

Интеграция PostgresPro со службой каталога ALD Pro



11/07/2025

Содержание

1	ВВЕДЕНИЕ.....	3
2	ОПИСАНИЕ ТЕСТОВОЙ СРЕДЫ	4
3	ПОДГОТОВКА ТЕСТОВОГО СТЕНДА.....	5
4	НАСТРОЙКА ВНЕШНЕЙ АУТЕНТИФИКАЦИИ В PostgresPro ЧЕРЕЗ ALD PRO	6
4.1	Регистрация службы базы данных на контроллере домена.....	6
4.2	Настройка сервисной учетной записи для LDAP Bind.....	6
4.3	Настройка сервисной записи для Kerberos	7
4.4	Настройка Kerberos на сервере PostgresPro	8
4.5	Настройка GSSAPI-аутентификации в PostgresPro.....	9
4.6	Настройка pg_hba.conf для аутентификации	9
4.7	Настройка pg_ident.conf для маппинга пользователей.....	10
4.8	Проверка GSSAPI-аутентификации	11
4.9	Проверка LDAP-аутентификации.....	11
5	СОЗДАНИЕ ТЕСТОВОЙ БАЗЫ ДАННЫХ И НАЗНАЧЕНИЕ ПРАВ	12
5.1	Создание базы данных testdb.....	12
5.2	Назначение прав ролям из pg2ldap	12
5.3	Проверка прав доступа.....	13
6	АВТОМАТИЗАЦИЯ ИНТЕГРАЦИИ С ПОМОЩЬЮ LDAP2PG	15
6.1	Установка и настройка ldap2pg.....	15
6.2	Применение синхронизации	18
6.3	Настройка журналирования и автоматической синхронизации.....	18
7	ПРОВЕРКА И ТЕСТИРОВАНИЕ	20
7.1	Проверка работы аутентификации	20
7.2	Проверка прав доступа.....	20
7.3	Проверка логов.....	20
8	РЕКОМЕНДАЦИИ ПО ЭКСПЛУАТАЦИИ	21
8.1	Безопасность	21

8.2 Мониторинг	21
8.3 Обслуживание	21

1 ВВЕДЕНИЕ

PostgresPro — это современная промышленная система управления базами данных, созданная на основе открытого PostgreSQL и значительно расширенная для корпоративного применения. Она обеспечивает высокую производительность, надежность и безопасность для работы с большими объемами данных, поддерживает масштабируемые решения, предоставляет инструменты репликации, резервного копирования, мониторинга и отличается официальной сертификацией для государства и бизнеса.

Интеграция с ALD Pro позволяет централизованно управлять пользователями и их правами через корпоративный каталог, дает единый механизм аутентификации с поддержкой LDAP и Kerberos, автоматизирует назначение прав группам пользователей, а также упрощает соблюдение корпоративных и государственных стандартов безопасности и контроля доступа.

2 ОПИСАНИЕ ТЕСТОВОЙ СРЕДЫ

Для интеграции PostgresPro 17 с ALD Pro 3.0.0 используется следующая тестовая среда:

Контроллер домена (ALD Pro):

- имя хоста: dc-1
- IP-адрес: 192.168.50.216
- операционная система: Astra Linux 1.7.7 UU2
- версия ALD Pro: 3.0.0
- домен: ALD.COMPANY.LAN
- DNS-сервер: dc-1.ald.company.lan
- сервисный аккаунт для PostgresPro: postgres/pgpro.ald.company.lan@ALD.COMPANY.LAN

Сервер баз данных PostgresPro:

- имя хоста: pgpro
- IP-адрес: 192.168.50.84
- операционная система: Astra Linux 1.7.7
- установленная версия: PostgresPro 17.5 (PostgresPro Enterprise Edition)
- домен: ALD.COMPANY.LAN
- каталог данных: /var/lib/pgpro/ent-17/data
- пользователь базы данных: test_user

Требования к сети:

- подсеть: 192.168.50.0/24
- все хосты должны разрешаться через DNS контроллера домена
- открытые порты: 53 (DNS), 88 (Kerberos), 389 (LDAP), 5432 (PostgresPro)

Унифицированные параметры для всей инструкции:

- домен DN: dc=ald,dc=company,dc=lan
- контроллер домена FQDN: dc-1.ald.company.lan
- сервисная учетная запись Bind DN: uid=ldap-bind,cn=sysaccounts,cn=etc,dc=ald,dc=company,dc=lan
- пароль сервисной учетной записи: <SECURE_PASSWORD>

3 ПОДГОТОВКА ТЕСТОВОГО СТЕНДА

Убедитесь, что оба сервера (dc-1 и pgpro) настроены в одной подсети и могут взаимодействовать друг с другом:

```
ping dc-1.ald.company.lan
```

Проверьте настройки DNS на сервере PostgresPro:

```
cat /etc/resolv.conf
```

Убедитесь, что в файле в качестве DNS-сервера указан контроллер домена :

```
nameserver 192.168.50.216  
search ald.company.lan
```

Проверьте разрешение имен:

```
nslookup dc-1.ald.company.lan  
nslookup pgpro.ald.company.lan
```

Оба имени должны разрешаться в соответствующие IP-адреса.

4 НАСТРОЙКА ВНЕШНЕЙ АУТЕНТИФИКАЦИИ В PostgresPro ЧЕРЕЗ ALD PRO

4.1 Регистрация службы базы данных на контроллере домена

1. Создайте тестового пользователя в ALD Pro:

```
ipa user-add test_user \  
  --first=Professional \  
  --last=Postgres \  
  --shell=/bin/bash \  
  --password
```

Система запросит установить пароль для пользователя. Пароль должен соответствовать политике сложности ALD Pro.

2. Создайте группы для управления доступом к PostgresPro:

```
ipa group-add pg_users --desc="PostgresPro Regular Users"  
ipa group-add pg_readonly --desc="PostgresPro Read-Only Users"
```

Группа `pg\_users` будет содержать пользователей с правами на запись. Группа `pg\_readonly` будет содержать пользователей только с правами на чтение.

3. Добавьте тестового пользователя в группу:

```
ipa group-add-member pg_users --users=test_user
```

Проверьте членство в группе:

```
ipa group-show pg_users
```

4.2 Настройка сервисной учетной записи для LDAP Bind

Для интеграции PostgresPro с ALD Pro необходимо создать специальную сервисную учетную запись, которая будет использоваться для аутентификации через LDAP. Эта учетная запись не является POSIX-пользователем, не имеет прав на вход в домен и используется только для чтения LDAP.

Создайте файл для добавления сервисной учетной записи:

```
sudo nano /tmp/ldap-bind.update
```

Добавьте следующее содержимое в файл:

```
dn: uid=ldap-bind,cn=sysaccounts,cn=etc,dc=ald,dc=company,dc=lan
add:objectclass: account
add:objectclass: simplesecurityobject
add:uid: ldap-bind
add:userPassword: <SECURE_PASSWORD>
add:passwordExpirationTime: 20380119031407Z
add:nsIdleTimeout: 0
```

Замените ``<SECURE_PASSWORD>`` на безопасный пароль в соответствии с политиками безопасности вашей организации. При необходимости адаптируйте ``passwordExpirationTime`` в соответствии с политиками безопасности.

Примените изменения:

```
kinit admin
sudo ipa-ldap-updater /tmp/ldap-bind.update
```

Система запросит пароль администратора ALD Pro. Убедитесь, что команда выполнена без ошибок.

Проверьте создание учетной записи с использованием аутентификации от учетной записи admin (анонимный доступ запрещен):

```
ldapsearch -x -H ldap://dc-1.ald.company.lan \
-D "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan" \
-W -b "uid=ldap-bind,cn=sysaccounts,cn=etc,dc=ald,dc=company,dc=lan" uid
```

Убедитесь, что вывод содержит информацию о созданной учетной записи.

4.3 Настройка сервисной записи для Kerberos

Для обеспечения единого входа (SSO) и аутентификации через GSSAPI необходимо создать сервисную запись в Kerberos.

Получите Kerberos-билет администратора:

```
kinit admin
```

Задайте переменные с нужными значениями:

```
IP_ADDRESS=192.168.50.84
HOSTNAME=pgpro.ald.company.lan
DOMAIN_CONTROLLER=dc-1.ald.company.lan
```

Проверьте, добавлен ли сервер в домен:

```
sudo ipa host-find $HOSTNAME
```

Если сервер не найден, добавьте его (если необходимо):

```
sudo ipa host-add --force --ip-address=$IP_ADDRESS $HOSTNAME
```

Добавьте сервисную службу (SPN):

```
sudo ipa service-add postgres/$HOSTNAME
```

Обратите внимание, что имя сервиса должно быть `postgres/`, так как это стандартное имя для PostgresPro.

Проверьте регистрацию сервиса:

```
sudo ipa service-find postgres/$HOSTNAME
```

4.4 Настройка Kerberos на сервере PostgresPro

Настройте файл `/etc/krb5.conf`:

```
sudo nano /etc/krb5.conf
```

Убедитесь, что файл содержит следующие настройки:

```
[libdefaults]
default_realm = ALD.COMPANY.LAN
dns_lookup_realm = false
dns_lookup_kdc = true
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true

[realms]
ALD.COMPANY.LAN = {
  kdc = dc-1.ald.company.lan
  admin_server = dc-1.ald.company.lan
}

[domain_realm]
.ald.company.lan = ALD.COMPANY.LAN
ald.company.lan = ALD.COMPANY.LAN
```

Проверьте подключение к контроллеру домена:

```
kinit admin
```

Проверьте полученный билет:

```
klist
```

4.5 Настройка GSSAPI-аутентификации в PostgresPro

1. Создайте keytab-файл для сервиса PostgresPro:

```
ipa-getkeytab -p postgres/pgpro.ald.company.lan \  
-k /var/lib/pgpro/ent-17/data/pgpro.keytab \  
-s dc-1.ald.company.lan
```

Этот файл содержит криптографические ключи для аутентификации сервиса.

2. Настройте права доступа к keytab-файлу:

```
sudo chown postgres:postgres /var/lib/pgpro/ent-17/data/pgpro.keytab  
sudo chmod 400 /var/lib/pgpro/ent-17/data/pgpro.keytab
```

Проверьте содержимое keytab:

```
klist -k /var/lib/pgpro/ent-17/data/pgpro.keytab
```

3. Настройте файл `postgresql.conf`:

```
sudo nano /var/lib/pgpro/ent-17/data/postgresql.conf
```

Добавьте или измените следующие параметры:

```
krb_server_keyfile = '/var/lib/pgpro/ent-17/data/pgpro.keytab'  
listen_addresses = '*'
```

4.6 Настройка pg_hba.conf для аутентификации



Важно: правильная настройка файла pg_hba.conf критична для работы аутентификации. Postgre Pro сопоставляет подключения по принципу «первое совпадение», поэтому порядок правил и точность указания параметров (DATABASE, USER, ADDRESS) влияют на успех аутентификации.

Отредактируйте файл pg_hba.conf:

```
sudo nano /var/lib/pgpro/ent-17/data/pg_hba.conf
```

Добавьте следующие строки ****перед**** более общими правилами (например, перед 127.0.0.1/32 для локального подключения):

```
# TYPE DATABASE USER ADDRESS METHOD OPTIONS
# GSSAPI authentication with mapping
host all all 192.168.50.0/24 gss map=kerberos

# LDAP authentication as fallback (if GSS is not applicable)
host all all 192.168.50.0/24 ldap ldapserver=dc-1.ald.company.lan \
\
ldapbasedn="cn=users,cn=accounts,dc=ald,dc=company,dc=lan" \
ldapsearchattribute=uid \
ldapbinddn="uid=ldap-bind,cn=sysaccounts,cn=etc,dc=ald,dc=company,dc=lan" \
ldapbindpasswd "<SECURE_PASSWORD>"

# Local connections - trust method (optional for local admin access)
local all postgres trust

# IPv4 local connections
host all all 127.0.0.1/32 md5

# IPv6 local connections
host all all ::1/128 md5
```

Важные параметры в pg_hba.conf:

- **TYPE**: тип соединения (host для TCP/IP, hostgssenc для обязательного GSS-шифрования, local для Unix-сокета);
- **DATABASE**: разрешённая база (all — любая база или конкретное имя, например testdb);
- **USER**: разрешённый пользователь (all — любой пользователь или конкретное имя);
- **ADDRESS**: CIDR-адрес сети (например, 192.168.50.0/24 означает подсеть с маской /24);
- **METHOD**: метод аутентификации (gss, ldap, md5, trust и др.);
- **OPTIONS**: параметры метода (для gss: map=kerberos, include_realm; для ldap: ldapserver, ldapbasedn и др.).

4.7 Настройка pg_ident.conf для маппинга пользователей

Файл pg_ident.conf используется для преобразования имён пользователей, полученных при GSSAPI-аутентификации, в имена ролей PostgresPro.

Отредактируйте файл pg_ident.conf:

```
sudo nano /var/lib/pgpro/ent-17/data/pg_ident.conf
```

Добавьте строку для маппинга Kerberos-принципалов:

```
kerberos /^(.*)@ALD\ .COMPANY\ .LAN$/ \1
```

Объяснение строки маппинга:

- **kerberos** — имя карты маппинга (должно совпадать с параметром `map=` в pg_hba.conf);
- **/^(.*)@ALD\ .COMPANY\ .LAN\$/** — регулярное выражение для поиска принципала (ищет всё перед @ALD.COMPANY.LAN);
- **\1** — замена на первую найденную группу (т.е. имя пользователя без @REALM).

Пример преобразования:

- входящий принципал: `test\_user@ALD.COMPANY.LAN`
- результат маппинга: `test\_user` (роль в PostgresPro)

Перезагрузите конфигурацию PostgresPro:

```
sudo systemctl reload postgrespro-ent-17
```

Или через SQL (если подключены как админ):

```
sudo -u postgres psql -c "SELECT pg_reload_conf();"
```

4.8 Проверка GSSAPI-аутентификации

Получите билет Kerberos для тестового пользователя:

```
kinit test_user@ALD.COMPANY.LAN
```

Проверьте полученный билет:

```
klist
```

Убедитесь, что билет успешно создан.

Подключитесь к PostgresPro:

```
psql -h pgpro.ald.company.lan -U test_user -d postgres
```

Если аутентификация успешна, вы увидите приглашение psql.

Проверьте логи PostgresPro:

```
tail -n 10 /var/lib/pgpro/ent-17/data/log/postgresql-*.log
```

Убедитесь, что в логах есть запись об успешной аутентификации через GSSAPI.

4.9 Проверка LDAP-аутентификации

Если GSSAPI не применяется (например, на клиентах без Kerberos-билета), PostgresPro использует LDAP-аутентификацию как fallback.

Попробуйте подключиться, указав пароль:

```
psql -h pgpro.ald.company.lan -U test_user -d postgres -W
```

Введите пароль пользователя test_user из ALD Pro.

Убедитесь, что подключение работает и проверьте логи для записи об LDAP-аутентификации.

5 СОЗДАНИЕ ТЕСТОВОЙ БАЗЫ ДАННЫХ И НАЗНАЧЕНИЕ ПРАВ

5.1 Создание базы данных testdb

Подключитесь к PostgresPro под суперпользователем postgres:

```
sudo -u postgres psql
```

Создайте базу данных testdb:

```
CREATE DATABASE testdb;
```

Или используйте утилиту createdb из командной строки:

```
sudo -u postgres createdb testdb
```

Проверьте создание базы:

```
sudo -u postgres psql -c "\l" | grep testdb
```

5.2 Назначение прав ролям из pg2ldap

После синхронизации через ldap2pg в PostgresPro будут созданы роли (группы) из LDAP, например `ldap\_pg\_users` и `ldap\_pg\_readonly`. Чтобы пользователи из этих групп могли работать с базой testdb, необходимо выдать им соответствующие права.

1. Подключитесь к базе данных testdb:

```
sudo -u postgres psql -d testdb
```

2. Для группы ldap_pg_users (пользователи с правами на чтение и запись):

```
-- Разрешить подключение к базе
GRANT CONNECT ON DATABASE testdb TO ldap_pg_users;

-- Разрешить использовать схему public
GRANT USAGE ON SCHEMA public TO ldap_pg_users;

-- Разрешить SELECT, INSERT, UPDATE, DELETE на все существующие таблицы
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO ldap_pg_users;

-- Разрешить работу с последовательностями
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO ldap_pg_users;
```

```
-- Установить права по умолчанию для новых таблиц
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT, INSERT, UPDATE, DELETE ON
TABLES TO ldap_pg_users;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT USAGE, SELECT ON SEQUENCES TO
ldap_pg_users;
```

3. Для группы ldap_pg_readonly (пользователи только с правами на чтение):

```
-- Разрешить подключение к базе
GRANT CONNECT ON DATABASE testdb TO ldap_pg_users;

-- Разрешить использовать схему public
GRANT USAGE ON SCHEMA public TO ldap_pg_users;

-- Разрешить SELECT, INSERT, UPDATE, DELETE на все существующие таблицы
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO ldap_pg_users;

-- Разрешить работу с последовательностями
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO ldap_pg_users;

-- Установить права по умолчанию для новых таблиц
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT, INSERT, UPDATE, DELETE ON
TABLES TO ldap_pg_users;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT USAGE, SELECT ON SEQUENCES TO
ldap_pg_users;
```

4. Проверьте назначенные права:

```
-- Проверка прав на таблицы
SELECT table_schema, table_name, grantee, privilege_type
FROM information_schema.table_privileges
WHERE table_schema = 'public'
ORDER BY grantee, table_name;

-- Проверка прав по умолчанию
SELECT * FROM pg_default_acl;
```

5. Выйдите из psql:

```
\q
```

5.3 Проверка прав доступа

Проверьте, может ли пользователь из группы ldap_pg_users подключиться и работать с базой testdb:

```
psql -h pgpro.ald.company.lan -U test_user -d testdb
```

Если подключение успешно, проверьте доступ к таблицам:

```
SELECT * FROM information_schema.tables WHERE table_schema = 'public';
```

```
CREATE TABLE test_table (id SERIAL, name VARCHAR(100));  
INSERT INTO test_table (name) VALUES ('test');  
SELECT * FROM test_table;  
DROP TABLE test_table;
```

Если все команды выполнены успешно, права назначены корректно.

6 АВТОМАТИЗАЦИЯ ИНТЕГРАЦИИ С ПОМОЩЬЮ LDAP2PG

6.1 Установка и настройка ldap2pg

Существует несколько способов установки пакета. Предпочтительным является установка из репозитория PostgresPro (<https://repo.postgrespro.ru>). Или, при наличии ISO образа дистрибутива одной из версий PostgresPro, можно установить оффлайн.

Другой способ - установка из открытого репозитория разработчиков утилиты через wget:

```
wget https://github.com/dalibo/ldap2pg/releases/download/v6.4.2/ldap2pg_6.4.2_linux_amd64.deb
sudo dpkg -i ldap2pg_6.4.2_linux_amd64.deb
```

Проверьте установку:

```
ldap2pg --version
```

Убедитесь, что вывод содержит версию утилиты (например, `ldap2pg 6.4.2`).

Создайте специального пользователя в PostgresPro для синхронизации:

```
sudo -u postgres psql -c "CREATE ROLE ldap2pg LOGIN CREATEDB CREATEROLE;"
sudo -u postgres psql -c "ALTER ROLE ldap2pg SET createrole_self_grant TO 'inherit,set';"
```

Этот пользователь будет использоваться для управления ролями в PostgresPro. Параметр `createrole\_self\_grant` необходим для корректной работы с группами.

Создайте конфигурационный файл:

```
sudo nano /etc/ldap2pg.yml
```

Добавьте следующее содержимое в формате yaml:

```
version: 6
postgres:
  dsn: postgres://ldap2pg@127.0.0.1:5432/postgres
  databases_query:
    - postgres
    - testdb
  fallback_owner: postgres
  roles_blacklist: ["pg_*", "postgres"]

ldap:
  known_rdns:
```

- cn
- dc
- ou
- uid

rules:

1. Создание LDAP-групп в Postgres

- description: Создание групп ldap_pg_users и ldap_pg_readonly

grants:

- database: __all__
- owner: __auto__
- role: '{cn}'
- schema: __all__

ldapssearch:

- base: cn=groups,cn=accounts,dc=ald,dc=company,dc=lan
- filter: (&(objectClass=groupOfNames)(|(cn=pg_users)(cn=pg_readonly)))
- scope: sub

roles:

- comment: Managed by ldap2pg
- name: "ldap_{cn}"
- options:
 - BYPASSRLS: **false**
 - CONNECTION LIMIT: -1
 - CREATEDB: **false**
 - CREATEROLE: **false**
 - INHERIT: **true**
 - LOGIN: **true**
 - REPLICATION: **false**
 - SUPERUSER: **false**
- parents: []

2. Пользователи группы pg_users

- description: Создание пользователей из группы pg_users

grants: []

ldapssearch:

- base: cn=users,cn=accounts,dc=ald,dc=company,dc=lan
- filter: (&(objectClass=person)

(memberOf=cn=pg_users,cn=groups,cn=accounts,dc=ald,dc=company,dc=lan))

scope: sub

roles:

- comment: Managed by ldap2pg
- name: '{uid}'
- options:
 - BYPASSRLS: **false**
 - CONNECTION LIMIT: -1
 - CREATEDB: **false**
 - CREATEROLE: **false**
 - INHERIT: **true**
 - LOGIN: **true**
 - REPLICATION: **false**
 - SUPERUSER: **false**
- parents:
 - name: ldap_pg_users

3. Пользователи группы pg_readonly

- description: Создание пользователей из группы pg_readonly

grants: []

ldapssearch:

```
base: cn=users,cn=accounts,dc=ald,dc=company,dc=lan
filter: (&(objectClass=person)
(memberOf=cn=pg_readonly,cn=groups,cn=accounts,dc=ald,dc=company,dc=lan))
scope: sub
roles:
- comment: Managed by ldap2pg
  name: '{uid}'
  options:
    BYPASSRLS: false
    CONNECTION_LIMIT: -1
    CREATEDB: false
    CREATEROLE: false
    INHERIT: true
    LOGIN: true
    REPLICATION: false
    SUPERUSER: false
  parents:
    - name: ldap_pg_readonly

logging:
  level: info
  format: '%(asctime)s [%(asctime)s] %(message)s'
  file: /var/log/ldap2pg.log
```



Важное примечание о роли групп:

В конфигурации ldap2pg используется встроенный черный список ролей `roles_blacklist`: `["pg_*", "postgres"]`, который исключает из синхронизации все роли, начинающиеся с `pg_`. Поскольку системные роли PostgresPro (`pg_read_all_data`, `pg_write_all_data` и т.д.) начинаются с этого префикса, рекомендуется использовать другое имя для групп в СУБД.

В данной конфигурации группы из ALD Pro (`pg_users` и `pg_readonly`) создаются в Postgres с префиксом `ldap_`, то есть как `ldap_pg_users` и `ldap_pg_readonly`. Это позволяет избежать конфликта с встроенным черным списком и системными ролями PostgresPro.

Добавьте или измените параметры в файле `/etc/ldap/ldap.conf`:

```
sudo nano /etc/ldap/ldap.conf
```

Приведите файл конфигурации к виду:

```
URI ldap://dc-1.ald.company.lan
BASE cn=accounts,dc=ald,dc=company,dc=lan
BINDDN uid=ldap-bind,cn=sysaccounts,cn=etc,dc=ald,dc=company,dc=lan
SIZELIMIT 500
TIMELIMIT 10
CONNECT_TIMEOUT 5
#SASL_MECH GSSAPI
```

Укажите пароль для каталога:

```
printf '%s\n' '<SECURE_PASSWORD>' | sudo tee /etc/ldap2pg/ldappass >/dev/null
sudo chmod 600 /etc/ldap2pg/ldappass
sudo chown postgres:postgres /etc/ldap2pg/ldappass
```

Замените `<SECURE\_PASSWORD>` на пароль сервисной учетной записи ldap-bind.
При необходимости создайте папку ldap2pg:

```
sudo mkdir /etc/ldap2pg
```

Проверьте конфигурацию (режим dry-run без внесения изменений):

```
cd /tmp
sudo -u postgres LDAPPASSWORD_FILE=/etc/ldap2pg/ldappass ldap2pg -c /etc/ldap2pg.yml
```

Смена директории на /tmp необходима для исключения проблем с доступом к файлу ldaprc. Этот режим показывает, какие изменения будут применены без их фактического выполнения.

Проанализируйте вывод на наличие ошибок.

6.2 Применение синхронизации

После проверки конфигурации выполните синхронизацию с параметром `--real` для внесения изменений:

```
cd /tmp
sudo -u postgres LDAPPASSWORD_FILE=/etc/ldap2pg/ldappass ldap2pg --real -c /etc/ldap2pg.yml
```

Проверьте созданные роли:

```
sudo -u postgres psql -c "\du"
```

Убедитесь, что отображаются группы `ldap\_pg\_users` и `ldap\_pg\_readonly`, а также пользователь `test\_user`.

6.3 Настройка журналирования и автоматической синхронизации

Настройте ротацию логов:

```
sudo nano /etc/logrotate.d/ldap2pg
```

Добавьте следующее содержимое:

```
/var/log/ldap2pg.log {
    daily
    missingok
}
```

```
rotate 7
compress
delaycompress
notifempty
create 640 postgres postgres
sharedscripts
}
```

Настройте автоматическую синхронизацию через cron (выполняется раз в час):

```
sudo nano /etc/cron.d/ldap2pg
```

Добавьте следующее содержимое:

```
# Синхронизация раз в час от имени пользователя postgres
0 * * * * postgres cd /tmp && LDAPPASSWORD_FILE=/etc/ldap2pg/ldappass ldap2pg --real
-c /etc/ldap2pg.yml >> /var/log/ldap2pg.log 2>&1
```

Параметр `--real` запускает утилиту в режиме внесения изменений. Параметр `cd /tmp` обеспечивает работу в безопасной директории.

Проверьте работу журналирования:

1. Запустите синхронизацию вручную:

```
cd /tmp
sudo -u postgres LDAPPASSWORD_FILE=/etc/ldap2pg/ldappass ldap2pg --real -c /etc/
ldap2pg.yml
```

2. Проверьте логи:

```
tail -n 20 /var/log/ldap2pg.log
```

Убедитесь, что в логах есть запись о синхронизации.

7 ПРОВЕРКА И ТЕСТИРОВАНИЕ

7.1 Проверка работы аутентификации

Тест LDAP-аутентификации (без Kerberos-билета):

```
psql -h pgpro.ald.company.lan -U test_user -d testdb -W
```

7.2 Проверка прав доступа

Создайте тестовую таблицу и проверьте возможность работы:

```
CREATE TABLE test_table (id SERIAL PRIMARY KEY, data VARCHAR(100));  
INSERT INTO test_table (data) VALUES ('test_data');  
SELECT * FROM test_table;
```

Убедитесь, что все операции выполняются успешно для пользователя из группы `ldap_pg_users`.

Для пользователя из группы `ldap_pg_readonly` попытайтесь вставить данные (должно быть отклонено):

```
INSERT INTO test_table (data) VALUES ('another_data'); -- Должна быть ошибка
```

7.3 Проверка логов

Проверьте логи PostgresPro:

```
tail -f /var/lib/pgpro/ent-17/data/log/postgresql-*.log
```

Проверьте логи ldap2pg:

```
tail -f /var/log/ldap2pg.log
```

8 РЕКОМЕНДАЦИИ ПО ЭКСПЛУАТАЦИИ

8.1 Безопасность

- Храните пароль сервисной учетной записи в защищенном файле с ограниченными правами доступа.
- Регулярно проверяйте права доступа к конфигурационным файлам.
- Используйте HTTPS/TLS для LDAP-соединений в production-среде.
- Ограничивайте доступ к серверу PostgresPro только необходимым подсетям через pg_hba.conf.

8.2 Мониторинг

- Регулярно проверяйте логи PostgresPro и ldap2pg на ошибки.
- Убедитесь, что синхронизация через ldap2pg выполняется по расписанию.
- Отслеживайте изменения в LDAP и своевременно применяйте их в PostgresPro.

8.3 Обслуживание

Резервное копирование конфигурации:

```
sudo cp /etc/ldap2pg.yml /etc/ldap2pg.yml.backup.$(date +%Y%m%d)
```

Ручное обновление прав при необходимости:

```
cd /tmp
sudo -u postgres LDAPPASSWORD_FILE=/etc/ldap2pg/ldappass ldap2pg --real -c /etc/ldap2pg.yml
```