



«Управление производственными активами путевого
комплекса железнодорожной инфраструктуры –
ЕАМ.Бетанкур» (ЕАМ.Бетанкур)

Установка и настройка

системы «Управление производственными активами путевого комплекса
железнодорожной инфраструктуры – ЕАМ.Бетанкур» (ЕАМ.Бетанкур)

Версия 1.0

2023

1.	Установка и настройка системы ЕАМ.Бетанкур.....	4
1.1.	Требования по аппаратной и программной конфигурации оборудования для системы ЕАМ.Бетанкур	4
1.1.1.	Аппаратная конфигурация.....	4
1.1.1.1.	Сервер БД.....	4
1.1.1.2.	Веб - сервер.....	4
1.1.1.3.	Интеграционный сервер.....	5
1.1.1.4.	Рабочие места (клиентская часть)	5
1.1.2.	Программная конфигурация.....	5
1.1.2.1.	Требования к серверу БД	5
1.1.2.2.	Требования к серверу приложений	5
1.1.2.3.	Требования к интеграционному серверу	5
1.1.2.4.	Требования к Рабочим местам (клиентская часть)	6
1.2.	Состав комплекса ЕАМ.Бетанкур.....	6
1.3.	Установка и настройка сервера БД	6
1.3.1.	Настройка дисков под установку СУБД PostgreSQL.....	6
1.3.2.	Настройка дисков под создание бекапов.....	6
1.3.3.	Инсталляция СУБД PostgreSQL.....	7
1.3.4.	Установка и настройка кластера БД в режиме master-slave.....	8
1.3.5.	Восстановление бекапа системы	9
1.3.6.	Настройка фаерволла	9
1.3.7.	Настройка бекапирования на серверах кластера PostgreSQL.....	10
1.3.7.1.	Требования к локальному хранению резервных копий.....	10
1.3.7.2.	Инструкция по созданию полной резервной копии БД	10
1.3.7.3.	Инструкция по созданию резервной копии отдельных объектов БД	11
1.3.7.4.	Инструкция по хранению транзакционных логов.....	11
1.3.7.5.	Инструкция по очистке устаревших транзакционных логов и бекапов.	12
1.4.	Установка и настройка сервера приложений	13
1.4.1.	Общие требования к серверам приложений	13

1.4.2.	Копирование дистрибутива.....	14
1.4.3.	Установка Java JDK и Java JRE.....	14
1.4.4.	Установка сервера приложений WildFly и инициализация сервиса wildfly ..	14
1.4.5.	Настройка директории для хранения статических файлов	15
1.4.6.	Настройка фаерволла на каждом узле кластера	15
1.4.7.	Настройка соединения с СУБД.....	15
1.4.8.	Запуск службы wildfly.....	15
1.4.9.	Настройка сетевого ресурса для хранения вложений и доступа к нему по протоколу NFS.....	15
1.4.10.	Настройка клиента NFS на каждом узле кластера.	16
1.4.11.	Установка и настройка на веб-сервере программного обеспечения - балансировщика нагрузки HAProxy.....	16
1.4.12.	Настройка параметров в конфигурационных файлах узлов кластера сервера приложений.	18
1.5.	Вход в систему	21
1.6.	Завершение работы.....	21

1. УСТАНОВКА И НАСТРОЙКА СИСТЕМЫ ЕАМ.БЕТАНКУР

1.1. ТРЕБОВАНИЯ ПО АППАРАТНОЙ И ПРОГРАММНОЙ КОНФИГУРАЦИИ ОБОРУДОВАНИЯ ДЛЯ СИСТЕМЫ ЕАМ.БЕТАНКУР

1.1.1. АППАРАТНАЯ КОНФИГУРАЦИЯ

1.1.1.1. 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 11, 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 (50Гб), 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.

1.1.1.2. 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 (50Гб), файлов настройки веб-сервера, журналов работы (логов) и другое (50 Гб).

1.1.1.3. 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.

1.1.1.4. РАБОЧИЕ МЕСТА (КЛИЕНТСКАЯ ЧАСТЬ)

В качестве рабочих мест необходимо использовать компьютеры со следующими характеристиками:

- Процессор: Не менее 2,8 ГГц;
- Монитор: разрешение от 1024x768;
- 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: 100Мбит/с.

1.1.2. ПРОГРАММНАЯ КОНФИГУРАЦИЯ

1.1.2.1. ТРЕБОВАНИЯ К СЕРВЕРУ БД

На сервере БД должны быть установлены:

- Операционная система CentOS Linux 7 с последними обновлениями;
- PostgreSQL 11.

1.1.2.2. ТРЕБОВАНИЯ К СЕРВЕРУ ПРИЛОЖЕНИЙ

На сервере приложений должна быть установлена операционная система CentOS Linux 7 с последними обновлениями.

1.1.2.3. ТРЕБОВАНИЯ К ИНТЕГРАЦИОННОМУ СЕРВЕРУ

На интеграционном сервере должна быть установлена операционная система CentOS Linux 7 с последними обновлениями.

1.1.2.4. ТРЕБОВАНИЯ К РАБОЧИМ МЕСТАМ (КЛИЕНТСКАЯ ЧАСТЬ)

Для корректной работы приложения необходимо, чтобы на рабочем месте пользователя был установлен Web-браузер Яндекс.Браузер версии 18 и выше.

1.2. СОСТАВ КОМПЛЕКСА ЕАМ.БЕТАНКУР

В состав комплекса ЕАМ.Бетанкур входит системное ПО, установку которого необходимо выполнять из централизованного распределенного репозитория для операционных систем CentOS и RHEL.

В состав поставляемого комплекса входят следующие компоненты:

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

1.3. 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;
- Инсталляция СУБД PostgreSQL;
- Установка и настройка кластера БД в режиме master-slave;
- Восстановление бекапа системы;
- Настройка фаерволла;
- Настройка бекапирования на серверах кластера PostgreSQL.

1.3.1. 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 с перспективой роста на 1 год).

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 предъявляются.

1.3.2. НАСТРОЙКА ДИСКОВ ПОД СОЗДАНИЕ БЕКАПОВ

Директории с БД и backup должны располагаться на отдельных смонтированной 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).

1.3.3. ИНСТАЛЛЯЦИЯ СУБД POSTGRESQL

Установите из официального репозитория версию СУБД PostgreSQL.

Инициализируйте базу данных galaktika_db (необходимо запускать команды от пользователя, под которым будет работать сервер БД).

- 1) 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.

1.3.4. 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 pgsq1 pgsq1 \
pgctl="/usr/pgsq1-11/bin/pg_ctl"\
psql="/usr/pgsq1-11/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 pgsq1 \
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.

1.3.5. ВОССТАНОВЛЕНИЕ БЕКАПА СИСТЕМЫ

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

1.3.6. НАСТРОЙКА ФАЕРВОЛЛА

Выполнить настройку фаерволла:

```
# iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

```
# iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

```
# iptables -A INPUT -p udp -m multiport --dports 10053,111,2049,32769,875,892 -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
# iptables -A INPUT -p tcp -m multiport --dports 10053,111,2049,32803,875,892 -m state --state NEW,ESTABLISHED -j ACCEPT
```

```
# iptables -A OUTPUT -p udp -m multiport --sports 10053,111,2049,32769,875,892 -m state --state ESTABLISHED -j ACCEPT
```

```
# iptables -A OUTPUT -p tcp -m multiport --sports 10053,111,2049,32803,875,892 -m state --state ESTABLISHED -j ACCEPT
```

```
# service iptables save
```

1.3.7. НАСТРОЙКА БЕКАПИРОВАНИЯ НА СЕРВЕРАХ КЛАСТЕРА POSTGRESQL

1.3.7.1. 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
	Бэкап виртуальной машины	1 раз в сутки	2
Серверы 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
	Бэкап виртуальной машины	1 раз в сутки	2

1.3.7.2. 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 супер пользователя.

Доп. параметры:

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

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

1.3.7.3. 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=sctm -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
```

1.3.7.4. 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) и добавить 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 обозначет 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.

1.3.7.5. 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-11/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-11/bin/pg_resetxlog -o 4716704 -x 1186399159 -f /pgdata/data/
```

1.4. УСТАНОВКА И НАСТРОЙКА СЕРВЕРА ПРИЛОЖЕНИЙ

Установку EAM.Бетанкур необходимо выполнять из централизованного распределенного репозитория для операционных систем CentOS и RHEL.

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.

1.4.1. ОБЩИЕ ТРЕБОВАНИЯ К СЕРВЕРАМ ПРИЛОЖЕНИЙ

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

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

На сетевом файловом хранилище должны быть установлены следующие пакеты:

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

1.4.2. КОПИРОВАНИЕ ДИСТРИБУТИВА

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

- jdk-8u181-linux-x64.rpm
- jre-8u181-linux-x64.rpm
- wildfly-8.2.1.Final.tar.gz

1.4.3. УСТАНОВКА JAVA JDK И JAVA JRE

Поочередно выполнить следующий набор команд:

```
# yum -y localinstall jre-8u181-linux-x64.rpm
# yum -y localinstall jdk-8u181-linux-x64.rpm
```

1.4.4. УСТАНОВКА СЕРВЕРА ПРИЛОЖЕНИЙ WILDFLY И ИНИЦИАЛИЗАЦИЯ СЕРВИСА WILDFLY

Распаковать архив в папку /opt/

```
# tar xvzf wildfly-8.2.1.Final.tar.gz -C /opt
```

Поочередно выполнить следующий набор команд:

```
# ln -s /opt/wildfly-8.2.1.Final /opt/wildfly
# cp /opt/wildfly/bin/init.d/wildfly-init-redhat.sh /etc/init.d/wildfly
# cp /opt/wildfly/bin/init.d/wildfly.conf /etc/default
# sudo chmod +x /etc/init.d/wildfly
# sudo chmod +x /opt/wildfly/bin/standalone.sh
# chkconfig --add wildfly
# chkconfig wildfly on
```

1.4.5. 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.

1.4.6. НАСТРОЙКА ФАЕРВОЛЛА НА КАЖДОМ УЗЛЕ КЛАСТЕРА

```
# iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT
# iptables -A OUTPUT -p tcp --sport 80 -m conntrack --ctstate ESTABLISHED -j ACCEPT
# iptables -A INPUT -p tcp --dport 8080 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT
# iptables -A OUTPUT -p tcp --sport 8080 -m conntrack --ctstate ESTABLISHED -j ACCEPT
# service iptables save
```

1.4.7. 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/galaktika_db

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

X.X.X.X — ip адрес сервера СУБД

main.page — страница публикации системы

Сохранить файл `system.properties`.

1.4.8. ЗАПУСК СЛУЖБЫ WILDFLY

Запустить службу `wildfly`:

```
#systemctl start wildfly
```

1.4.9. 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:

```
# iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT
# iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
# iptables -A INPUT -p udp -m multiport --dports 10053,111,2049,32769,875,892 -m
state --state NEW,ESTABLISHED -j ACCEPT
# iptables -A INPUT -p tcp -m multiport --dports 10053,111,2049,32803,875,892 -m state
--state NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p udp -m multiport --sports 10053,111,2049,32769,875,892 -m
state --state ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp -m multiport --sports 10053,111,2049,32803,875,892 -m
state --state ESTABLISHED -j ACCEPT
# service iptables save
```

1.4.10. НАСТРОЙКА КЛИЕНТА NFS НА КАЖДОМ УЗЛЕ КЛАСТЕРА.

Для настройка 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
```

1.4.11. УСТАНОВКА И НАСТРОЙКА НА ВЕБ-СЕРВЕРЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ - БАЛАНСИРОВЩИКА НАГРУЗКИ HAProxy.

На сервере балансировки нагрузки дополнительно должны быть установлены следующие пакеты:

```
# yum -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

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$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBASS_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"
```

сохранить файл standalone.conf.

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

В файле /opt/wildfly/standalone/configuration/standalone.xml найти блок с **именем mail-smtp**

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.

Сохранить изменения в файле standalone.xml

В файле /opt/wildfly/standalone/configuration/system.properties

Указать следующие значения параметров:

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

Сохранить изменения в файле system.properties

Перезапустить сервер приложений.

```
# systemctl restart wildfly
```

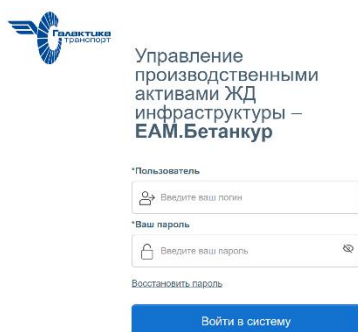
1.4.12. НАСТРОЙКА ПАРАМЕТРОВ В КОНФИГУРАЦИОННЫХ ФАЙЛАХ УЗЛОВ КЛАСТЕРА СЕРВЕРА ПРИЛОЖЕНИЙ.

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: <ul style="list-style-type: none"> • 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/java/jdk1.8.0_181	/usr/java/jdk1.8.0_181
javaHomeJRE	Расположение Java JRE (используется для выполнения кода на Java) <i>Пример:</i> javaHomeJRE=/usr/java/jre1.8.0_181	/usr/java/jre1.8.0_181
deploymentFolder	Имя папки или WAR файла приложения <i>Пример:</i>	deployments/

Параметр	Описание	Значение по умолчанию
	deploymentFolder=deployments/galaktika_SAAS-0.0.1-SNAPSHOT.war	galaktika_SAAS-0.0.1-SNAPSHOT.war
attachmentMaxSize	Максимальный размер вложения в Мб attachmentMaxSize=25	10
isTraceHttpRequestTimeOn	Флаг трассировки и времени выполнения http запросов (результаты записываются в лог с маркером HTTP REQUEST). Пример: isTraceHttpRequestTimeOn=true	false
isStartupTextReindexOn	Флаг автоматической переиндексации тестовых индексов при запуске системы. Пример: isStartupTextReindexOn=true	false
baseUrl	Ссылка, по которой система доступна извне. Пример: baseUrl=http://main.page	В соответствии с проектом
defaultAuthenticationType	Способ аутентификации пользователя по умолчанию (если не задан в свойствах пользователя). Возможные значения: <ul style="list-style-type: none"> LDAP – для аутентификации через LDAP APPLICATION – для аутентификации через встроенные в приложение средства Пример: defaultAuthenticationType=LDAP	APPLICATION
enableSendingEmails	Флаг включения/выключения отправки почты Пример: enableSendingEmails=true	false
sendEmailFromAddress	Имя ящика, от имени которого будет выполняться рассылка. Пример: sendEmailFromAddress=eam@betancourteam.ru (Адрес SMTP сервера задается в файле /opt/wildfly/standalone/configuration/standalone.xml в блоке <outbound-socket-binding name="mail-smtp"> <remote-destination host="eam.betancourteam.ru" port="25"/> </outbound-socket-binding>)	В соответствии с проектом
clusterNodes	Список адресов серверов приложений в формате ip:port разделенных через точку с запятой (включая сервер, для которого выполняется настройка) Пример: clusterNodes=EAM-APP-1; EAM-APP-2; EAM-APP-3	В соответствии с проектом
clusterKey	Ключ шифрования сообщений между узлами кластера. Должен быть одинаков для всех узлов кластера. Пример: clusterKey=thisTestIsSecretClusterKey1	thisTestIsSecretClusterKey1
ipDataPath	Ссылка на сетевую папку для хранения вложений. Пример:	ipDataPath=/mnt/IPS

По умолчанию ссылка на систему имеет следующий вид: <http://31.129.101.73:8080/#/login>

На открывшейся странице входа введите логин и пароль:



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

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

1.6. ЗАВЕРШЕНИЕ РАБОТЫ

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

Использование указанных средств повлечет за собой завершение работы системы.