password": "",  
    "db": 0,  
    "client_type": "regular"  
}
```

2. Выполнение требований инженеров по безопасности

Для работы с Redis необходимо учесть рекомендации инженеров по безопасности ДИБ ПСБ. Основные изменения касаются переименования команд Redis ('RENAME-COMMAND'). Для этого реализуется 2 варианта:

2.1. Указать mapper в конфигурационном файле
В секции 'redis' добавьте 'cmd' с параметром 'mapper':

```
"redis": {
  "address": [
    "127.0.0.1:6379",
  ],
  "username": "",
  "password": "",
  "db": 0,
  "client_type": "regular",
  "cmd": {
    "mapper": {
      "SAVE": "SomeUniqkeyFor_SAVE",
      "DEL": "SomeUniqkeyFor_DEL",
      "KEYS": "SomeUniqkeyFor_KEYS",
      "PEXPIRE": "SomeUniqkeyFor_PEXPIRE",
      "RENAME": "SomeUniqkeyFor_RENAME",
      "FLUSHALL": "SomeUniqkeyFor_FLUSHALL"
    }
  }
}
```

2.2. Создать mapper в отдельном файле /etc/biintellect/path_to_mapper_config.json

```
{
  "SAVE": "SomeUniqkeyFor_SAVE",
  "DEL": "SomeUniqkeyFor_DEL",
  "KEYS": "SomeUniqkeyFor_KEYS",
  "PEXPIRE": "SomeUniqkeyFor_PEXPIRE",
  "RENAME": "SomeUniqkeyFor_RENAME",
  "FLUSHALL": "SomeUniqkeyFor_FLUSHALL"
}
```

И в конфигурационном файле /etc/biintellect/platform/manager/config.json Добавить атрибут redis.cmd.file_path

```
"redis": {
  "address": [
    "127.0.0.1:6379",
  ],
  "username": "",
  "password": "",
  "db": 0,
  "client_type": "regular",
  "cmd": {
    "file_path":
"/etc/biintellect/path_to_mapper_config.json"
  }
}
```

```
}
```

2.3. Настройка TLS:

```
"redis": {
  "address": [
    "127.0.0.1:6379",
  ],
  "username": "",
  "password": "",
  "db": 0,
  "client_type": "regular",
  "cmd": {
    "file_path":
"/etc/biintellect/path_to_mapper_config.json"
  },
  "tls_cert_file_path": "",
  "tls_key_file_path": "",
  "tls_ca_file_path": "",
  "tls_enabled": false
}
```

4. НАСТРОЙКА ПрЭВМ

4.1. Настройка серверной части

Сервисы, параметры которых открыты для конфигурирования:

- Manager;
- Dashboard;
- Gateway;
- Logstash;
- Renderer;
- Notifications;
- File-storage;
- Data transformer.

Для конфигурации доступны следующие параметры в разрезе сервисов.

4.1.1. Manager

4.1.1.1. Настройки сервиса

- `biintellect_CORE_CACHE_TYPE` – тип базы данных для работы с датасетами.

Поддерживаются значения:

- `clickhouse`;

o postgres;

- biintellect_CORE_CACHE_DSN – DSN базы данных для работы с датасетами.
- biintellect_CORE_APPLICATION_MODE_BETA – включение бета режима

По умолчанию:

false

4.1.1.2. Настройка аутентификации

- biintellect_CORE_EXPIRE_SESSION_EXPIRE_SECONDS - Срок жизни токена аутентификации

По умолчанию:

1800

- biintellect_CORE_AUTH_DEFAULT_METHOD – метод аутентификации по умолчанию.

По умолчанию:

api

Поддерживаются значения:

- o api;
- o ldap;
- o openIDConnect;
- o custom_client_rec.

4.1.1.3. Метод аутентификации API (стандартная, логин/пароль)

- biintellect_CORE_AUTH_METHOD_API_ACTIVE - Доступность метода стандартной аутентификации.

По умолчанию:

true

4.1.1.4. Метод аутентификация AD/LDAP

- biintellect_CORE_AUTH_METHOD_LDAP_ACTIVE - Доступность метода аутентификация с использованием службы каталогов

По умолчанию:

false

- biintellect_CORE_AUTH_ACTIVE_DIRECTORY_SERVER - Адрес контроллера домена
- biintellect_CORE_AUTH_ACTIVE_DIRECTORY_PORT - Порт контроллера домена
- biintellect_CORE_AUTH_ACTIVE_DIRECTORY_BASE_DN – Домен или организационная единица, где следует искать запись каталога

- `biintellect_CORE_AUTH_ACTIVE_DIRECTORY_LOGIN_KEY` - Атрибут записи каталога, значение которого следует ассоциировать с логином, указанным для аутентификации

По умолчанию:

`uid`

Поддерживаются значения:

- o `cn`;
- o `uid`

4.1.1.5. Метод аутентификация OpenID Connect

- `biintellect_CORE_OIDC_AUTH_METHOD_ACTIVE` – Доступность метода аутентификация OIDC

По умолчанию:

`false`

- `biintellect_CORE_OIDC_PROVIDER_ISSUER` - Провайдер-поставщик аутентификации
- `biintellect_CORE_OIDC_PROVIDER_REDIRECT_URL` - Адрес переадресации после аутентификации в провайдере
- `biintellect_CORE_OIDC_PROVIDER_CLIENT_ID` – Client ID, указываемый в провайдере для приложения
- `biintellect_CORE_OIDC_PROVIDER_CLIENT_SECRET` – Client secret, выдаваемый провайдером
- `biintellect_CORE_OIDC_PROVIDER_SCOPE` - Скоуп, запрашиваемый у провайдера
- `biintellect_CORE_OIDC_PROVIDER_NAME` - Произвольное имя провайдера
- `biintellect_CORE_OIDC_PROVIDER_CLAIMS_PREFERRED_USERNAME_KEY` - Ключ claims, по которому будет определено значение username пользователя в BI Intellect
- `biintellect_CORE_OIDC_PROVIDER_CLAIMS_EMAIL_KEY` - Ключ claims, по которому будет определено значение email пользователя в BI Intellect
- `biintellect_CORE_OIDC_PROVIDER_CLAIMS_FIRST_NAME_KEY` - Ключ claims, по которому будет определено значение "имя пользователя" в BI Intellect
- `biintellect_CORE_OIDC_PROVIDER_CLAIMS_LAST_NAME_KEY` - Ключ claims, по которому будет определено значение "фамилия пользователя" в BI Intellect

- `biintellect_CORE_OIDC_PROVIDER_CLAIMS_MIDDLE_NAME_KEY` - Ключ `claims`, по которому будет определено значение "отчество пользователя" в BI Intellect
- `biintellect_CORE_OIDC_INSECURE_OVERRIDE_ISSUER` - Не выполнять валидацию `issuer_url`

По умолчанию:
false

- `biintellect_CORE_OIDC_PROVIDER_ACTUAL_ISSUER` - Эндпоинт `issuer_url` для валидации `id_token` когда валидация `issuer_url` не выполняется
- `biintellect_CORE_OIDC_PROVIDER_AUTHORIZE_IAM_URL` - Адрес эндпоинта `/authorize` ЕЦП "Гостех" для получения роли пользователя

4.1.1.6. Метод Custom

- `biintellect_CORE_INTEGRATION_AUTH_METHOD_CLIENT_REC_ACTIVE` - Доступность метода

По умолчанию:
false

- `biintellect_CORE_INTEGRATION_AUTH_CLIENT_REC_HEADER_KEY` - Заголовок запроса, содержащий имя пользователя (`username`) должен приходить на POST-запрос `"/auth/custom-client-rec/login"`

4.1.1.7. Параметры логирования

- `biintellect_CORE_LOG_LVL` – Уровень логирования действий сервиса.

По умолчанию:
4

Поддерживаются значения:

- o 0 – Panic level
 - o 1 – Fatal level
 - o 2 – Error level
 - o 3 – Warning level
 - o 4 – Info level
 - o 5 – Debug level
 - o 6 – Trace level
- `biintellect_CORE_LOG_WITH_RESPONSE` – Запись в лог тела ответа

По умолчанию:
False

4.1.1.8. Параметры защиты от атаки перебором паролей

- `biintellect_CORE_BLOCKED_FAILED_IP` – максимальное количество неуспешных попыток ввода пароля с одного IP-адреса

По умолчанию:

6

- `biintellect_CORE_BLOCKED_FAILED_USER` – максимальное количество неуспешных попыток ввода пароля под одной учетной записью

По умолчанию:

6

- `biintellect_CORE_BLOCKED_DURATION_RESET_FAILED_IP` – время до сброса счетчика неуспешных попыток ввода пароля с одного IP-адреса (в секундах)

По умолчанию:

604800

- `biintellect_CORE_BLOCKED_DURATION_RESET_FAILED_USER` – время до сброса счетчика неуспешных попыток ввода пароля под одной учетной записью

По умолчанию:

604800

- `biintellect_CORE_BLOCKED_DURATION_BAN_IP` – продолжительность блокировки IP-адреса (в секундах)

По умолчанию:

1800

- `biintellect_CORE_BLOCKED_DURATION_BAN_USER` – продолжительность блокировки учетной записи (в секундах)

По умолчанию:

1800

- `biintellect_CORE_BLOCKED_DURATION_TMP_ACCESS_USER` – продолжительность временного доступа пользователя после разблокировки (в секундах)

По умолчанию:

86400

- `biintellect_CORE_BLOCKED_DURATION_BAN_USER_FROM_DB` – продолжительность блокировки учетной записи из БД (в секундах)

По умолчанию:

300

- `biintellect_CORE_BLOCKED_ENABLE_CHECK_BLOCKED_IP` – включение проверки блокировки по IP-адресу

По умолчанию:

true

4.1.1.9. Параметры блокировки неактивных учетных записей

- `biintellect_CORE_EXPIRE_INACTIVE_USER_BLOCK_DAYS` – количество дней неактивности, после которого УЗ пользователя блокируется

По умолчанию:

90

- `biintellect_CORE_JOB_USER_ACTIVITY_CHECK_PERIOD_SECONDS` – периодичность проверки учетных записей на неактивность (в секундах)

По умолчанию:

86400

4.1.1.10. Параметры доступа к встроенным инструментам кастомизации

- `biintellect_CORE_USER_SCRIPTS_THEME_EDITOR_ENABLED` – доступность в разделе “Роли” пункта “Доступ к CSS/JS редактору” для тем проектов. При установлении значения false у всех пользователей, кроме Суперпользователя, отзывается доступ к CSS/JS редактору.

По умолчанию:

true

- `biintellect_CORE_USER_SCRIPTS_HTML_SETTING_ITEM_EDITOR_ENABLED` – доступность в разделе “Роли” пункта “Доступ к HTML-редактору” для проектов. При установлении значения false у всех пользователей, кроме Суперпользователя, отзывается доступ к HTML-редактору.

По умолчанию:

true

4.1.1.11. Параметры завершения неактивной сессии пользователя

- `biintellect_CORE_SESSION_DATA_INACTIVE_DURATION_TIME` – время неактивности пользователя, после которого сессия должна быть завершена автоматически (в секундах)

По умолчанию:

900

- `biintellect_CORE_SESSION_DATA_HEARTBIT_TIME` – периодичность отправки на backend информации об активности пользователя (в секундах)

По умолчанию:

300

- `biintellect_CORE_SESSION_DATA_USE_SESSION_INACTIVE_ALGORITHM` – параметр, определяющий, применять ли алгоритм по контролю неактивной сессии

По умолчанию:

`true`

- `biintellect_CORE_SESSION_DATA_USE_IS_REMEMBER` – использовать ли кнопку "Запомнить" в окне аутентификации пользователя. Если значение параметра `biintellect_CORE_SESSION_DATA_USE_SESSION_INACTIVE_ALGORITHM` установлено `"true"`, то значение данного параметра игнорируется, кнопка "Запомнить" не используется.

По умолчанию:

`true`

4.1.2. Dashboard

4.1.2.1. Параметры логирования

- `biintellect_DASHBOARD_LOG_LVL` – Уровень логирования действий сервиса

По умолчанию:

`4`

Поддерживаются значения:

- o `0` – Panic level
- o `1` – Fatal level
- o `2` – Error level
- o `3` – Warning level
- o `4` – Info level
- o `5` – Debug level
- o `6` – Trace level

- `biintellect_DASHBOARD_LOG_WITH_RESPONSE` – Запись в лог тела ответа

По умолчанию:

`False`

4.1.2.2. Параметры подключения к OSM-серверу

- `biintellect_DASHBOARD_MAPS_TILE_SERVER_BASE_URL` – адрес OSM сервера.

- `biintellect_DASHBOARD_MAPS_TILE_SERVER_SKIP_INSECURE` – Пропускать незащищенное http-соединение

По умолчанию:

`False`

- `biintellect_DASHBOARD_MAPS_TILE_SERVER_URL_POSTFIX` – Формат тайлов

По умолчанию:

`.png`

4.1.2.3. Параметры проекта

- `biintellect_DASHBOARD_SHARING_DISABLED` - Блокировка функциональности «Поделиться доступом».

По умолчанию:

`False`

4.1.3. Gateway

Сервис обеспечивает маршрутизацию запросов между фронтендом и остальными сервисами.

4.1.3.1. Параметры логирования

- `biintellect_GATEWAY_LOG_LVL` – Уровень логирования действий сервиса.

По умолчанию:

`4`

Поддерживаются значения:

- o 0 – Panic level
- o 1 – Fatal level
- o 2 – Error level
- o 3 – Warning level
- o 4 – Info level
- o 5 – Debug level
- o 6 – Trace level

- `biintellect_GATEWAY_LOG_PATH` – Файл для записи логов сервиса

По умолчанию:

`/tmp/platform/gateway.log`

- `biintellect_GATEWAY_PROXY_DOCUMENTATION` - Управление доступом к Swagger-клиентам сервисов (спецификациям API)

По умолчанию:

`False`

4.1.4. Logstash

Сервис обеспечивает сбор, хранение и отправку по http логов событий приложения.

4.1.4.1. Параметры логирования

- `biintellect_LOGSTASH_LOG_LVL` – Уровень логирования действий сервиса.

По умолчанию:

4

Поддерживаются значения:

- o 0 – Panic level
- o 1 – Fatal level
- o 2 – Error level
- o 3 – Warning level
- o 4 – Info level
- o 5 – Debug level
- o 6 – Trace level

4.1.4.2. Параметры сервиса

- `biintellect_LOGSTASH_REPOSITORY_PATH` – Путь до директории хранения логов

По умолчанию:

`/usr/share/biintellect/platform/logstash/`

- `biintellect_LOGSTASH_PIPELINE_CONFIG_PATH` - Путь до файла конфигурации

По умолчанию:

`/etc/biintellect/platform/service/logstash/`

4.1.5. Renderer

Сервис обеспечивает для фронтенда экспорт в изображение.

4.1.5.1. Параметры логирования

- `biintellect_RENDERER_LOG_CHROME_DRIVER_LOG_LVL` - Уровень логирования действий сервиса

По умолчанию:

ALL

- `biintellect_RENDERER_LOG_CHROME_DRIVER_LOG_PATH` - Файл для записи логов сервиса

По умолчанию:

`/tmp/renderer.log`

4.1.5.2. Параметры сервиса

- `biintellect_RENDERERER_DRIVER_WEB_DRIVER_POOL_WARM_START` – начальный размер активных соединений.

По умолчанию:

10

- `biintellect_RENDERERER_DRIVER_WEB_DRIVER_POOL_MAX_IDLE_SIZE` – Максимальный размер пула соединений

По умолчанию:

20

- `biintellect_RENDERERER_DRIVER_RENDER_TIMEOUT_SEC` – таймаут.

По умолчанию:

300

4.1.6. Notifications

Сервис обеспечивает для фронтенда работу функциональности «Уведомления».

4.1.6.1. Параметры логирования

- `biintellect_NOTIFICATIONS_LOG_LVL` – Уровень логирования действий сервиса.

По умолчанию:

4

Поддерживаются значения:

- o 0 – Panic level
- o 1 – Fatal level
- o 2 – Error level
- o 3 – Warning level
- o 4 – Info level
- o 5 – Debug level
- o 6 – Trace level

4.1.7. File-storage

Сервис обеспечивает для фронтенда хранение статических файлов (компоненты, изображения, видео).

4.1.7.1. Параметры логирования

- `biintellect_STORAGE_LOG_LVL` – Уровень логирования действий сервиса.

По умолчанию:

4

Поддерживаются значения:

- o 0 – Panic level
- o 1 – Fatal level
- o 2 – Error level
- o 3 – Warning level
- o 4 – Info level
- o 5 – Debug level
- o 6 – Trace level

4.1.7.2. Параметры сервиса

- `biintellect_STORAGE_ASSETS_STATIC_PATH` - Путь до директории хранения статичных файлов

По умолчанию:

`/usr/share/biintellect/platform/file-storage/assets/`

4.1.8. data-transformer

Сервис обеспечивает для фронтенда работу расчетных колонок датасета

4.1.8.1. Параметры логирования

- `biintellect_DATA_TRANSFORMER_LOG_LVL` – Уровень логирования действий сервиса.

По умолчанию:

4

Поддерживаются значения:

- o 0 – Panic level
- o 1 – Fatal level
- o 2 – Error level
- o 3 – Warning level
- o 4 – Info level
- o 5 – Debug level
- o 6 – Trace level

4.1.8.2. Параметры Python-скрипта

- `biintellect_DATA_TRANSFORMER_PY_SCRIPT_PYTHON_PATH` - Путь до интерпретатора Python

По умолчанию:

`/usr/share/biintellect/platform/data-transformer/python/venv/bin/python`

4.2. Настройка клиентской части

Сервисы, параметры которых открыты для конфигурирования:

- Manager;

- Dashboard;

4.2.1. Manager

4.2.1.1. Параметры датасета

- `biintellect_CORE_DATASETS_RIGHTS_TO_DATA_FOR_ALL_USERS` - Устанавливает глобальное правило ограничения данных по умолчанию при создании датасета.

По умолчанию:

`allow-all`

Поддерживаются значения:

- o `allow-all` – по умолчанию устанавливается правило «Разрешить всем», аналогично отсутствию параметра;
- o `restrict-all` – (а также любое значение отличное от `allow-all`) по умолчанию устанавливается правило «Запретить всем»

4.2.2. Dashboard

4.2.2.1. Параметры проекта

- `biintellect_DASHBOARD_SHARING_DISABLED` - Блокировка функциональности «Поделиться доступом».

По умолчанию:

`False`

5. Деинсталляция ПрЭВМ

Для деинсталляции BI Intellect выполните следующие команды:

```
systemctl stop biintellect-platform-*
dpkg -l | grep platform | \
awk '{print $2}' | \
xargs apt remove -y
```

6. Эксплуатация ПрЭВМ

6.1. Резервное копирование / восстановление

6.1.1. Резервное копирование

Для резервного копирования необходимо выполнить следующие действия:

1. Остановить приложение:

```
systemctl stop biintellect-platform-*
```

2. Создать директории для резервных копий:

```
mkdir -p /srv/backup/$(date +"%F")/etc  
mkdir -p /srv/backup/$(date +"%F")/usr/share
```

3. Скопировать рабочие директории:

```
cp -pr /etc/{nginx,biintellect} /srv/backup/$(date  
+"%F")/etc  
  
cp -pr /usr/share/biintellect /srv/backup/$(date  
+"%F")/usr/share
```

4. При необходимости переноса заархивировать резервную копию:

```
tar -C /srv/backup/$(date +"%F") -czf  
/srv/backup/$(date +"%F").tar.gz {etc,usr}
```

5. Произвести резервное копирование БД PostgreSQL

6.1.2. Восстановление

Для восстановления необходимо выполнить следующие действия:

1. Остановить приложение:

```
systemctl stop biintellect-platform-*
```

2. Удалить рабочие директории:

```
rm -r /etc/biintellect /usr/share/biintellect  
rm -r /etc/nginx - при необходимости
```

3. Скопировать рабочие директории из резервной копии:

```
cp -pr /srv/backup/<date>/etc/biintellect /etc  
cp -pr /srv/backup/<date>/usr/share/biintellect
```

```
/usr/share
```

```
cp -pr /srv/backup/<date>/etc/nginx /etc - при  
необходимости
```

4. Произвести восстановление БД PostgreSQL

6.2. Обновление

Для того, чтобы при не успешном обновлении не потерять уже имеющиеся данные, необходимо сделать резервное копирование БД и конфигурационных файлов приложения.

Для обновления необходимо выполнить следующие действия:

1. Скачать архив репозитория в директорию /srv/biintellect/repo (старый архив удалить или переместить в директорию для дистрибутива/резервных копий):

```
cd /srv/biintellect/repo/ && wget https://nexus-  
external/repository/biintellect.ru-proxy-  
raw/v1.26.0/BI_Intellect.zip
```

где v1.26.0 – версия продукта в формате x.y.z.

2. Разархивировать архив нового репозитория:

```
cd /srv/biintellect/repo/  
  
rm -r astra  
  
unzip BI_Intellect.zip
```

3. Выполнить обновление списка доступных пакетов:

```
apt update
```

4. Остановить приложение:

```
systemctl stop biintellect-platform*
```

5. Выполнить обновление пакетов приложения:

```
apt install --only-upgrade biintellect* (Y –  
согласиться на версию из пакета новой версии)
```

6. Запустить приложение:

```
systemctl restart $(systemctl list-unit-files  
biintellect-platform* | grep .service | cut -d. -f1)
```

7. При необходимости добавить в конфигурационные файлы приложения нестандартные параметры. После редактирования конфигурации необходимо перезапустить сервис:

```
systemctl restart biintellect-platform-<service>
```

8. Проверить статус сервисов приложения:

```
for service in $(systemctl list-unit-files  
biintellect-platform* | grep .service | cut -d. -f1);  
do echo "$service is $(systemctl show -p SubState --  
value $service)"; done
```

После обновления в связи с миграциями на запуск приложения может понадобится больше времени, чем обычно.

В состав пакетов приложения входят такие коннекторы как:

database_clickhouse
database_mssql
database_oracle
database_postgres
file_csv
file_xlsx
biintellect_analytics
r7

Коннекторы, перечисленные далее, не присутствуют в репозитории и не устанавливаются с приложением:

database_mysql
database_vertica
biintellect_epm
file_xml
file_json

Если коннектор из этого списка был установлен в предыдущей версии,

но больше не используется, его можно удалить следующими командами:

```
systemctl stop biintellect-platform-plugin-<имя плагина>  
apt purge biintellect-platform-plugin-<имя плагина>*  
systemctl daemon-reload  
systemctl restart biintellect-platform*
```

6.3. Перенос

6.3.1. Перенос на "чистый" сервер (ВМ)

Целевой сервер (ВМ) должен соответствовать требованиям.

Для переноса приложения необходимо выполнить следующие действия:

1. Выполнить установку приложения версии, которая соответствует приложению из резервной копии
2. Остановить приложение:

```
systemctl stop biintellect-platform*
```

3. Создать директорию для резервных копий:

```
mkdir -p /srv/backup
```

4. Скачать на целевой сервер (ВМ) резервную копию приложения в созданную директорию
5. Разархивировать резервную копию:

```
tar -C /srv/backup -xzf /srv/backup/<date>.tar.gz
```

<date> - дата создания резервной копии (имя архива резервной копии)

6. Удалить рабочие директории:

```
rm -r /etc/biintellect  
rm -r /usr/share/biintellect  
rm -r /etc/nginx - при необходимости
```

7. Переместить рабочие директории из резервной копии:

```
mv /srv/backup/etc/biintellect /etc
mv /srv/backup/usr/share/biintellect /usr/share
mv /srv/backup/etc/nginx /etc - при необходимости
```

8. Выполнить восстановление баз данных PostgreSQL
9. Запустить приложение:

```
systemctl restart $(systemctl list-unit-files
biintellect-platform* | grep .service | cut -d. -f1)
```

10. Проверить статус сервисов приложения:

```
for service in $(systemctl list-unit-files
biintellect-platform* | grep .service | cut -d. -f1);
do echo "$service is $(systemctl show -p SubState --
value $service)"; done
```

6.3.2. Перенос на сервер (ВМ) с установленным приложением

При необходимости сделать резервную копию установленного приложения на целевом сервере (ВМ).

Перенос возможен только при соответствии версий приложения на целевом сервере (ВМ) и сервере (ВМ), на котором была сделана резервная копия.

При несоответствии указанных версий следует выполнить обновление приложения на целевом сервере (ВМ).

Для переноса приложения необходимо выполнить следующие действия:

1. Выполнить пункты 2-10 раздела 6.3.1. Настоящего руководства.

6.4. Действия при сбое

Программным сбоем (далее – сбой) считается такое поведение приложения, которое отличается от ожидаемого. Ожидаемое поведение описано в руководствах пользователя и администратора.

Перед устранением сбоя убедитесь, что он не вызван внешними факторами. Примеры: ошибки обновления данных датасета могут быть вызваны недоступностью источника; ошибки аутентификации могут быть вызваны неправильной конфигурацией приложения, и т. п.

После того, как внешние факторы или возможные причины сбоя устранены, перезапустите приложение командой:

```
systemctl restart biintellect-platform-manager
```


Если программный сбой повторяется после перезапуска приложения, выполните *восстановление*.

Логи компонентов приложения доступны в системном журнале. Для просмотра воспользуйтесь командой:

```
journalctl -xeu biintellect-platform-<имя компонента>
```