



**ОБЩЕСТВО С ОГРАНИЧЕННОЙ  
ОТВЕТСТВЕННОСТЬЮ  
«ГАЛАКТИКА ТРАНСПОРТ»**  
ИНН 7708395045, КПП 770801001,  
ОГРН 1217700143231

Юридический адрес:  
129090, Москва, ул. Каланчевская,  
д. 20, стр. 5, пом. 1, 1А  
Генеральный директор – Поздеев Александр  
Владимирович  
e-mail: info@galaxy-team.com

**Установка и настройка  
программы для ЭВМ «Цифровая производственная платформа  
управления производственными процессами хозяйства автоматики и  
телемеханики «Эйлер» (ЦПП Ш «Эйлер»)).  
Версия 1.0**

## Оглавление

1	Общие положения.....	3
2	Перечень сокращений .....	4
3	Установка и настройка программы для ЭВМ.....	5
3.1	Требования по аппаратной и программной конфигурации оборудования .....	5
3.1.1	Аппаратная конфигурация.....	5
3.1.2	Программная конфигурация .....	7
3.2	Состав комплекса.....	7
3.3	Установка и настройка сервера БД .....	7
3.3.1	Настройка дисков под установку СУБД PostgreSQL .....	8
3.3.2	Настройка дисков под создание бекапов .....	8
3.3.3	Инсталляция СУБД PostgreSQL .....	8
3.3.4	Установка и настройка кластера БД в режиме master-slave .....	9
3.3.5	Восстановление бекапа системы.....	11
3.3.6	Настройка фаерволла .....	12
3.3.7	Настройка бекапирования на серверах кластера PostgreSQL .....	13
3.4	Установка и настройка сервера приложений .....	16
3.4.1	Общие требования к серверам приложений .....	17
3.4.2	Копирование дистрибутива.....	17
3.4.3	Установка Java JDK и Java JRE .....	18
3.4.4	Установка сервера приложений WildFly и инициализация сервиса WILDFLY.....	18
3.4.5	Настройка директории для хранения статических файлов .....	18
3.4.6	Настройка фаерволла на каждом узле кластера .....	18
3.4.7	Настройка соединения с СУБД .....	18
3.4.8	Запуск службы WILDFLY .....	19
3.4.9	Настройка сетевого ресурса для хранения вложений и доступа к нему по протоколу NFS .....	19
3.4.10	Настройка клиента NFS на каждом узле кластера .....	19
3.4.11	Установка и настройка на веб-сервере программного обеспечения - балансировщика нагрузки HAProxy .....	20
3.4.12	Настройка параметров в конфигурационных файлах узлов кластера сервера приложений .....	21
3.5	Вход в программу .....	25
3.6	Завершение работы.....	26

## **1 Общие положения**

Настоящий документ содержит описание установки и настройки программы для ЭВМ «Цифровая производственная платформа управления производственными процессами хозяйства автоматики и телемеханики «Эйлер» (ЦПП Ш «Эйлер»)).

## 2 Перечень сокращений

IP-адрес	Уникальный сетевой адрес узла в компьютерной сети
PostgreSQL	Свободная объектно-реляционная система управления базами данных (СУБД)
БД	База данных
ПО	Программное обеспечение
СУБД	Система управления базами данных
ЭВМ	Электронно-вычислительная машина

### **3 Установка и настройка программы для ЭВМ**

#### **3.1 Требования по аппаратной и программной конфигурации оборудования**

##### **3.1.1 Аппаратная конфигурация**

###### **3.1.1.1 Сервер БД**

Рекомендуется использовать кластер высокой доступности, состоящий из 2-х узлов. Узлы кластера расположены на отдельных серверах, имеющих следующую конфигурацию:

- Процессор: 8 ядер;
- Память: 32 Гб;
- Дисковое пространство: 3 Тб.

Дисковое пространство сервера базы данных должно быть разбито следующим образом:

- Для хранения операционной системы должен быть выделен системный раздел (размер: 50 Гб);
- Для хранения файлов СУБД PostgreSQL 15 должен быть выделен раздел данных размером: 950 Гб;
- Для хранения 2-х последних дампов системы должен быть выделен раздел данных в 2 Тб.

Рекомендуется использовать кластер серверов приложений, состоящий из 2-х и более узлов, находящихся на различных физических серверах.

Каждый узел кластера серверов приложений должен иметь следующую конфигурацию:

- Процессор: 4 ядра;
- Память: 8 Гб;
- Дисковое пространство: 120 Гб.

Дисковое пространство сервера приложений содержит 2 раздела: хранения операционной системы (50 Гб), сервера приложений, файлов настройки сервера-приложений, журналов работы (логов) сервера приложений (70 Гб).

Также к каждому узлу кластера должно быть подключено внешнее файловое хранилище по протоколу NFS размером 2 Тб для хранения, приложенных к объектам системы электронных документов.

### **3.1.1.2 Веб - сервер**

Для веб-сервера, выполняющего функцию балансировщика нагрузки между узлами кластера, необходима следующая конфигурация:

- Процессор: 4 ядра;
- Память: 8 ГБ;
- Дисковое пространство: 100 ГБ.

Дисковое пространство веб-сервера содержит 2 раздела: хранения операционной системы (50 ГБ), файлов настройки веб-сервера, журналов работы (логов) и другое (50 ГБ).

### **3.1.1.3 Интеграционный сервер**

Интеграционный сервер должен иметь следующую конфигурацию:

- Процессор: 8 ядер;
- Память: 16 ГБ;
- Дисковое пространство: 300 ГБ.

С учётом местных условий конфигурация сервера может быть модифицирована. Дисковое пространство сервера базы данных должно быть разбито следующим образом:

- Для хранения операционной системы должен быть выделен системный раздел (размер: 50 ГБ);
- Для сервера интеграции, файлов настройки сервера-интеграции, журналов работы (логов) сервера интеграции, должен быть выделен раздел размером: 250 ГБ.

### **3.1.1.4 Рабочие места (клиентская часть)**

В качестве рабочих мест необходимо использовать компьютеры со следующими характеристиками:

- Процессор: не менее 2,8 ГГц;
- Монитор: разрешение от 1920x1080;
- Клавиатура: стандартная, англо-русская;
- Манипулятор «Мышь»: стандартная;
- Оперативная память: не менее 2048 Мб;
- Жесткий диск (HDD): не менее 40 ГБ;
- Устройство бесперебойного питания: любое;

- Сетевая плата: 100 Мбит/с.

### **3.1.2 Программная конфигурация**

#### **3.1.2.1 Требования к серверу БД**

На сервере БД должны быть установлены:

- Операционная система РЭД ОС;
- PostgreSQL 15

#### **3.1.2.2 Требования к серверу приложений**

На сервере приложений должна быть установлена операционная система РЭД ОС.

#### **3.1.2.3 Требования к интеграционному серверу**

На интеграционном сервере должна быть установлена операционная система РЕД ОС.

#### **3.1.2.4 Требования к Рабочим местам (клиентская часть)**

Для корректной работы приложения необходимо, чтобы на рабочем месте пользователя был установлен Web-браузер Яндекс.Браузер версии 23 и выше.

### **3.2 Состав комплекса**

В состав комплекса входит системное ПО, установку которого необходимо выполнять из централизованного распределенного репозитория для операционной системы РЭД ОС.

В состав поставляемого комплекса входят следующие компоненты:

- ApplicationServer – дистрибутив Java JRE, JDK и сервера приложений;
- LoadBalancer – дистрибутив балансировщика нагрузки между серверами приложений;
- DatabaseBackup – дампы пустой базы данных;
- ManufUnitImport – дистрибутив сервера интеграции;
- Documentation – документация.

### **3.3 Установка и настройка сервера БД**

Установка и настройка сервера БД состоит из следующих шагов:

- Настройка дисков под установку СУБД PostgreSQL;
- Настройка дисков под создание бекапов;

- Установка СУБД PostgreSQL;
- Установка и настройка кластера БД в режиме master-slave;
- Восстановление бекапа системы;
- Настройка фаерволла;
- Настройка бэкапирования на серверах кластера PostgreSQL.

### 3.3.1 Настройка дисков под установку СУБД PostgreSQL

База данных и каталог с backup прикладной системы под управлением СУБД PostgreSQL должны содержаться в отдельных созданных каталогах на независимых физических дисках:

Директория для хранения БД должна иметь объем 950 Гб (объем, достаточный для того, чтобы вмещать в себя корневой каталог БД).

Файлы данных БД PostgreSQL разместить в дочерней директории каталога:

- /pgdata/data/ – для данных БД;
- /pgdata/tmp/ – директория хранения lock-файлов кластера БД;
- /pgdata/backup/ – директория для хранения backup файлов БД.

Дополнительных требований к настройкам ОС, отличных от установленных по умолчанию, не предъявляется.

### 3.3.2 Настройка дисков под создание бэкапов

Директории с БД и backup должны располагаться на отдельных смонтированных файловых системах:

- VG – vg\_pgdata;
- LV – lvpdata точка монтирования /pgdata/ файловой системы ext4/pgdata/;
- VG – vg\_pgbackup;
- LV - lvpbackup точка монтирования /pgdata/backups файловой системы ext4.

Каталог с backup должен иметь объем 2 Тб (объем, достаточный для того, чтобы вмещать в себя дампы за 2-е последних суток).

### 3.3.3 Установка СУБД PostgreSQL

Установите из официального репозитория СУБД PostgreSQL.

```
# dnf install -y postgresql15-server
```

Инициализируйте базу данных

```
# sudo postgresql-15-setup initdb
```

Установите необходимые настройки в файле postgresql.conf согласно таблице:



Параметр	Значение	Описание
listen_addresses	*,	Задаёт адреса TCP/IP, по которым сервер будет принимать подключения клиентских приложений. Элемент * обозначает все имеющиеся IP-интерфейсы
port	5432	TCP-порт, открываемый сервером
wal_level	replica	Настройка необходимая при использовании кластера
max_wal_senders	3	Задаёт максимальное число слотов репликации, которое сможет поддерживать сервер
hot_standby	on	Определяет, можете ли Вы подключиться для выполнения запросов к серверу во время восстановления
max_connections	300	Определяет максимальное число одновременных подключений к серверу БД
shared_buffers	6GB	Задаёт объём памяти, который будет использовать сервер баз данных для буферов в разделяемой памяти
effective_cache_size	18GB	Определяет представление планировщика об эффективном размере дискового кеша, доступном для одного запроса
maintenance_work_mem	1536MB	Задаёт максимальный объём памяти для операций обслуживания БД, в частности VACUUM, CREATE INDEX и ALTER TABLE ADD FOREIGN KEY
checkpoint_completion_target	0.7	Задаёт целевое время для завершения процедуры контрольной точки, как коэффициент для общего времени между контрольными точками
wal_buffers	16MB	Объём разделяемой памяти, который будет использоваться для буферизации данных WAL, ещё не записанных на диск
default_statistics_target	100	Устанавливает целевое ограничение статистики по умолчанию, распространяющееся на столбцы, для которых командой ALTER TABLE SET STATISTICS не заданы отдельные ограничения.

### 3.3.4 Установка и настройка кластера БД в режиме master-slave

Необходимо выполнить настройку ресурсов кластера, согласно данным настройкам:

#### ***PostgreSQL команды:***

```
pcs property set no-quorum-policy="ignore"
pcs property set stonith-enabled="false"
pcs resource defaults resource-stickiness="1000"
pcs resource defaults migration-threshold="3"
pcs resource create vip-master IPaddr2 \
```

```

ip="x.x.x.x"\
nic="eno16777984"\
cidr_netmask="24"\
op start timeout="60s" interval="0s" on-fail="restart"\
op monitor timeout="60s" interval="10s" on-fail="restart"\
op stop timeout="60s" interval="0s" on-fail="block"
pcs resource create postgresql postgresql \
pgctl="/usr/postgresql-15/bin/pg_ctl"\
psql="/usr/postgresql-15/bin/psql"\
pgdata="/pgdata/data/"\
rep_mode="sync"\
node_list="name1 name2"\
primary_conninfo_opt="keepalives_idle=60 keepalives_interval=5
keepalives_count=5
password="pass"\
repuser=replica replication_slot_name=standby_slot2 \
master_ip="x.x.x.x"\
op start timeout="60s" interval="0s" on-fail="restart"\
op monitor timeout="60s" interval="4s" on-fail="restart"\
op monitor timeout="60s" interval="3s" on-fail="restart" role="Master"\
op promote timeout="60s" interval="0s" on-fail="restart"\
op demote timeout="60s" interval="0s" on-fail="stop"\
op stop timeout="60s" interval="0s" on-fail="block"\
op notify timeout="60s" interval="0s"
pcs resource master msPostgresql postgresql \
master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true
pcs resource group add master-group vip-master
pcs constraint colocation add master-group with Master msPostgresql INFINITY
pcs constraint order promote msPostgresql then start master-group symmetrical=false
score=INFINITY
pcs constraint order demote msPostgresql then stop master-group symmetrical=false
score=0

```

**vip-master** – название кластерного ресурса виртуального IP адреса

**ip=x.x.x.x** – кластерный IP адрес, не может совпадать ни с 1 IP адресом кластера и должен быть свободен для использования.

**nic="eno16777984"** – имя сетевого интерфейса, должно совпадать на обеих нодах кластера, можно посмотреть набрав команду «*ip a*».

**cidr\_netmask="24"** – маска подсети, в которой предполагается использовать кластерный ip-адрес.

**node\_list="name1 name2"** – имена узлов используемое внутри кластера, рекомендуется задавать pg1 и pg2, предварительно добавив их в файл /etc/hosts.

**password=pass** – пароль от пользователя replica, необходимо для репликации.  
**master\_ip="x.x.x.x"** – кластерный IP адрес.

### 3.3.5 Восстановление бекапа системы

Необходимо восстановить дампы базы данных PostgreSQL, для этого выполняем следующие действия

1) Проверяем статус службы PostgreSQL и запускаем ее.

```
sudo systemctl status postgresql-15.service  
sudo systemctl start postgresql-15.service
```

2) Создаем пользователей и пустую БД

```
psql -U postgres  
CREATE USER system_user WITH LOGIN SUPERUSER INHERIT  
NOCREATEDB NOCREATOROLE NOREPLICATION PASSWORD  
'StrongPassword123!' VALID UNTIL 'infinity';  
CREATE USER template_user WITH LOGIN SUPERUSER INHERIT  
NOCREATEDB NOCREATOROLE NOREPLICATION PASSWORD  
'StrongPassword123!' VALID UNTIL 'infinity';  
CREATE USER fkv39o1z_user WITH LOGIN NOSUPERUSER INHERIT  
NOCREATEDB NOCREATOROLE NOREPLICATION PASSWORD  
'StrongPassword123!';  
  
create database tsppsh_db;
```

Выходим из psql комбинацией клавиш Ctrl+D

3) Загружаем бекап во временный каталог /tmp и распаковываем архив с бекапом БД а также и перенаправляем вывод в СУБД

```
gunzip -c /tmp/tsppsh_PROD_pg-tsppsh_db__2025_07_18_15_19.gz | psql -U  
postgres -d tsppsh_db
```

4) Предоставляем доступ созданным пользователям к базе данных

```
psql -U postgres  
\c demo_db;  
ALTER ROLE system_user SET search_path TO SYSTEM;  
ALTER ROLE template_user SET search_path = template, SYSTEM;
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA template to
template_user;
GRANT USAGE ON SCHEMA SYSTEM TO template_user;
GRANT SELECT ON ALL TABLES IN SCHEMA SYSTEM TO template_user;
ALTER ROLE fkv39o1z_user SET search_path = fkv39o1z, SYSTEM;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA fkv39o1z to
fkv39o1z_user;
GRANT USAGE ON SCHEMA SYSTEM TO fkv39o1z_user;
GRANT SELECT ON ALL TABLES IN SCHEMA SYSTEM TO fkv39o1z_user;
Выходим из psql комбинацией клавиш Ctrl+D
```

### ***PostgreSQL команды:***

psql -f dump - новая база должна быть создана из template0 (--create это учитывает), заранее должны быть созданы роли и табличные пространства, после восстановления имеет смысл выполнить ANALYZE

### ***Восстановление БД из pg\_dump***

```
pg_restore -d db dump - прямое соединение с базой db и ее восстановление.
pg_restore dump | psql - передача данных в psql
pg_restore --list dump >db.list - получить список объектов (оглавление резервной
копии)
pg_restore --use-list=db.list -d db dump - восстановление, с использованием
отредактированного в ручную файла.
pg_restore -d db --jobs=N dump – восстанавливает БД в несколько потоков. Доп.
параметры к pg_restore
--clean - включает удаление объектов базы, если они уже присутствуют до
восстановления
--create - предварительно создает ту базу, которая резервировалась, а не из -d
--table=tbl -таблицы по шаблону
--index=idx - индексы
--function=fun - функции
--trigger=trg - триггеры
--schema=scm - схемы со всеми объектами
```

### **3.3.6 Настройка фаерволла**

Выполнить настройку фаерволла:

```
# sudo firewall-cmd --zone=public --add-port=22/tcp --permanent
# sudo firewall-cmd --zone=public --add-port=5432/tcp --permanent
```

```
# sudo firewall-cmd --reload
```

### 3.3.7 Настройка бекапирования на серверах кластера PostgreSQL

#### 3.3.7.1 Требования к локальному хранению резервных копий

Полное резервное копирование базы данных должно производиться не реже одного раза в сутки, во время наименьшей загрузки сервера. Время наименьшей нагрузки определяет системный администратор самостоятельно по результатам наблюдения за работой системы. Необходимо регулярно контролировать систему и планировать её рост.

Администратор обязан следить за выходом обновлений операционной системы, ее компонентов, служб, и производить их своевременную установку.

Установку и обновление программного обеспечения, в том числе антивирусов, производить с согласования разработчиков программного обеспечения.

Необходимо документально подтверждать все особенности работы базы данных, в том числе изменение аппаратной и программной конфигурации, установку и обслуживание, а также модификацию приложений. Эти сведения необходимы для восстановления системы. В случае сбоя или потери данных документирование хронологии событий позволит быстро и точно определить источник неполадки и восстановить работоспособность системы.

Требования к локальному хранению резервных копий:

Сервер	Объект резервного копирования	Периодичность	Количество копий
Сервер БД#1	Полный бэкап базы данных	Еженедельно по четвергам в 23:00	2
	Инкрементальный бэкап БД	Ежедневно в 21:30	7
	Архив, логи	Каждые 2 часа	14
	Бекап виртуальной машины	1 раз в сутки	2
Сервер БД#2	Полный бэкап базы данных	Еженедельно по четвергам в 23:00	2
	Инкрементальный бэкап БД	Ежедневно в 21:30	7
	Архив, логи	Каждые 2 часа	14
	Бекап виртуальной машины	1 раз в сутки	2

#### 3.3.7.2 Инструкция по созданию полной резервной копии БД

Утилита `pg_dumpall` предназначена для записи («выгрузки») всех баз данных кластера PostgreSQL в один файл в формате скрипта. Этот файл содержит команды SQL, так что передав его на вход `psql`, можно восстановить все базы данных. Для

формирования этого файла вызывается `pg_dump` для каждой базы данных в кластере. `pg_dumpall` также выгружает глобальные объекты, общие для всех баз данных, то есть роли и табличные пространства.

`pg_dumpall -f dump` -сохраняет все, включая роли, табличные пространства, запуск осуществляется от суперпользователя.

Доп. параметры:

`--clean` - включает удаление объектов базы, если они уже присутствуют до восстановления

`--globals-only` - выгружает только роли и табличные пространства

### 3.3.7.3 Инструкция по созданию резервной копии отдельных объектов БД

#### *Резервирование на уровне таблиц:*

Копирование таблицы в файл:

`copy tbl to 'file';`

Вывод запроса на консоль формат (text, csv, binary):

`copy (select * from tbl) to stdout (format text, delimiter ',', null '<null>');`

#### *Резервирование при помощи утилиты pg\_dump:*

Программа `pg_dump` выгружает только одну базу данных. Чтобы выгрузить весь кластер или сохранить глобальные объекты, относящиеся ко всем базам в кластере, например, роли и табличные пространства, воспользуйтесь программой `pg_dumpall`.

`pg_dump --table=tbl -d db` - таблицы по шаблону

`pg_dump --schema=scm -d db` - схемы со всеми объектами `pg_dump -d db` - полное резервирование на уровне БД

`pg_dump --format=custom -f dump -d db` - сжатие по умолчанию, можно выбрать объекты, которые необходимо восстановить

`pg_dump --format=directory --jobs=N -f dump -d db` - выгрузить в каталог, в несколько файлов, настройка `--jobs` - дает возможность использовать при дампе много ядер

дополнительные параметры к `pg_dump`:

`--data-only` - только DML

`--schema-only` - только DDL

`--clean` - предварительное удаление

`--compress=N` - сжатие (нужен `zlib`), `N=0..9`

`--file=dump (-f)` - сохранение скрипта в файл

`--clean` - включить команды удаления объектов

`--create` включить команду создания БД

### 3.3.7.4 Инструкция по хранению транзакционных логов

Настройка расширенного бекапирования, включает в себя помимо основной копии БД, еще и копию всех транзакционных логов. Это необходимо для возможности восстановления на любую точку во времени, либо докатки БД по транзакционным логам до точки расхождения мастер-слейв узлов:

Убедиться, что директории для бекапов созданы согласно требованиям, описанным в пункте 3 данного документа.

- 1) В директории /pgdata/backups создать директорию archive/ и archive\_xlog
- 2) Разместить скрипт fullbackupDB.sh в директорию /var/lib/pgsql/
- 3) Дать права на использование данного скрипта пользователю postgres и сделать файл исполняемым:

```
chown postgres:postgres /var/lib/pgsql/fullbackupDB.sh
chmod +x /var/lib/pgsql/fullbackupDB.sh
```

- 4) Под пользователем postgres выполнить команду crontab -e
- 5) Добавить в crontab строку

```
«0 1 * * * /var/lib/pgsql/backupDB.sh >> /var/lib/pgsql/cron.log»
Минута час день месяц день_недели /путь/к/исполняемому/файлу
```

(пример обозначает, что backup будет выполняться ежедневно в 01:00 по времени сервера).

- 6) Настроить в скрипте параметр «-mtime +5» (по умолчанию +5, это глубина хранения +5 обозначает 5 копий в директории backups/archive, то есть, 1 копия в директории backups/, остальные попадают в archive/)
- 7) Логирование backup настроено в директорию /pgdata/data/pg\_log/
- 8) Настроить конфигурационный файл /pgdata/data/postgresql.conf

```
archive_command = 'test ! -f /pgdata/backups/archive_xlog/%f && cp %p
/pgdata/backups/archive_xlog/%f'
```

- 9) Выполнить перезапуск узлов для применения параметров.

### **3.3.7.5 Инструкция по очистке устаревших транзакционных логов и бекапов**

Всё время в процессе работы PostgreSQL ведёт журнал предзаписи (WAL), который расположен в подкаталоге pg\_wal/ каталога с данными кластера баз данных.

Число файлов сегментов WAL в каталоге pg\_wal зависит от min\_wal\_size, max\_wal\_size и объёма WAL, сгенерированного в предыдущих циклах контрольных точек. Когда старые файлы сегментов оказываются не нужны, они удаляются или перерабатываются (то есть переименовываются, чтобы стать будущими сегментами в нумерованной последовательности). Если вследствие кратковременного скачка

интенсивности записи в журнал, предел `max_wal_size` превышает, ненужные файлы сегментов будут удаляться, пока система не опустится ниже этого предела.

Ручное удаление файлов из `pg_wal/` может нарушить работоспособность БД.

В случае повреждения или разрастания журналов транзакции необходимо выполнить:

1) Выключить кластер сначала на slave, потом на master

```
pcs cluster stop
```

суффикс (`--force`) выполнять на master ноде

2) Выполнить проверку процессов

```
postgres ps -ef | grep postgres
```

3) Если процессов не найдено, то выполнить `pg_controldata` указывая путь до базы `postgresql`

```
/usr/pgsql-15/bin/pg_controldata /pgdata/data/
```

нас интересуют строки:

```
Latest checkpoint's NextXID: 0/1186399159
```

```
Latest checkpoint's NextOID: 4716704
```

4) Выполнить команду `pg_resetxlog` в которой указываем `NextOID` и `NextXID` (команда выполняется из под пользователя `postgres`)

```
/usr/pgsql-15/bin/pg_resetxlog -o 4716704 -x 1186399159 -f /pgdata/data/
```

### 3.4 Установка и настройка сервера приложений

Установку программы для ЭВМ необходимо выполнять из централизованного распределенного репозитория для операционной системы РЭД ОС.

Для установки и настройки кластера серверов приложений необходимо выполнить следующие действия для каждого узла кластера:

- Копирование дистрибутива;
- Установка Java JDK и Java JRE;
- Установка сервера приложений WildFly и инициализация сервиса wildfly;
- Настройка директории для хранения статических файлов;
- Настройка фаерволла на каждом узле кластера;
- Настройка соединения с СУБД;
- Запуск службы wildfly.

Для настройки хранилища документов необходимо выполнить следующие шаги:



- Настройка сетевого ресурса для хранения вложений и доступа к нему по протоколу NFS;
- Настройка клиента NFS на каждом узле кластера.

Далее необходимо выполнить настройку балансировщика нагрузки между узлами кластера, выполнив следующие шаги:

- Установка и настройка на веб-сервере программного обеспечения - балансировщика нагрузки HAProxy.
- Настройка параметров в конфигурационных файлах узлов кластера сервера приложений.

### **3.4.1 Общие требования к серверам приложений**

На всех серверах приложений, входящих в состав Системы, должны быть установлены следующие пакеты:

```
# dnf -y install zip
# dnf -y install unzip
# dnf -y install nano
# dnf -y install wget
# dnf -y install screen
# dnf -y install telnet
# dnf -y install dejavu-fonts-common
# dnf -y install dejavu-sans-fonts
# dnf -y install dejavu-sans-mono-fonts
# dnf -y install dejavu-serif-fonts
# dnf -y install nfs-utils
# dnf -y install rpcbind
# dnf -y install libreoffice
```

На сетевом файловом хранилище должны быть установлены следующие пакеты:

```
# dnf -y install nfs-utils
# dnf -y install rpcbind
```

### **3.4.2 Копирование дистрибутива**

Скопировать из дистрибутива из папки ApplicationServer в папку /tmp одного из серверов приложений следующие файлы:

- wildfly.zip
- tsppsh\_PROD\_ipdata.tar

### 3.4.3 Установка Java JDK и Java JRE

Выполнить следующую команду:

```
#dnf install -y java-17-openjdk java-17-openjdk-devel
```

### 3.4.4 Установка сервера приложений WildFly и инициализация сервиса WILDFLY

Распаковать архив в папку /opt/

```
# unzip -v wildfly.zip -d /opt
```

Поочередно выполнить следующий набор команд:

```
# sudo ln -s /opt/wildfly-26.1.3.Final /opt/wildfly
```

```
# sudo mkdir -p /etc/wildfly/
```

```
# sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
```

```
# sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/
```

```
# sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/
```

```
# sudo chmod +x /opt/wildfly/bin/launch.sh
```

```
# sudo chmod +x /opt/wildfly/bin/standalone.sh
```

```
# sudo systemctl daemon-reload
```

```
# sudo systemctl enable wildfly
```

### 3.4.5 Настройка директории для хранения статических файлов

Создать папку /usr/IPData/static

```
# mkdir -p /usr/IPData/static
```

и скопировать в нее содержимое папки ApplicationServer/static из дистрибутива.

### 3.4.6 Настройка фаерволла на каждом узле кластера

```
# sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

```
# sudo firewall-cmd --reload
```

### 3.4.7 Настройка соединения с СУБД

В файле /opt/wildfly/standalone/configuration/system.properties задать основные параметры запуска системы (описание параметров см в разделе 2.4.6):

```
# Продуктивный контур
```

```
dbConnectionUrl=jdbc:postgresql://X.X.X.X:5432/tsppsh_db
```

```
baseURL=http://main.page
```

Х.Х.Х.Х – ip адрес сервера СУБД main.page – страница публикации системы  
Сохранить файл system.properties.

### 3.4.8 Запуск службы WILDFLY

Запустить службу wildfly:

```
# sudo systemctl start wildfly
```

### 3.4.9 Настройка сетевого ресурса для хранения вложений и доступа к нему по протоколу NFS

На сетевом файловом хранилище необходимо выполнить следующие действия:  
Инициализировать службы rpcbind и nfs-server:

```
# systemctl enable rpcbind
# systemctl start rpcbind
# systemctl enable nfs-server
# systemctl start nfs-server
```

Создать папку для хранения вложений /usr/IPDataShared и настроить учетную запись:

```
# mkdir -p /usr/IPDataShared
# groupadd -g 1999 ipshareduser # useradd -g 1999 -u 1999 ipshareduser
# chown -R 1999:1999 /usr/IPDataShared
```

Выполнить настройку NFS директории:

Открыть на редактирование файл /etc/exports

```
# nano /etc/exports
```

И добавить туда строку:

```
/usr/IPDataShared *(rw,sync,anonuid=1999,anongid=1999,all_squash)
```

Сохранить файл и выполнить команду экспорта настроек:

```
# exportfs -ra
```

Выполнить настройку фаерволла:

```
# sudo firewall-cmd --zone=public --add-port=22/tcp --permanent
# sudo firewall-cmd --zone=public --add-port=111/tcp --permanent
# sudo firewall-cmd --zone=public --add-port=2049/tcp --permanent
# sudo firewall-cmd --reload
```

### 3.4.10 Настройка клиента NFS на каждом узле кластера

Для настройки клиента NFS для подключения к сетевому ресурсу для хранения вложений необходимо:

Создать папку, в которую будет выполняться монтирование сетевой папки:

```
# mkdir -p /mnt/IPSharedData
```

Смонтировать сетевую папку:

```
# mount <fileServerIp>:/IPSharedData /mnt/IPSharedData
```

### 3.4.11 Установка и настройка на веб-сервере программного обеспечения - балансировщика нагрузки HAProxy

На сервере балансировки нагрузки дополнительно должны быть установлены следующие пакеты:

```
# dnf -y install haproxy
```

#### Настройка балансировки нагрузки между серверами приложений

Настроить балансировку нагрузки между серверами приложений при помощи ПО HAProxy в соответствии со следующими требованиями:

- Проверка доступности ресурса путем запроса css файла по адресу: <http://<appServerIP>:<port>/static/variables.css>. Недоступные ресурсы должны быть исключены из пула адресов, на которые распределяется нагрузка;
- Алгоритм балансировки static-rr.

#### Настройка потребления ресурсов

- 1) В файле /opt/wildfly/bin/standalone.conf задать значения параметров -Xms и -Xmx равными 1024 и 2048 соответственно (см выделенную строку на скриншоте ниже).

```
# Uncomment the following line to prevent manipulation of JVM options
# by shell scripts.
#
#PRESERVE_JAVA_OPTS=true
#
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms1024m -Xmx2048m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true"
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
else
    echo "JAVA_OPTS already set in environment; overriding default settings with values: $JAVA_OPTS"
fi

# Sample JPDA settings for remote socket debugging
#JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,address=8787,server=y,suspend=n"

# Sample JPDA settings for shared memory debugging
#JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_shmem,server=y,suspend=n,address=jboss"

# Uncomment to not use JBoss Modules lockless mode
#JAVA_OPTS="$JAVA_OPTS -Djboss.modules.lockless=false"

# Uncomment to gather JBoss Modules metrics
#JAVA_OPTS="$JAVA_OPTS -Djboss.modules.metrics=true"

# Uncomment this in order to be able to run WildFly on FreeBSD
# when you get "epoll_create function not implemented" message in dmesg output
#JAVA_OPTS="$JAVA_OPTS -Djava.nio.channels.spi.SelectorProvider=sun.nio.ch.PollSelectorProvider"
```

- 2) Сохранить файл standalone.conf.

- 3) Повторить все действия данного раздела для каждого сервера приложений, входящего в состав кластера.

### Настройка соединения с SMTP сервером

Для отправки исходящей почты необходимо выполнить следующие настройки только на одном из серверов приложений:

- 1) В файле `/opt/wildfly/standalone/configuration/standalone.xml` найти блок с именем `mail-smtp`
- 2) и указать имя хоста и порт SMTP сервера

```
<outbound-socket-binding name="mail-smtp">
<remote-destination host="X.X.X.X" port="25"/>
</outbound-socket-binding>
```

Где "X.X.X.X" – ip адрес почтового сервера

Если для подключения к SMTP серверу требуется аутентификация, то необходимо задать следующие значения в блок `urn:jboss:domain:mail`

```
<subsystem xmlns="urn:jboss:domain:mail:2.0">
<mail-session name="default" jndi-name="java:jboss/mail/Default">
<smtp-server          outbound-socket-binding-ref="mail-smtp"          ssl="true"
username="<sendMailAddress> " password="<mailPassword>"/>
</mail-session>
</subsystem>
```

Где:

- `ssl="true"` если при подключении к SMTP серверу используется SSL;
- `username` – адрес почтового ящика, с которого будет отправляться почта;
- `password` – пароль от почтового ящика.

- 3) Сохранить изменения в файле `standalone.xml`

- 4) В файле `/opt/wildfly/standalone/configuration/system.properties` Указать следующие значения параметров:

```
enableSendingEmails=true
sendEmailFromAddress= username
```

- 5) Сохранить изменения в файле `system.properties` Перезапустить сервер приложений.

```
# systemctl restart wildfly
```

### 3.4.12 Настройка параметров в конфигурационных файлах узлов кластера сервера приложений

Параметр	Описание	Значение по умолчанию
dbConnectionURL	Адрес базы данных в формате JDBC <i>Пример:</i> Продуктивная среда: dbConnectionURL =jdbc:postgresql://X.X.X.X:5432/galaktika_db	В соответствии с проектом
dbSystemSchema	Имя системной схемы (обычно равно значению system). Для всех СУБД кроме postgresql имя схемы обычно совпадает с именем БД <i>Пример:</i> dbSystemSchema=system	system
dbUserName	Имя пользователя для подключения к СУБД	В соответствии с проектом
dbPassword	Пароль пользователя для подключения к СУБД	В соответствии с проектом
ipDataPath	Адрес основной папки с данными IP в данной папке хранятся следующие данные: – attachments – вложения – indexes/text – текстовые индексы – classlibrary – скомпилированные библиотеки классов – templates – шаблоны выгрузок	Для *nix: /usr/IPData/
javaHomeJDK	Расположение Java JDK (используется для компиляции классов ClassLibrary) <i>Пример:</i> javaHomeJDK=/usr/lib/jvm/java-17-openjdk	usr/lib/jvm/java-17-openjdk
javaHomeJRE	Расположение Java JRE (используется для выполнения кода на Java) <i>Пример:</i> javaHomeJRE=/usr/lib/jvm/java-17-openjdk	/usr/lib/jvm/java-17-openjdk
deploymentFolder	Имя папки или WAR файла приложения <i>Пример:</i> deploymentFolder=deployments/galaktika_SaaS-0.0.1-SNAPSHOT.war	deployments/ galaktika_SaaS-0.0.1-SNAPSHOT.war
attachmentMaxSize	Максимальный размер вложения в Мб attachmentMaxSize=25	10

Параметр	Описание	Значение по умолчанию
isTraceHttpRequestTimeOn	<p>Флаг трассировки и времени выполнения http запросов (результаты записываются в лог с маркером <a href="#">HTTP REQUEST</a>).</p> <p><i>Пример:</i> isTraceHttpRequestTimeOn=true</p>	false
isStartupTextReindexOn	<p>Флаг автоматической переиндексации тестовых индексов при запуске системы.</p> <p><i>Пример:</i> isStartupTextReindexOn=true</p>	false
baseURL	<p>Ссылка, по которой система доступна извне.</p> <p><i>Пример:</i> baseURL=http://main.page</p>	В соответствии с проектом
defaultAuthenticationType	<p>Способ аутентификации пользователя по умолчанию (если не задан в свойствах пользователя).</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> <li>– LDAP – для аутентификации через LDAP</li> <li>– APPLICATION – для аутентификации через встроенные в приложение средства</li> </ul> <p><i>Пример:</i> defaultAuthenticationType=LDAP</p>	APPLICATION
enableSendingEmails	<p>Флаг включения/выключения отправки почты</p> <p><i>Пример:</i> enableSendingEmails=true</p>	false
sendEmailFromAddress	<p>Имя ящика, от имени которого будет выполняться рассылка.</p> <p><i>Пример:</i> sendEmailFromAddress=cppsh@team-galaxy.ru (Адрес SMTP сервера задается в файле /opt/wildfly/standalone/configuration/standalone.xml в блоке</p> <pre>&lt;outbound-socket-binding name="mail-smtp"&gt; &lt;remote-destination host="cppsh@team-galaxy.ru" port="25"/&gt; &lt;/outbound-socket-binding&gt;</pre> <p>)</p>	В соответствии с проектом

Параметр	Описание	Значение по умолчанию
clusterNodes	Список адресов серверов приложений в формате ip:port разделенных через точку с запятой (включая сервер, для которого выполняется настройка) <i>Пример:</i> clusterNodes=CPPSH-APP-1; CPPSH -APP-2; CPPSH -APP-3	В соответствии с проектом
clusterKey	Ключ шифрования сообщений между узлами кластера. Должен быть одинаков для всех узлов кластера. <i>Пример:</i> clusterKey=thisTestIsSecretClusterKey1	thisTestIsSecretClusterKey1
ipDataPath	Ссылка на сетевую папку для хранения вложений. <i>Пример:</i> ipDataPath=/mnt/IPSharedData/	ipDataPath=/mnt/IPSharedData/
passwordComplexityRegex	Регулярное выражение для проверки сложности пароля. Если параметр не задан – проверка сложности пароля не выполняется. <i>Пример:</i> passwordComplexityRegex=^(?:(?=.*[a-z])(?=.*[A-Z])(?=.*[\d\W]) (?=.*\W)(?=.*\d) (?=.*\W)(?=.*[A-Z])(?=.*\d)).{8,}\$	Не задан
passwordComplexityMessage	Сообщение с требованиями к сложности пароля в кодировке UTF-8 (работает в паре с параметром passwordComplexityRegex). <i>Пример:</i> passwordComplexityMessage=\u0414\u043b\u0438\u043d\u0020\u043f\u0430\u0440\u043e\u043b\u044f\u0020\u0434\u043e\u043b\u0436\u0435\u0442 \u0431\u044b\u0442\u044c \u0434\u043b\u0438\u043d\u043d\u044b\u043c \u0438 \u0432\u0435\u0441\u0442\u0438 \u0432\u043e\u0434\u043d\u043e\u0439 \u0432\u0435\u0440\u0441\u0438\u0438. \u0414\u043e\u043b\u0436\u043d\u043e \u0432\u043e\u0434\u0435\u043b\u0438\u0442\u044c \u0432\u0435\u0441\u0442\u043e\u0439 \u0432\u043e\u0434\u043d\u043e\u0439 \u0432\u0435\u0440\u0441\u0438\u0438 \u0432\u043e\u0434\u043d\u043e\u0439 \u0432\u0435\u0440\u0441\u0438\u0438.	Не задан





## Цифровая производственная платформа для хозяйства автоматики и телемеханики

\*Пользователь



Введите ваш логин

\*Ваш пароль



Введите ваш пароль



[Восстановить пароль](#)

Войти в систему



В случае успешного входа вы перейдёте на первую страницу с данными программы (вид первой страницы определяется ролью пользователя).

В случае недоступности программы или ошибке входа на экране будет выведено соответствующее сообщение.

### 3.6 Завершение работы

Завершить работу программы, можно закрыв страницу браузера или перейти в меню в раздел Профиль > Выйти общесистемного меню.

Выполнение указанных действий повлечет за собой завершение работы пользователя.

**Контактная информация по вопросам установки и настройки программы для ЭВМ: Даниленко Дмитрий Игоревич, моб.: +7(921)792-89-25, режим работы: 9-00 – 18-00.**