Руководство пользователя: Автоинсталятор кластера HA

Центр разработки PostgreSQL

Exported on 05/29/2020

Table of Contents

1 Руководство для сотрудника сопровождения

2 Общая информация

Автоматическая установка PostgreSQL Sber Edition возможна с использованием Jenkins или ansible.

Для установки с помощью Jenkins необходимо заполнить входные параметры job'a, список и описание которых представлены в соответствующем разделе.

В случае установки с помощью ansible, параметры необходимо корректировать в inventory для каждого конкретного случая.

Рекомендуется использовать установку с помощью Jenkins.

Установка PostgreSQL SE возможна на виртуальные машины с ОС RedHat не ниже 7.7

ВАЖНО! Может возникнуть ошибка при установке linux пакетов. Перед запуском Jenkins или ansible необходимо удостовериться, что на КТС-ах, на которые планируется развертывание, в /etc/yum.repos.d/mirror.repo включен следующий репозиторий:

```
[EPEL7]
name=EPEL7
baseurl=http://mirror.ca.sbrf.ru/rhel-extras/EPEL7
gpgcheck=0
enabled=1
```

```
[EPEL7]
name=EPEL7
baseurl=http://mirror.sigma.sbrf.ru/rhel-extras/EPEL7
gpgcheck=0
enabled=1
```

3 Установка PostgreSQL Sber Edition с помощью Jenkins

3.1 Ссылки на Jenkins job's сегмента sigma и alpha:

- *Jenkins CI SIGMA* https://sbt-jenkins.sigma.sbrf.ru/job/PGSQL/job/Install_PostgreSQL_SE/job/install_postgreSQL_SE/
- *Jenkins CI ALPHA* https://sbt-jenkins.ca.sbrf.ru/jenkins/job/PostgreSQL/job/Install_PostgreSQL_SE/
- *Jenkins CDL SIGMA* https://sbt-qa-jenkins.sigma.sbrf.ru/jenkins/job/PostgreSQL_SE/job/Install_PostgreSQL_SE/
- *Jenkins CDL ALPHA* https://sbt-qa-jenkins.ca.sbrf.ru/jenkins/job/PostgreSQL_SE/job/Install_PostgreSQL_SE/

3.2 *Для запуска установки PostgreSQL SE с помощью Jenkins требуется передать следующие параметры:*

<table>
<tr><th>Параметр</th><th>Описание</th><th>Пример заполненного параметра</th></tr>
<tr><td>installation_type</td><td>Тип установки. В данном поле выбирается тип установки из выпадающего списка:
standalone-postgresql-only - установка только postgreSQL SE на один узел
standalone-patroni-etcd-pgbouncer - установка postgreSQL SE с сервисами patroni, pgbouncer и etcd на один узел
cluster-patroni-etcd-pgbouncer - установка postgreSQL SE с сервисами patroni, pgbouncer и etcd
cluster-patroni-etcd-pgbouncer-haproxy - установка postgreSQL SE с сервисами patroni, pgbouncer, etcd и haproxy
cluster-patroni-etcd-pgbouncer-load_balancer - установка postgreSQL SE с сервисами patroni, pgbouncer, etcd и сервисом балансировки нагрузки</td><td><table><tr><th>Пример заполненного параметра для standalone</th><th>Пример заполненного параметра для cluster</th></tr><tr><td>standalone-postgresql-only
standalone-patroni-etcd-pgbouncer</td><td>cluster-patroni-etcd-pgbouncer
cluster-patroni-etcd-pgbouncer-haproxy</td></tr></table></td></tr>
</table>

<table>
<tr><th>Параметр</th><th>Описание</th><th>Пример заполненного параметра</th></tr>
<tr><td>hosts_list</td><td>Список серверов, на которые будет выполняться установка. В архитектуре standalone передается один сервер, в cluster - три сервера. Сервера в Jenkins параметре указываются через пробел. Можно указывать доменные имена или ip адреса. В списке с серверами первый сервер всегда будет выступать в роли master, второй в роли standby, третий сервер - арбитр.</td><td><table><tr><th>Пример заполненного параметра для standalone</th><th>Пример заполненного параметра для cluster</th></tr><tr><td>10.53.84.26</td><td>10.53.84.26 10.53.84.27 10.53.84.28</td></tr></table></td></tr>
<tr><td>ssh_user</td><td>Имя пользователя в ОС Linux, обладающий правами sudo all. Данный пользователь должен быть создан на всех серверах, которые будут переданы в Jenkins параметре hosts_list</td><td>pprb_dev</td></tr>
<tr><td>ssh_password</td><td>Пароль пользователя ОС Linux, обладающего правами sudo all. Пароль должен быть одинаковым на всех серверах, которые будут переданы в Jenkins параметре hosts_list</td><td>123456</td></tr>
<tr><td>version</td><td>Версия PostgreSQL SE, которая будет загружена с Nexus и установлена на ранее указанные узлы. Данный Jenkins параметр содержит в себе по умолчанию самую свежую сборку PostgreSQL SE.
С информацией по каждой доступной версии можно ознакомиться по ссылке(see page 3)</td><td>D-03.003.00-3.3.0.2</td></tr>
<tr><td>database_name</td><td>Имя базы данных, которая будет создана в процессе установки</td><td>test_database</td></tr>
<tr><td>port</td><td>Порт базы данных</td><td>5432</td></tr>
<tr><td>tablespace_name</td><td>Имя табличного пространства, которое будет создано в процессе установки</td><td>test_tablespace</td></tr>
</table>

<table>
<tr><th>Параметр</th><th>Описание</th><th>Пример заполненного параметра</th></tr>
<tr><td>tablespace_location</td><td>Расположение создаваемого tablespace на диске в файловой системе</td><td>/pgsql</td></tr>
<tr><td>pgdata</td><td>Путь до каталога в файловой системе, где будут расположены файлы с данными базы данных</td><td>/pgdata</td></tr>
<tr><td>pglogs</td><td>Путь до каталога в файловой системе, где будут расположены файлы логирования базы данных</td><td>/home/postgresql/logs</td></tr>
<tr><td>clustername</td><td>Имя кластера базы данных, которое в дальнейшем будет использоваться в patroni и etcd. Если планируется установка БД в standalone архитектуре, то данный Jenkins параметр можно оставить со стандартным значением</td><td>test_cluster</td></tr>
<tr><td>customer</td><td>Заказчик БД, владелец БД по умолчанию</td><td><table><tr><th>Пример заполненного параметра для sigma</th><th>Пример заполненного параметра для alpha</th></tr><tr><td>18223423</td><td>Sidorov-EE</td></tr></table></td></tr>
<tr><td>as_admins</td><td>Список администраторов АС. В данном параметре передаются логины пользователей из active directory, которые в дальнейшем смогут входить в созданную базу данных с помощью личной учетной записи(LDAP).

В случае, когда необходимо передать несколько значений, то их необходимо указывать через запятую, без пробелов.</td><td><table><tr><th>Пример заполненного параметра для sigma</th><th>Пример заполненного параметра для alpha</th></tr><tr><td>18249011,18249012</td><td>Ivanov-IA,Petrov-VV</td></tr></table></td></tr>
</table>

Параметр	Описание	Пример заполненного параметра
as_tuz	Список логинов технических учетных записей в БД. Данные записи будут созданы в базе данных. Если имя ТУЗ не принципиально, то данный Jenkins параметр можно оставить со стандартным значением. В случае, когда необходимо передать несколько значений, то их необходимо указывать через запятую, без пробелов.	test_one, test_two
environment	Тип стенда, к которому относится/относятся виртуальные машины, указанные в Jenkins параметре **hosts_list**	DEV
security_level	Уровень конфиденциальности хранимых данных. Для уровня К1 требуется шифрование данных	К3
critical_level	Уровень критичности системы	Office Productivity

4 Контакты

С возникающими в процессе использования PostgreSQL Sber Edition Installer вопросами просьба обращаться к его разработчикам:

- Карпенко Андрей Сергеевич Karpenko.An.Ser@sberbank.ru[1]
- Краскин Павел Михайлович PMKraskin@sberbank.ru[2]

1 mailto:Karpenko.An.Ser@sberbank.ru
2 mailto:PMKraskin@sberbank.ru

5 Установка PostgreSQL Sber Edition с помощью Ansible

5.1 Для установки PostgreSQL SE с помощью ansible, необходимо:

1) Загрузить дистрибутив с PostgreSQL Sber Edition из nexus. Перед попыткой загрузки дистрибутива необходимо получить права на чтение с помощью портала "Друг".
Ссылки на Nexus:

- Nexus Sigma[3]
- Nexus Alpha[4]

2) Распаковать дистрибутив на Linux сервере с установленным Ansible не ниже 2.9.2, установить linux пакет *sshpass* а так же установить следующие python модули:

- jmespath==0.9.4
- netaddr==0.7.19
- PyYAML==5.3

3) Перейти в каталог с распакованным дистрибутивом. Далее перейти в каталог installer.

4) Перед запуском установки необходимо заполнить hosts.ini файл в соответствии с шаблоном и требуемым типом установки(installer/inventories/cluster/hosts.ini или installer/inventories/standalone/hosts.ini), добавив информацию о хостах и учетных данных пользователя, которые будет использовать ansible. Переменная ansible_password должна содержать пароль пользователя в чистом виде или же имя переменной, которая будет содержать зашифрованный с помощью ansible-vault пароль. Ниже представлены шаблоны файла hosts.ini для standalone и cluster архитектур.

hosts.ini для standalone

```
[standalone:children]
postgres_group

[postgres_group:children]
postgres_nodes

[postgres_group:vars]
ansible_connection=ssh

[postgres_nodes]
master                ansible_host=hostname/ip address               ansible_user=логин пользователя
ansible_password=пароль пользователя
```

3 https://sbtatlas.sigma.sbrf.ru/nexus/content/repositories/SBT_CI_distr_repo/as_postgresql/CI02289206_PostgreSQL_Sber_Edition/
4 http://sbtnexus.ca.sbrf.ru:8081/nexus/content/repositories/SBT_CI_distr_repo/as_postgresql/CI02289206_PostgreSQL_Sber_Edition/

hosts.ini для cluster

```
[cluster:children]
postgres_group
etcd_group

[postgres_group:children]
postgres_nodes

[etcd_group:children]
etcd_nodes

[postgres_group:vars]
ansible_connection=ssh

[etcd_group:vars]
ansible_connection=ssh

[postgres_nodes]
master                ansible_host=hostname/ip address               ansible_user=логин пользователя
ansible_password=пароль пользователя
replica         ansible_host=hostname/ip address                ansible_user=логин пользователя
ansible_password=пароль пользователя
[etcd_nodes]
etcd                ansible_host=hostname/ip address                ansible_user=логин пользователя
ansible_password=пароль пользователя
```

Важно! *Для корректной установки ansible_user должен иметь права для эскалации до root, т.е. должен иметь возможность выполнять все команды от имени root*

5) В случае, если в файле hosts.ini переменная ansible_password будет содержать ansible-vault, то пароль необходимо зашифровать следующей командой на хосте с установленным Ansible:

```
ansible-vault encrypt_string ${шифруемый пароль}
```

В случае, если пароль для хостов групп postgres_group и etcd_group одинаков, то достаточно полученный вывод команды ansible-vault поместить в inventory файл cluster.yml/standalone.yml, расположенный в каталоге inventories.

Аналогичным образом необходимо перешифровать пароли в файле **all.yml** (частично приведен ниже).

Важно! *Пароли должны быть зашифрованы используя один и тот же секрет и в случае работы с зашифрованным паролем, при запуске ansible необходимо добавить ключ **--ask-vault-pass***

6) После ранее выполненных шагов необходимо скорректировать(при необходимости) переменные в файле all.yml, расположенном в каталоге group_vars. Ниже представлены параметры, которые, при необходимости, нужно изменить

all.yml

```
PGDATA: "{{ '' | default('/pgdata/11/data',
true) }}"
#Директория для файлов данных PostgreSQL SE
PGLOGS: /home/postgres/logs
#Директория для логирующих файлов
PGUSERHOME: /home/postgres
#Домашняя директория пользователя, под которым будет запущена БД PostgreSQL SE
PGARCLOGS: /pgarclogs/
11
#Директория для архивных логирующих файлов
version:
"3.2.0.2"
#Версия rpm файла, который расположен в уже дистрибутиве PostgreSQL SE
#default_db
tablespace_name: "{{ '' | default('first_tablespace',
true) }}"                                                                    #Имя
табличного пространства, которое будет создано в процессе установки
tablespace_location: "{{ '' | default('/pgsql',
true) }}"
#Расположение табличного пространства
db_name: "{{ '' | default('first_db',
true) }}"
#Имя БД
#Roles for LDAP authentification
customer: "{{ '' | default('17644671', true) }}" #strictly 1 login
#Логин пользователя, которым выполняется установка PostgreSQL SE. Пользователь

#включается в групповую роль db_admin и может становиться суперпользователем
support: "{{ '' | default([13289436, 13289437], true) }}" #list of logins
#Список логинов группы сопровождения, так включаются в групповую роль db_admin
sec_officer: "{{ '' | default('17644671', true) }}" #strictly 1 login
#Представитель безопасности, включается в групповую роль sec_admin
as_admins: "{{ '' | default([17644671, 17631822], true) }}" #list of logins
#Список логинов администраторов АС, не имеют никаких привилегий.
as_TUZ: "{{ '' | default(['cdm', 'cdm_devops'], true) }}" #list of logins
#Локальные пользователи для работы приложений

postgres_password: !vault |
#Зашифрованный пароль для пользователя postgres, который будет создан в БД
                  $ANSIBLE_VAULT;1.1;AES256
                  643162346364653166356235306531623636643538366134343037303538346463646531306431 31
                  356336616163363164653936316231626166613032396432 0a373738373765386363316635633933
                  303537393865376132643966666662633764386561623731646662653130306432333366316262 36
                  306536383865623435 0a343230373239613565633537653239353631363238323030303637636432
                  6262
```

5) Находясь в каталоге *installer* инициировать запуск ansible-playbook в зависимости от типа установки

Одиночная установка postgeSQL без дополнительного программного обеспечения:

standalone-postgresql-only

```
ansible-playbook playbook.yaml -i inventories/standalone/hosts.ini --extra-vars "local_distr_path=${путь до дистрибутива} installation_type=standalone" -t always,standalone-postgresql-only
```

Одиночная установка postgeSQL с patroni, etcd и pgbouncer:

standalone-patroni-etcd-pgbouncer

```
ansible-playbook playbook.yaml -i inventories/standalone/hosts.ini --extra-vars "local_distr_path=${путь до дистрибутива} installation_type=standalone" -t always,standalone-patroni-etcd-pgbouncer
```

Кластерная установка postgeSQL с patroni, etcd и pgbouncer:

cluster-patroni-etcd-pgbouncer

```
ansible-playbook playbook.yaml -i inventories/cluster/hosts.ini --extra-vars "local_distr_path=${путь до дистрибутива} installation_type=cluster" -t always,cluster-patroni-etcd-pgbouncer
```

Кластерная установка postgeSQL с patroni, etcd, pgbouncer и haproxy:

cluster-patroni-etcd-pgbouncer-haproxy

```
ansible-playbook playbook.yaml -i inventories/cluster/hosts.ini --extra-vars "local_distr_path=${путь до дистрибутива} installation_type=cluster" -t always,cluster-patroni-etcd-pgbouncer-haproxy
```

Используемые в представленных выше командах переменные:

- **${путь до дистрибутива}** - абсолютный путь до загруженного и распакованного дистрибутива PostgreSQL SE

Значения используемых в команде запуска ansible ключей:

- -i - путь до inventory файла
- --extra-vars - переменные, которые по приоритету важнее переменных из inventory
- -t - теги для запуска
- -v - уровень логирования ansible. Может быть как пустым, так и -vvvvvv, где запуск без v - минимальное логирование

6 Руководство для администратора безопасности

Ссылка в JIRA

Ключ	Тема	Описание	Sprint	Team	Кэ
PGSQL-109[5]	Создание автоинсталятора для разворота кластера PostgreSQL[6]				ППРБ. PostgreSQL Sber Edition(2289206)

1 проблема[7]

6.1 Описание требований либо ссылки на страницы с ними:

Бизнес-требование: создание и включение в дистрибутив автоинсталятора кластера HA(see page 3)

6.1.1 1. UI/UX

В рамках работы реализуются или используются следующие виды интерфейсов пользователей:

1. Пакет RPM PostgreSQL Sber Edition
2. Скрипты автоматического развертывания и настройки кластера, в основе которого лежит дистрибутив PostgreSQL Sber Edition и RPM
3. UI Continious Integration Jenkins
4. Интеграционный слой для интеграции с системой Динамической конфигурации (далее ДИ)

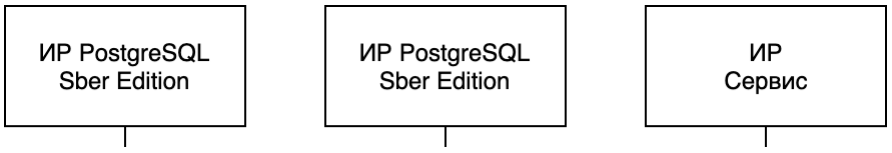
6.1.2 Регистрация продукта в АС SM

Созданный кластер регистрируется службами портала ДИ в АС SM.

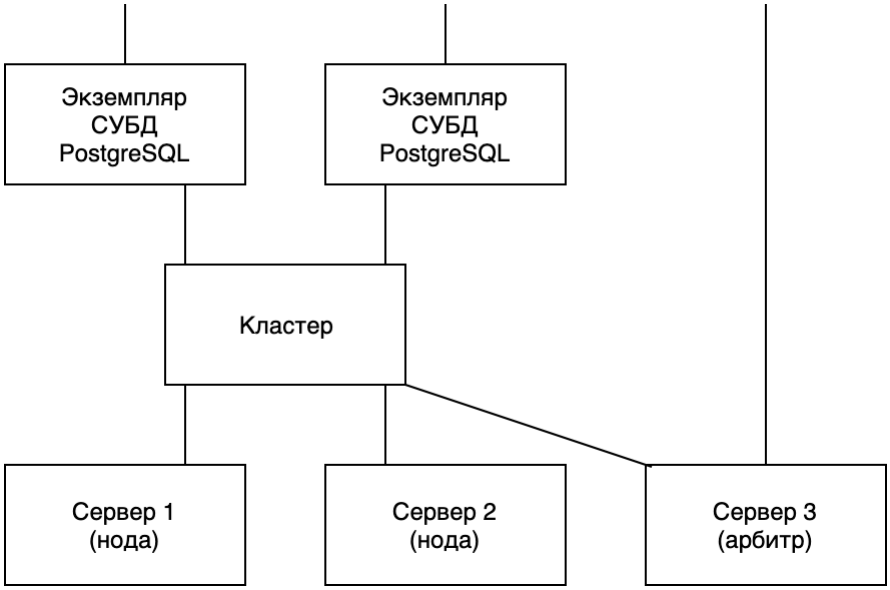
При этом создаются сущности нескольких типов.

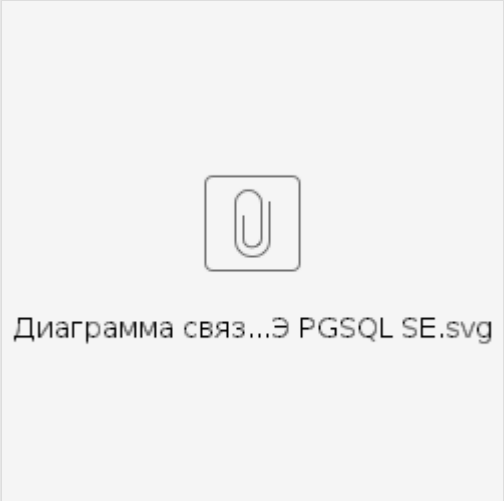
1. Категория - "Сервер", Тип "Виртуальный" или "Физический" - в зависимости от платформы развёртывания. Создаются по одному для каждого создаваемого сервера.
2. Категория - "Кластер".
3. Категория - "Экземпляр СУБД", Тип - "Экземпляр СУБД PostgreSQL". Создаётся два КЭ, по одному для каждого сервера БД.
4. Категория - "Информационные ресурсы", Тип - "PostgreSQL Sber Edition". Создаётся два КЭ, по одному для каждого сервера БД.
5. Категория - "Информационные ресурсы", Тип - "Сервис".

Выполняется связывание КЭ по следующей схеме:

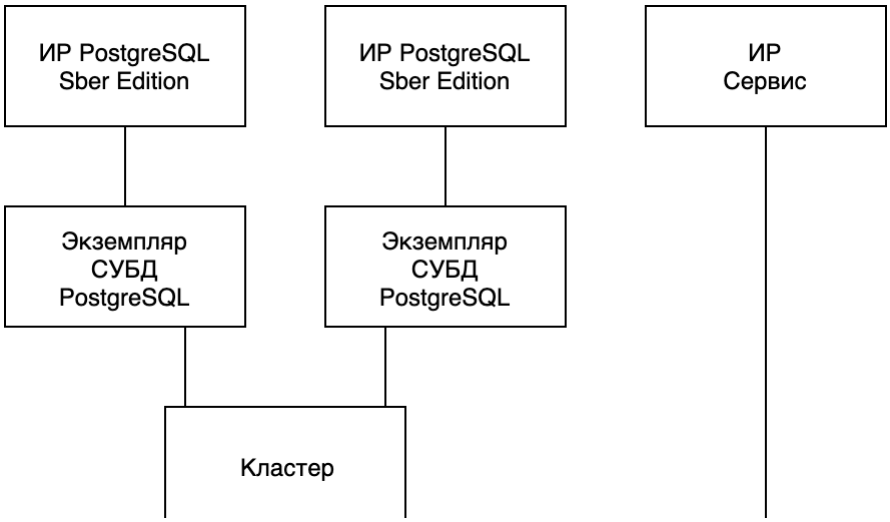


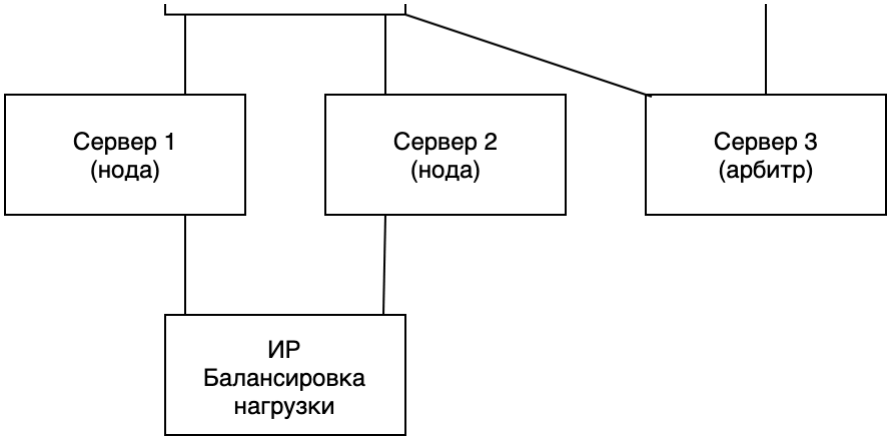
5 https://sbtatlas.sigma.sbrf.ru/jira/browse/PGSQL-109?src=confmacro
6 https://sbtatlas.sigma.sbrf.ru/jira/browse/PGSQL-109?src=confmacro
7 https://sbtatlas.sigma.sbrf.ru/jira/secure/IssueNavigator.jspa?reset=true&jqlQuery=key+in+%28PGSQL-109%29++++++++++&src=confmacro





Связывание КЭ при заказе БН (Балансировщик Нагрузки) выполняется по следующей схеме:





ИР Балансировка нагрузки согласно Заказ балансировщика нагрузки через ЗНИ(see page 3)

Поля КЭ Категории "Информационные ресурсы", Тип - "PostgreSQL Sber Edition" заполняются следующим образом : **Таблица 1**

№	Наименование поля в КЭ	Системное наименование в HPSM	Значения	Описание поля
1	Категория	type	infresource	Соответствует "Информационные ресурсы"
2	Тип	subtype	PostgreSQL Sber Edition	
3	Статус	hpc.status	Эксплуатируется	
4	Класс среды	environment	Тестовый Промышленный	Наследуется от родительского Стенда
5	Наименование	tps.name[8]	Передаётся значение поля "Имя базы данных" с Портала	
6	Служебное наименование	tps.comments1	Передается значение "Название услуги" с Портала	
7	Версия СУБД/СП	tps.subd.version	Передаётся значение "Версия PosgreSQL Sber Edition" с Портала	

8 http://tps.name

№	Наименование поля в КЭ	Системное наименование в HPSM	Значения	Описание поля
8	Категория информации	tps.information.category	И1 И2 И3	Наследуется от родительского Стенда
9	Организация-Заказчик	tps.owner.name[9]	Указывается заказчик продукта	Наследуется от родительского Стенда
10	Администраторы	tps.support.groups	Администраторы АС	Наследуется от родительского Стенда
11	Группа Сопровождение	assignment	~~Наследуется от поля "Группа Сопровождение" у КЭ ИТ-услуга PostgreSQL Sber Edition~~ SberInfra УБД Администрирование СУБД MS SQL (Лесных А.П.)	
12	Группа-Владелец	assignment	~~Команда развития и разработки PostgreSQL~~ SberInfra УБД Администрирование СУБД MS SQL (Лесных А.П.)	~~Команда является разработчиком и стэйкхолдером PostgreSQL Sber Edition~~
13	Группа Администраторов	sb.administrator.group	Заполняется аналогично stand-alone конфигурации	
14	Администратор Безопасности	assignment	Администратор Безопасности	Новое поле, создан issue на изменение ИР. Добавлено будет после появления в SberInfra команды Администраторов Безопасности.

9 http://tps.owner.name

6.1.3 2. Модель данных

Сборка RPM осуществляется по инструкции Сборка rpm с дистрибутивом PostgreSQL Sber Edition(see page 3). RPM включается в дистрибутив PostgreSQL Sber Edition

При автоматическом разворачивании кластера, согласно Стандарт конфигурирования настроек безопасности PostgreSQL Sber Edition(see page 0), создаются следующие групповые роли (пользователи СУБД):

- db_admin (SUPERUSER)
- sec_admin
- backup_admin
- tuz_monitoring
- as_admin

При заказе стенда через портал ДИ передаются ldap логины администраторов и пользователей стенда на вход автоинсталятору для настройки кластера СУБД (привязка их к групповым ролям):

1. Организация-Заказчик
2. Эксплуатация (в зависимости от того, где работает продукт: в Платформе или нет) - не надо передавать (хардкод эксплуатация SberInfra)
3. Офицер безопасности (Новое поле в ИР PostgreSQL Sber Edition - **Таблица 1**) - Временно не передавать, пока не появится ответственное подразделение
4. Администраторы АС
5. Администраторы стендов ИФТ и Администраторы стендов НТ в ролевой модели не участвуют.

В зависимости от среды кластер настраивается следующим образом: **Таблица 2**

Настройка \Среда	**DEV**	**IFT**	**НТ**	**ПСИ/ПРОМ**
Ролевая модель	Заказчик: 1. db_admin 2. sec_admin 3. backup_admin 4. tuz_monitoring 5. as_admin Эксплуатация: 1. db_admin 2. backup_admin 3. tuz_monitoring Офицер безопасности: 1. sec_admin Администраторы АС: 1. as_admin ТУЗы АС: 1. as_admin	Организация-Заказчик: 1. as_admin Эксплуатация: 1. db_admin 2. backup_admin 3. tuz_monitoring Офицер безопасности: 1. sec_admin Администраторы АС: 1. as_admin ТУЗы АС: 1. as_admin	Организация-Заказчик: 1. as_admin Эксплуатация: 1. db_admin 2. backup_admin 3. tuz_monitoring Офицер безопасности: 1. sec_admin Администраторы АС: 1. as_admin ТУЗы АС: 1. as_admin	Организация-Заказчик: 1. as_admin Эксплуатация: 1. db_admin 2. backup_admin 3. tuz_monitoring Офицер безопасности: 1. sec_admin Администраторы АС: 1. as_admin ТУЗы АС: 1. as_admin

Настройка \Среда	**DEV**				**IFT**				**НТ**				**ПСИ/ПРОМ**			
Уровень конфиденциальности хранимых данных	К1 или ДПК	К2	К3	К4	К1 или ДПК	К2	К3	К4	К1 или ДПК	К2	К3	К4	К1 или ДПК	К2	К3	К4
pgaudit	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
password-policy	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
LDAP	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Мониторинг													+	+	+	+
Резервирование													+	+	+	+
TDE									+				+			
SSL					+	+	+	+	+	+	+	+	+	+	+	+
Защита от администраторов					+	+			+	+			+	+		

6.1.4 Список параметров, ожидаемых от ДИ (интеграционный слой)

Имя параметра(например app_version)
label: выводимая пользователю портала информация об инпуте
type: string, integer, enum
constrain: тут может быть regexp или указать возможные значения ['значение1','значение2'] или [1,2,3,4,]
Доп.аттрибуты - скрытый, маскировка ввода звездочками

Таблица 3

Имя параметра	label	type	constrain	Доп. аттрибуты	Комментарий
postgresql_first_node	- (не выводится на UI портала)				
login/password first node	- (не выводится на UI портала)				
postgresql_second_node	- (не выводится на UI портала)				
login/password second node	- (не выводится на UI портала)				
etcd_node	- (не выводится на UI портала)				
login/password etcd node	- (не выводится на UI портала)				

Имя параметра	label	type	constrain	Доп. аттрибуты	Комментарий
Environment	Среда развертывания	string Выпадающий список	DEV, ST1, ST2, MINOR-CHECK, ИФТ, НТ, EDU, ПСИ, HF, MINOR-GO, ПРОМ		В зависимости от среды предполагается различная конфигурация кластера (настройка ролевой модели, включение шифрования)
clustername	Имя кластера базы данных	string	Ограничения: 1. Без пробелов и табуляций 2. Без перевода строк 3. Латиница, цифры, _, - 4. Без слэшей		Сквозная переменная, применяется при конфигурировании HashiCorp Vault, Patroni
security_level	Уровень конфиденциальности хранимых данных	string Выпадающий список	K1, K2, K3, K4		В зависимости от уровня конфиденциальности будет включено или выключено прозрачное шифрование
Critical_level	Уровень критичности системы	string Выпадающий список	Mission Critical, Business Critical, Business Operational, Office Productivity		В зависимости от уровня критичности системы будет по-разному настроено резервирование, мониторинг и т.д. кластера
Postgre SQL SE version	Postgre SQL SE version	string Выпадающий список	По умолчанию предлагается последняя версия В выпадающем списке версии отсортированы от старшей к младшей (сверху вниз) Например, 3.2.0.17		Версия задается в виде версии дистрибутива в Nexus.

Имя параметра	label	type	constrain	Доп. аттрибуты	Комментарий
Description	Описание версии	string	Заполняется в зависимости от версии		Краткое описание версии - чем одна версия отличается от другой. Хранится в дистрибутиве.
installation_type	Тип установки	string Выпадающий список	1. standalone-postgresql-only - установка только PostgreSQL SE на один узел 2. standalone-patroni-etcd-pgbouncer - установка всех компонент кластера на один узел 3. cluster-patroni-etcd-pgbouncer - установка PostgreSQL SE с сервисами patroni, pgbouncer и etcd 4. cluster-patroni-etcd-pgbouncer-haproxy - установка PostgreSQL SE с сервисами patroni, pgbouncer, etcd и haproxy 5. *cluster-patroni-etcd-pgbouncer-load_balancer - установка PostgreSQL SE с сервисами patroni, pgbouncer, etcd, haproxy и сервисом балансировки нагрузки		пункт 5 должен быть в списке, но неактивный (1Q2020). В 2Q2020 делаем компенс. меру - заказ через ЗНИ/ЗНО (автоматическое создание ЗНИ/ЗНО). В конце 2Q2020 по готовности БН в ДИ делаем целевую интеграцию.
port	Порт для подключения к базе	integer	5432		Дефолтный порт доступа к базе
PGDATA	Расположение базы данных	string			Расположение базы данных на диске. Проверяется наличие свободного места на диске

Имя параметра	label	type	constrain	Доп. аттрибуты	Комментарий
tablespace_name	Имя табличного пространства по умолчанию	string			Имя табличного пространства, создающегося инсталятором
tablespace_location	Расположение табличного пространства по умолчанию	string			Расположение табличного пространства по умолчанию
Database_name	Имя базы данных	string			Имя базы данных, создающейся инсталятором
customer	- (не выводится на UI портала)	string	логин ldap в зависимости от среды (Sigma/Alpha)		Логин AD заказчика стенда.
support	- (не выводится на UI портала)	список	Список логинов ldap в зависимости от среды (Sigma/Alpha)		Список логинов группы сопровождения
sec_admin	- (не выводится на UI портала)	string	логин ldap в зависимости от среды (Sigma/Alpha)		Временно не передавать, пока не появится ответственное подразделение

Имя параметра	label	type	constrain	Доп. аттрибуты	Комментарий
as_admins	Администраторы АС	список	Список логинов ldap в зависимости от среды (Sigma/Alpha)		Список Администраторов АС - владельцев базы данных по умолчанию
as_TUZ	ТУЗы АС	список	Список имен ТУЗ, которые будут созданы в БД для работы приложения АС		

Обработка параметров производится в Jenkins по следующему алгоритму:

- Python скрипт читает json с параметрами для хостов(hostname, login, password), который приходит в один из jenkins параметров.
- Остальные jenkins параметры читаются обрабатываются из groovy и передаются(при необходимости) в ansible.
- Далее ansible читает хостовые переменные с помощью dynamic inventory plugin'a и запускает плейбук с переданными ранее jenkins'ом переменными

Все переменные имеют дефолтные значения для разворота кластера по умолчанию.

6.1.5 3. Алгоритм бизнес-логики

В рамках работы реализовывается установка каждой утилиты в виде отдельной атомарной ansible-роли. А также один универсальный ansible-playbook, в котором реализовано развертывание всех необходимых схем, путем вызова атомарных ролей и тэгирования.

При написании ролей учитывается идемпотентность запуска ролей: на первом этапе с ограничением, что развертывание осуществляется "с нуля", на втором и последующих этапах ограничение снимается и учитываются все варианты идемпотентности, в том числе и обновление утилит и конфигураций.

Существует 4 общие для всех схем развертывания роли:

- checkup
- common
- postgresql
- configure

В роли "checkup" проверяется состояние серверов/КТС/нод, на которые предполагается осуществить развертывание. Проверка заключается в сравнении версий ansible, python и наличие необходимых пакетов, их версий. А также проверяется наличие свободного места на дисках, на которые планируется осуществить установку (PGHOME, PGDATA, Tablespace), согласно параметрам из Таблицы 3

В роли "common" осуществляется доустановка пакетов, отсутствие которых было выявлено в предыдущей роли.

В роли "postgresql" осуществляется установка пакета postgresql-sber-edition указанной в инвентори версии из RPM с последующим конфигурированием, исходя из характеристик ноды и согласно требованиям кибербезопасности.

Для реализации **требований кибербезопасности** в качестве параметра инвентори объявлен список администраторов АС ("as_admins" Таблица 3), которые в результате развертывания создаются в качестве пользователей СУБД и прописываются в файл pg_hba.conf в виде:

```
host all as_admins 0.0.0.0/0 ldap ldapserver="ca.sbrf.ru" ldapport=3268 ldapbasedn="" ldapbinddn="CN=SBT-SA-POSTGRESQL_A,CN=Users,DC=ca,DC=sbrf,DC=ru" ldapbindpasswd="passwd" ldapsearchattribute=cn
либо
host all as_admins 0.0.0.0/0 ldap ldapserver="sigma.sbrf.ru" ldapport=3268 ldapbasedn=""
 ldapbinddn="cn=SBT-SA-POSTGRESQL_S,OU=Service Accounts,OU=SBT,OU=Sberbank,dc=sigma,dc=sbrf,dc=ru"
 ldapbindpasswd="passwd" ldapsearchattribute=cn
```

Также для всех создаваемых в процессе развертывания администраторов настраивается ролевой аудит событий в расширении pgaudit.

В роли "configure" осуществляется создание табличного пространства, групповых ролей (согласно Ролевой модели), создание пользовательской базы данных, конфигурирование аудита событий.

Помимо общих ролей существует ряд специфичных для хостов ролей:

- etcd
- patroni
- pgbouncer
- confd
- HAProxy

В каждой из этих ролей осуществляется разворачивание соответствующей утилиты и настройка для работы в кластере. В результате разворачивания запускаются проверки доступности сетевых интерфейсов, по которым происходит связь экземпляров утилит.

Ограничения:

1. Для маскирования паролей использовать ansible-vault
2. Скачивание дистрибутива, в состав которого входит инсталятор, не является задачей инсталятора. Реализовывается вне его, например скриптами bash или groovy.
3. Распаковка дистрибутива, в состав которого входит инсталятор, не является задачей инсталятора. Реализовывается вне его, например скриптами bash или groovy.
4. запрещается использовать ролевые переменные

Общий вид структуры инсталятора:

```
group_vars
        all.yml
inventories/
   production/
      hosts.ini
      group_vars/
         group1.yml
         group2.yml
   IFT/PSI/
      hosts.ini
      group_vars/
         group1.yml
         group2.yml
    NT/
      hosts.ini
      group_vars/
         group1.yml
         group2.yml
library/
module_utils/
filter_plugins/

playbook.yml

roles/
    checkup
        common
        postgresql
        configure
    patroni/
    etcd/
    pgbouncer/
    confd/
        HAProxy/
```

Содержание файла hosts.ini

```
[cluster:children]
postgres_group
etcd_group

[postgres_group:children]
postgres_nodes

[etcd_group:children]
etcd_nodes

[postgres_group:vars]
ansible_connection=ssh

[etcd_group:vars]
ansible_connection=ssh


#TEMPLATE
#[postgres_nodes]
#master                ansible_host=hostname or ip address               ansible_user=sudo user on linux
host         ansible_password=password for linux sudo user
#replica        ansible_host=hostname or ip address               ansible_user=sudo user on linux host
ansible_password=password for linux sudo user
#[etcd_nodes]
#etcd                ansible_host=hostname or ip address               ansible_user=sudo user on linux
host         ansible_password=password for linux sudo user
#If you want add custom variable for any host, just do it here.
# [postgres_nodes]
#master                ansible_host=hostname or ip address               ansible_user=sudo user on linux
host         ansible_password=password for linux sudo user        test_variable=test
#replica        ansible_host=hostname or ip address               ansible_user=sudo user on linux host
ansible_password=password for linux sudo user        test_variable=test
# [etcd_nodes]
#etcd                ansible_host=hostname or ip address               ansible_user=sudo user on linux
host         ansible_password=password for linux sudo user               test_variable=test
```

Для возможности выполнять проверки включения настроек кибербезопасности в БД созданы специальные функции:

```
postgres=# select * from check_password_policy_is_on();
 check_password_policy_is_on
-----------------------------
 t
(1 row)

postgres=# select * from check_pg_audit_is_on();
 check_pg_audit_is_on
----------------------
 t
(1 row)

postgres=# select * from check_ldap_is_on();
 check_ldap_is_on
------------------
 t
(1 row)

postgres=# select * from check_roles_is_on();
 check_roles_is_on
-------------------
 t
(1 row)

postgres=# select * from check_tde_is_on();
 check_tde_is_on
-----------------
 f
(1 row)

postgres=# select * from check_admin_protect_is_on();
 check_admin_protect_is_on
---------------------------
 f
(1 row)

postgres=# select * from check_ssl_is_on();
 check_ssl_is_on
------------------
 f
(1 row)
```

7 Руководство для разработчика прикладных сервисов

Реализованная функциональность не содержит инструкций для разработчика прикладных сервисов.