

# **BIOTECH**

## **DOCUMENT D'EXPLOITATION (DEX)**

**Procédures d'installation et de configuration**

# Table des matières

Table des matieres.....	2
1. Introduction .....	4
1.1 Objectifs du document .....	4
1.2 Architecture globale .....	4
2. Installation bthanu01p (Annuaire LDAP) .....	5
2.1 Installation Rocky Linux .....	6
2.2 Installation 389 Directory Server.....	7
2.3 Verification.....	7
3. Installation bthbdd01p (PostgreSQL Master) .....	100
3.1 Installation Rocky Linux .....	100
3.2 Installation PostgreSQL .....	101
3.3 Creation de la base et des utilisateurs.....	111
3.4 Configuration pour la replication .....	111
4. Installation bthbdd02p (PostgreSQL Slave) .....	122
4.1 Installation Rocky Linux .....	122
4.2 Configuration de la replication .....	122
4.3 Verification de la replication.....	122
5. Installation bthstk01p (Stockage secours).....	18
5.1 Installation Rocky Linux .....	18
5.2 Configuration du stockage .....	18
6. Procedures de sauvegarde .....	19
6.1 Configuration SSH (sur bthbdd01p).....	19
6.2 Script de sauvegarde.....	19
6.3 Planification automatique (cron) .....	19
7. Procedures de restauration .....	21
7.1 Restauration depuis le backup.....	21
7.2 Basculement Master/Slave (failover) .....	21
8. Informations de connexion .....	29
8.1 Comptes systeme .....	29
8.2 Comptes applicatifs .....	29
9. Contraintes PRA.....	23
10. Contacts .....	29



# 1. Introduction

Ce document décrit les procédures d'installation, de configuration et d'exploitation de l'infrastructure BioTech. Il est destiné aux équipes d'exploitation et d'administration système.

## 1.1 Objectifs du document

- Fournir les procédures d'installation des serveurs
- Décrire la configuration des services (LDAP, PostgreSQL, Sauvegarde)
- Documenter les procédures de sauvegarde et restauration
- Permettre la reproductibilité de l'infrastructure

## 1.2 Architecture globale

VM	Hostname	Role	IP
1	bthanu01p	Annuaire LDAP (389 DS)	192.168.74.158
2	bthbdd01p	PostgreSQL Master	192.168.74.159
3	bthbdd02p	PostgreSQL Slave	192.168.74.128
4	bthstk01p	Stockage secours	192.168.74.161

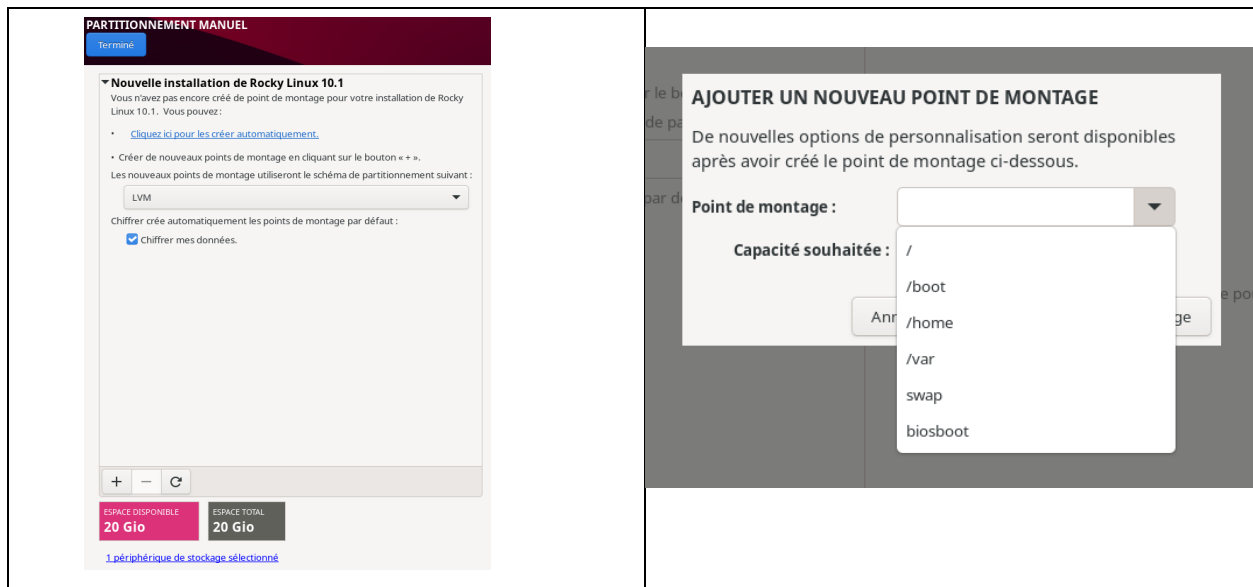
## 2. Installation bthanu01p (Annuaire LDAP)

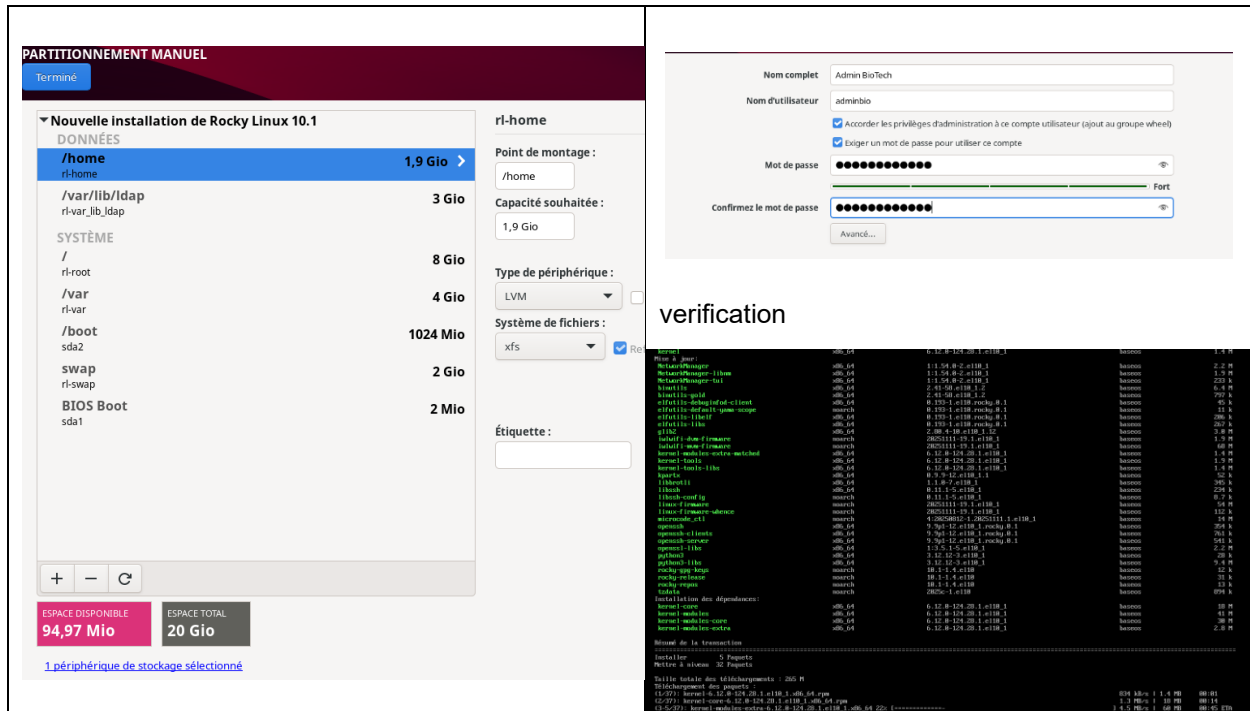
### 2.1 Installation Rocky Linux

#### Partitionnement LVM :

Point de montage	Taille	Description
/boot	1 Go	Partition de démarrage
/	8 Go	Système racine
/var	4 Go	Données variables
/var/lib/ldap	3 Go	Données LDAP (dédié)
/home	2 Go	Répertoires utilisateurs
swap	2 Go	Mémoire virtuelle
biosboot	2 Mo	Amorce BIOS

#### Screen





## 2.2 Installation 389 Directory Server

Commandes d'installation :

```
# Installation du paquet
dnf install 389-ds-base -y

# Creation du fichier de configuration
cat > /root/instance.inf << EOF
[general]
config_version = 2

[slapd]
root_password = Biotech2026!
instance_name = biotech
port = 389
secure_port = 636

[backend-userroot]
suffix = dc=biotech,dc=lan
sample_entries = yes
EOF
```

```

# Creation de l'instance
dscreate from-file /root/instance.inf

# Activation du service
systemctl enable dirsrv@biotech
systemctl start dirsrv@biotech

# Configuration du firewall
firewall-cmd --permanent --add-port=389/tcp
firewall-cmd --permanent --add-port=636/tcp
firewall-cmd --reload

```

## 2.3 Verification

```

# Test de connexion LDAP
ldapsearch -x -H ldap://localhost -b "dc=biotech,dc=lan"

# Résultat attendu : result: 0 Success

```

## Screen

### Installation de l'annuaire LDAP

```

Dernière connexion : jeudi 29 janvier 2026 à 11:03:12 CET sur pts/0
[root@bthanu01p ~]# dnf install openldap openldap-servers openldap-clients -y

```

### Création de l'instance LDAP

```

[adminbio@bthanu01p ~]$ sudo su
[sudo] Mot de passe de adminbio :
xsudo: il est nécessaire de saisir un mot de passe
[adminbio@bthanu01p ~]$ sudo su -
[sudo] Mot de passe de adminbio :
Dernière connexion : jeudi 29 janvier 2026 à 11:32:38 CET sur pts/0
[root@bthanu01p ~]# cat > /root/instance.inf << EOF
[general]
config_version = 2

[slapd]
root_password = Biotech2026!
instance_name = biotech
port = 389
secure_port = 636

[backend-userroot]
suffix = dc=biotech,dc=lan
sample_entries = yes
EOF
[root@bthanu01p ~]# dscreate from-file /root/instance.inf
Starting installation ...
Validate installation settings ...
Create file system structures ...
Create self-signed certificate database ...
Perform SELinux labeling ...
Create database backend: dc=biotech,dc=lan ...
Perform post-installation tasks ...
Completed installation for instance: slapd-biotech
[root@bthanu01p ~]#

```

```

[root@bthanu01p ~]# dscreate from-file /root/instance.inf
Starting installation ...
Validate installation settings ...
Create file system structures ...
Create self-signed certificate database ...
Perform SELinux labeling ...
Create database backend: dc=biotech,dc=lan ...
Perform post-installation tasks ...
Completed installation for instance: slapd-biotech
[root@bthanu01p ~]#

```

## Vérification du status

```
[adminbio@bthanu01p ~]$ systemctl status dirsrv@biotech
● dirsrv@biotech.service - 389 Directory Server biotech.
   Loaded: loaded (/usr/lib/systemd/system/dirsrv@.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/dirsrv@.service.d
            └─custom.conf
   Active: active (running) since Tue 2026-03-10 10:41:23 CET; 1min 35s ago
   Invocation: 1ce861e8c1d743a3b425db72fee9679e
   Process: 1115 ExecStartPre=/usr/libexec/dirsrv/ds_systemd_ask_password_acl /etc/dirsrv/slapd-biotech/dse.ldif (code=exited, status=0/SUCCESS)
   Process: 1155 ExecStartPre=/usr/libexec/dirsrv/ds_selinux_restorecon.sh /etc/dirsrv/slapd-biotech/dse.ldif (code=exited, status=0/SUCCESS)
   Main PID: 1197 (ns-slapd)
   Status: "slapd started: Ready to process requests"
     Tasks: 24 (limit: 10440)
    Memory: 87.8M (peak: 88M)
       CPU: 2.832s
   CGroup: /system.slice/system-dirsrv.slice/dirsrv@biotech.service
            └─1197 /usr/sbin/ns-slapd -D /etc/dirsrv/slapd-biotech -i /run/dirsrv/slapd-biotech.pid

mars 10 10:41:21 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:21.685121813 +0100] - INFO - ldbm_instance_config_cachememsize_set - force a minimal value 512000
mars 10 10:41:21 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:21.712811945 +0100] - INFO - dbmdb_make_env - MDB environment created with maxsize=3221225472.
mars 10 10:41:21 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:21.714382203 +0100] - INFO - dbmdb_make_env - MDB environment created with max readers=126.
mars 10 10:41:21 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:21.715816779 +0100] - INFO - dbmdb_make_env - MDB environment created with max database instances=
mars 10 10:41:22 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:22.837088633 +0100] - INFO - connection_table_new - Number of connection sub-tables 1, each conta
mars 10 10:41:22 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:22.837088633 +0100] - INFO - slapd_daemon - slapd started. Listening on all interfaces port 389
mars 10 10:41:23 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:23.881821879 +0100] - INFO - slapd_daemon - Listening on all interfaces port 636 for LDAPs request
mars 10 10:41:23 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:23.882863641 +0100] - INFO - slapd_daemon - Listening on /run/slapd-biotech.socket for LDAPi request
mars 10 10:41:25 bthanu01p ns-slapd[1197]: [10/Mar/2026:10:41:25.818585329 +0100] - INFO - vattr_check_thread - No role/cos definition in dc=biotech,dc=lan
```

## Ouverture des ports du firewall

```
[root@bthanu01p ~]# firewall-cmd --permanent --add-port=389/tcp
firewall-cmd --permanent --add-port=636/tcp
firewall-cmd --reload
success
success
success
```

## Test de la connexion LDAP

```
[root@bthanu01p ~]# ldapsearch -x -H ldap://localhost -b "dc=biotech,dc=lan"
# extended LDIF
#
# LDAPv3
# base <dc=biotech,dc=lan> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# biotech.lan
dn: dc=biotech,dc=lan
objectClass: top
objectClass: domain
dc: biotech
description: dc=biotech,dc=lan
# groups, biotech.lan
dn: ou=groups,dc=biotech,dc=lan
objectClass: top
objectClass: organizationalunit
```

```
dn: uid=demo_user,ou=people,dc=biotech,dc=lan
objectClass: top
objectClass: nsPerson
objectClass: nsAccount
objectClass: nsOrgPerson
objectClass: posixAccount
uid: demo_user
cn: Demo User
displayName: Demo User
uidNumber: 99998
gidNumber: 99998
homeDirectory: /var/empty
loginShell: /bin/false
# demo_group, Groups, biotech.lan
dn: cn=demo_group,ou=Groups,dc=biotech,dc=lan
objectClass: top
objectClass: groupOfNames
objectClass: posixGroup
objectClass: nsMemberOf
cn: demo_group
gidNumber: 99999
# search result
search: 2
result: 0 Success
# numResponses: 8
# numEntries: 7
[root@bthanu01p ~]#
```

## Récap de ce qui est configuré :

- PostgreSQL installé
- Base de données biotech créée
- Utilisateur biotech\_admin créé
- Utilisateur replicator pour la réplication
- Configuration réplication (wal\_level, max\_wal\_senders, etc.)
- Accès réseau configuré (pg\_hba.conf)

## 3. Installation bthbdd01p (PostgreSQL Master)

### 3.1 Installation Rocky Linux

Partitionnement LVM (2 disques : 20 Go + 10 Go) :

Point de montage	Taille	Description
/boot	1 Go	Partition de démarrage
/	8 Go	Système racine
/var	4 Go	Données variables
/var/lib/pgsql	10 Go	Données PostgreSQL (disque 2)
/home	2 Go	Répertoires utilisateurs
swap	2 Go	Mémoire virtuelle
biosboot	2 Mo	Amorce BIOS

### Screen

▼ Nouvelle installation de Rocky Linux 10.1

**DONNÉES**

- /home 2 Gio
- rl-home
- /var/lib/pgsql 10 Gio >**
- rl-var\_lib\_pgsql

**SYSTÈME**

- / 8 Gio
- rl-root
- /var 4 Gio
- rl-var
- /boot 1024 Mio
- sda2
- swap 2 Gio
- rl-swap
- BIOS Boot 2 Mio
- sda1

```
#1) Respectez la vie privée des autres.
#2) Réfléchissez avant d'utiliser le clavier.
#3) De grands pouvoirs confèrent de grandes responsabilités.

Pour des raisons de sécurité, le mot de passe que vous tapez ne sera pas visible.

[sudo] Mot de passe de adminbio :
[root@bthbdd01p ~]# hostname
bthbdd01p
[root@bthbdd01p ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                  8:0      0   10G  0 disk
├─sda1                               8:1      0    2M  0 part
├─sda2                               8:2      0    1G  0 part /boot
├─sda3                               8:3      0   7,5G  0 part
│   ├─rl-home                       253:2    0    2G  0 lvm  /home
│   ├─rl-var                       253:3    0    4G  0 lvm  /var
│   └─rl-var_lib_pgsql             253:4    0   10G  0 lvm  /var/lib/pgsql
sdb                                  8:16     0   20G  0 disk
├─sdb1                              8:17     0  18,5G  0 part
│   ├─rl-root                     253:0    0    8G  0 lvm  /
│   ├─rl-swap                     253:1    0    2G  0 lvm  [SWAP]
│   └─rl-var_lib_pgsql             253:4    0   10G  0 lvm  /var/lib/pgsql
sr0                                  11:0     1 1024M  0 rom
[root@bthbdd01p ~]# dnf update -y
```

```
User root = adminbio
root_password = Biotech2026!
```

## 3.2 Installation PostgreSQL

```
# Installation
dnf install postgresql-server postgresql -y

# Initialisation
postgresql-setup --initdb

# Demarrage
systemctl start postgresql
systemctl enable postgresql

# Configuration du firewall
firewall-cmd --permanent --add-port=5432/tcp
firewall-cmd -reload
```

## 3.3 Creation de la base et des utilisateurs

```
# Connexion à PostgreSQL
sudo -u postgres psql

# Creation de la base
CREATE DATABASE biotech;
CREATE USER biotech_admin WITH ENCRYPTED PASSWORD 'Biotech2026!';
GRANT ALL PRIVILEGES ON DATABASE biotech TO biotech_admin;

# Creation de l'utilisateur de replication
CREATE USER replicator WITH REPLICATION ENCRYPTED PASSWORD 'Repl2026!';
\q
```

## 3.4 Configuration pour la replication

Modifier `/var/lib/pgsql/data/postgresql.conf` :

```
listen_addresses = '*'
wal_level = replica
max_wal_senders = 3
wal_keep_size = 64MB
```

Modifier `/var/lib/pgsql/data/pg_hba.conf` (ajouter a la fin) :

```
host    replication    replicator    192.168.74.0/24    md5
host    all              all          192.168.74.0/24    md5
```

Redémarrer PostgreSQL :

```
systemctl restart postgresql
```

## 4. Installation bthbdd02p (PostgreSQL Slave)

### 4.1 Installation Rocky Linux

Partitionnement identique à bthbdd01p.

### 4.2 Configuration de la réplication

```
# Installation PostgreSQL
dnf install postgresql-server postgresql -y

# Arrêt du service et suppression des données
systemctl stop postgresql
rm -rf /var/lib/pgsql/data/*

# Copie des données depuis le Master
PGPASSWORD='Repl2026!' pg_basebackup -h 192.168.74.159 \
-U replicator -D /var/lib/pgsql/data -Fp -Xs -P -R

# Correction des permissions
chown -R postgres:postgres /var/lib/pgsql/data

# Démarrage du service
systemctl start postgresql
systemctl enable postgresql
```

### 4.3 Vérification de la réplication

```
# Sur le Slave
sudo -u postgres psql -c "SELECT * FROM pg_stat_wal_receiver;"

# Résultat attendu : status = streaming
```

#### Validation fonctionnelle de la réplication à chaud :

Afin de valider l'intégrité du flux de données entre les deux instances PostgreSQL, un

test d'écriture et de lecture croisée a été effectué : \* **Côté Master (bthbdd01p)** :

Création d'une table technique validation\_finale et insertion d'une ligne de données "Test de flux OK".

```
[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "CREATE TABLE validation_finale (id serial PRIMARY KEY, message text);"
CREATE TABLE
[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "INSERT INTO validation_finale (message) VALUES ('Test de flux OK');"
INSERT 0 1
```

- **Côté Slave (bthbdd02p)** : Vérification immédiate de la présence des données. La commande SELECT confirme que l'enregistrement a été répliqué instantanément sans erreur de cohérence.

Ce test confirme que l'architecture répond aux exigences de disponibilité et que le Slave est prêt à prendre le relais en cas de défaillance du Master (respect des contraintes PRA).

```

[adminbio@bthbdd02p ~]# sudo -u postgres psql -d biotech -c "SELECT * FROM validation_finale;"
 id |      message
-----+-----
  1 | Test de flux OK
(1 ligne)

```

## Identifiant et mot de passe

```

Root user = adminbio
root_password = Biotech2026!

```

## Screen

### Initialisation de la base

```

Total 8.3 MB/s | 9.1 MB
Test de la transaction
La vérification de la transaction a réussi.
Lancement de la transaction de test
Transaction de test réussie.
Exécution de la transaction
Préparation :
Installation : postgresql-private-libs-16.11-1.el10_1.x86_64
Installation : postgresql-16.11-1.el10_1.x86_64
Exécution du scriptlet : postgresql-server-16.11-1.el10_1.x86_64
Installation : postgresql-server-16.11-1.el10_1.x86_64
Exécution du scriptlet : postgresql-server-16.11-1.el10_1.x86_64

Installé:
postgresql-16.11-1.el10_1.x86_64 postgresql-private-libs-16.11-1.el10_1.x86_64
postgresql-server-16.11-1.el10_1.x86_64

Terminé !
[root@bthbdd01p ~]# postgresql-setup --initdb
WARNING: Note that either your data directory '/var/lib/pgsql/data' or
the parent directory '/var/lib/pgsql'
is a direct mountpoint. This is usually a bad idea and your
filesystem layout should ideally look like:
/ROOT_OWNED_MOUNTPOINT/POSTGRES_OWNED_DIRECTORY/DATADIR.
See the upstream documentation for more info:
http://www.postgresql.org/docs/16/static/creating-cluster.html
* Initializing database in '/var/lib/pgsql/data'
* Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
[root@bthbdd01p ~]#

```

### Demarre et activation du service

```

[root@bthbdd01p ~]# postgresql-setup --initdb
WARNING: Note that either your data directory '/var/lib/pgsql/data' or
the parent directory '/var/lib/pgsql'
is a direct mountpoint. This is usually a bad idea and your
filesystem layout should ideally look like:
/ROOT_OWNED_MOUNTPOINT/POSTGRES_OWNED_DIRECTORY/DATADIR.
See the upstream documentation for more info:
http://www.postgresql.org/docs/16/static/creating-cluster.html
* Initializing database in '/var/lib/pgsql/data'
* Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
[root@bthbdd01p ~]# systemctl start postgresql
systemctl enable postgresql
Created symlink '/etc/systemd/system/multi-user.target.wants/postgresql.service' → '/usr/lib/systemd/system/postgresql.service'.
[root@bthbdd01p ~]#

```

### Ouvre le port PostgreSQL

```
Dernière connexion : jeudi 29 janvier 2026 à 17:11:44 CET sur pts/2
[root@bthbdd01p ~]# firewall-cmd --permanent --add-port=5432/tcp
firewall-cmd --reload
success
success
[root@bthbdd01p ~]#
```

## On vérifie que PostgreSQL fonctionne

```
[root@bthbdd01p ~]# systemctl status postgresql
● postgresql.service - PostgreSQL database server
  Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: disabled)
  Active: active (running) since Thu 2026-01-29 16:13:00 CET; 1h 1min ago
  Invocation: 927775664c3345538855d1b83b9dfca2
  Main PID: 5942 (postgres)
  Tasks: 7 (limit: 10440)
  Memory: 19.6M (peak: 22.3M)
  CPU: 910ms
  CGroup: /system.slice/postgresql.service
          └─5942 /usr/bin/postgres -D /var/lib/pgsql/data
            └─5943 "postgres: logger "
              └─5944 "postgres: checkpointer "
                └─5945 "postgres: background writer "
                  └─5947 "postgres: walwriter "
                    └─5948 "postgres: autovacuum launcher "
                      └─5949 "postgres: logical replication launcher "
```

```
janv. 29 16:13:00 bthbdd01p systemd[1]: Starting postgresql.service - PostgreSQL database server...
janv. 29 16:13:00 bthbdd01p postgres[5942]: 2026-01-29 16:13:00.274 CET [5942] LOG: redirection des traces vers le
janv. 29 16:13:00 bthbdd01p postgres[5942]: 2026-01-29 16:13:00.274 CET [5942] ASTUCE : Les prochaines traces appar
janv. 29 16:13:00 bthbdd01p systemd[1]: Started postgresql.service - PostgreSQL database server.
lines 1-21/21 (END)
```

## On se connecte au PostgreSQL et une fois dans la base de données on crée la base de biotech

```
[root@bthbdd01p ~]# sudo -u postgres psql
psql (16.11)
Saisissez « help » pour l'aide.

postgres=#
```

```
[root@bthbdd01p ~]# sudo -u postgres psql
psql (16.11)
Saisissez « help » pour l'aide.

postgres=# CREATE DATABASE biotech;
CREATE USER biotech_admin WITH ENCRYPTED PASSWORD 'Biotech2026!';
GRANT ALL PRIVILEGES ON DATABASE biotech TO biotech_admin;
\q
CREATE DATABASE
CREATE ROLE
GRANT
[root@bthbdd01p ~]#
```

## Crée un utilisateur pour la réplication

```
[root@bthbdd01p ~]# sudo -u postgres psql
psql (16.11)
Saisissez « help » pour l'aide.

postgres=# CREATE USER replicator WITH REPLICATION ENCRYPTED PASSWORD 'Repl2026!';
\q
CREATE ROLE
[root@bthbdd01p ~]#
```

Permet aux autres serveurs (Slave, applications) de se connecter à PostgreSQL

```
#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
#port = 5432                    # (change requires restart)
max_connections = 100          # (change requires restart)
#reserved_connections = 0      # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
```

Active les journaux WAL nécessaires pour la réplication Master-Slave

```
#-----
# WRITE-AHEAD LOG
#-----

# - Settings -

wal_level = replica            # minimal, replica, or logical
                                # (change requires restart)
#fsync = on                    # flush data to disk for crash safety
                                # (turning this off can cause
                                # unrecoverable data corruption)
#synchronous_commit = on      # synchronization level;
                                # off, local, remote_write, remote_apply, or on
                                # the default is the first option
#wal_sync_method = fsync       # supported by the operating system:
                                #   open_datasync
                                #   fdatasync (default on Linux and FreeBSD)
                                #   fsync
                                #   fsync_writethrough
```

Autorise jusqu'à 3 serveurs Slave à se connecter pour répliquer les données

```

#-----
# REPLICATION
#-----
# - Sending Servers -

# Set these on the primary and on any standby that will send replication data.

max_wal_senders = 3          # max number of walsender processes
                             # (change requires restart)
#max_replication_slots = 10 # max number of replication slots
                             # (change requires restart)
#wal_keep_size = 0          # in megabytes; 0 disables
#max_slot_wal_keep_size = -1 # in megabytes; -1 disables
#wal_sender_timeout = 60s   # in milliseconds; 0 disables
#track_commit_timestamp = off # collect timestamp of transaction commit
                             # (change requires restart)

# - Primary Server -

```

Autorise le Slave à se connecter pour la réplication host all all... Autorise les connexions depuis le réseau local (applications, admin)

```

GNU nano 8.1 /var/lib/pgsql/data/pg_hba.conf
# include_if_exists FILE
# include_dir DIRECTORY
#
# FILE is the file name to include, and DIR is the directory name containing
# the file(s) to include. Any file in a directory will be loaded if suffixed
# with ".conf". The files of a directory are ordered by name.
# include_if_exists ignores missing files. FILE and DIRECTORY can be
# specified as a relative or an absolute path, and can be double-quoted if
# they contain spaces.
#
#-----
# Miscellaneous
#-----
#
# This file is read on server startup and when the server receives a
# SIGHUP signal. If you edit the file on a running system, you have to
# SIGHUP the server for the changes to take effect, run "pg_ctl reload",
# or execute "SELECT pg_reload_conf()".
#
#-----
# Put your actual configuration here
#-----
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 ident
# IPv6 local connections:
host all all ::1/128 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 ident
host replication all ::1/128 ident
host replication replicator 192.168.74.0/24 md5
host all all 192.168.74.0/24 md5

```

Et là on vérifie que la réplication fonctionne et c'est bien le cas

```

[adminbio@bthbddd02p ~]$ sudo cat /var/lib/pgsql/data/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
primary_conninfo = 'user=replicator password='Repl2026!' channel_binding=prefer host=192.168.74.159 port=5432 sslmode=prefer sslcompression=0 sslcertmode=allow
sslmode=prefer sslmin_protocol_version=TLSSLv1.2 gssencmode=prefer krbrsname=postgres gssdelegation=0 target_session_attrs=any load_balance_hosts=disable'
[adminbio@bthbddd02p ~]$ sudo systemctl start postgresql
[adminbio@bthbddd02p ~]$ sudo -u postgres psql -c "SELECT * FROM pg_stat_wal_receiver;"
 pid | status | receive_start_lsn | receive_start_tli | written_lsn | flushed_lsn | received_tli | last_msg_send_time | last_msg_receipt_time
-----+-----+-----+-----+-----+-----+-----+-----+-----
 1910 | streaming | 0/50000000 | 1 | 0/5000148 | 0/5000148 | 1 | 2026-03-10 11:57:05.883481+01 | 2026-03-10 11:57:05.88489
2+01 | 0/5000148 | 2026-03-10 11:56:35.869845+01 | 1 | 192.168.74.159 | 5432 | user=replicator password=***** channel_binding=prefer dbna
me=replication host=192.168.74.159 port=5432 fallback_application_name=walreceiver sslmode=prefer sslcompression=0 sslcertmode=allow sslsni=1 ssl_min_protocol_v
ersion=TLSSLv1.2 gssencmode=prefer krbrsname=postgres gssdelegation=0 target_session_attrs=any load_balance_hosts=disable
(1 line)

```

On peut voir que

- status = streaming → Réplication active en temps réel ✓
- sender\_host = 192.168.74.159 → Connecté au Master ✓
- sender\_port = 5432 → Port PostgreSQL ✓

## 5. Installation bthstk01p (Stockage secours)

### 5.1 Installation Rocky Linux

Partitionnement LVM (disque 40 Go) :

Point de montage	Taille	Description
/boot	1 Go	Partition de démarrage
/	8 Go	Système racine
/var	4 Go	Données variables
/backup	20 Go	Stockage des sauvegardes
/home	2 Go	Répertoires utilisateurs
swap	2 Go	Mémoire virtuelle
biosboot	2 Mo	Amorce BIOS

### 5.2 Configuration du stockage

```

# Création des répertoires
mkdir -p /backup/postgresql
mkdir -p /backup/logs
chmod 700 /backup/postgresql
chmod 700 /backup/logs

# Configuration SSH pour les transferts automatiques

```

```
# Activer PermitRootLogin yes dans /etc/ssh/sshd_config
systemctl restart sshd
```

### Identifiant et mot de passe

```
Root user = adminbio
root_password = Biotech2026!
```

## 6. Procédures de sauvegarde

### 6.1 Configuration SSH (sur bthbdd01p)

```
# Génération de la clé SSH
ssh-keygen -t rsa -b 4096 -N "" -f /root/.ssh/id_rsa

# Copie de la clé vers le serveur de stockage
ssh-copy-id root@192.168.74.161

# Test de connexion sans mot de passe
ssh root@192.168.74.161 "hostname"
```

### 6.2 Script de sauvegarde

Fichier /opt/backup\_postgresql.sh :

```
#!/bin/bash

# Variables
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/tmp/backup_pgsql"
REMOTE_HOST="192.168.74.161"
REMOTE_DIR="/backup/postgresql"
LOG_FILE="/var/log/backup_postgresql.log"

# Création du dossier local
mkdir -p $BACKUP_DIR

# Dump de la base de données
echo "[$DATE] Debut du backup" >> $LOG_FILE
sudo -u postgres pg_dumpall > $BACKUP_DIR/biotech_$DATE.sql

# Compression
gzip $BACKUP_DIR/biotech_$DATE.sql

# Transfert vers le serveur de stockage
scp $BACKUP_DIR/biotech_$DATE.sql.gz root@$REMOTE_HOST:$REMOTE_DIR/

# Nettoyage local
rm -rf $BACKUP_DIR

echo "[$DATE] Backup termine et transfère" >> $LOG_FILE
```

### 6.3 Planification automatique (cron)

```
# Rendre le script exécutable
chmod +x /opt/backup_postgresql.sh
```

```
# Ajouter au crontab (sauvegarde a 21h00)
crontab -e
# Ajouter la ligne :
0 21 * * * /opt/backup_postgresql.sh

# Verification
crontab -l
```

#### Vérification :

- Stocker les sauvegardes chiffrées
- Recevoir les dumps quotidiens à 21h00

## 7. Procédures de restauration

### 7.1 Restauration depuis le backup

```
# 1. Récupérer le backup depuis le serveur de stockage
scp root@192.168.74.161:/backup/postgresql/biotech_XXXXXXXX_XXXXXX.sql.gz /tmp/

# 2. Décompresser
gunzip /tmp/biotech_XXXXXXXX_XXXXXX.sql.gz

# 3. Restaurer
sudo -u postgres psql -f /tmp/biotech_XXXXXXXX_XXXXXX.sql

# 4. Vérifier
sudo -u postgres psql -d biotech -c "SELECT * FROM <table>;"
```

### 7.2 Basculement Master/Slave (failover)

En cas de panne du Master, promouvoir le Slave :

```
# Sur bthbdd02p (Slave)
sudo -u postgres pg_ctl promote -D /var/lib/pgsql/data

# Vérifier que le Slave est devenu Master
sudo -u postgres psql -c "SELECT pg_is_in_recovery();"
# Résultat attendu : f (false = mode Master)
```

## Screen

### Vérification

```
Mis à niveau:
NetworkManager-1:1.54.0-2.el10_1.x86_64
NetworkManager-tui-1:1.54.0-2.el10_1.x86_64
binutils-gold-2.41-58.el10_1.2.x86_64
crypto-policies-scripts-20250905-2.gitc7eb7b2.el10_1.1.noarch
elfutils-debuginfod-client-0.193-1.el10.rocky.0.1.x86_64
elfutils-libelf-0.193-1.el10.rocky.0.1.x86_64
glib2-2.80.4-10.el10_1.12.x86_64
glibc-all-langpacks-2.39-58.el10_1.7.x86_64
glibc-gconv-extra-2.39-58.el10_1.7.x86_64
iwlib-firmware-20260107-19.2.el10_1.noarch
kernel-tools-6.12.0-124.35.1.el10_1.x86_64
kpartx-0.9.9-12.el10_1.1.x86_64
libbrotli-1.1.0-7.el10_1.x86_64
libfdisk-2.40.2-15.el10_1.x86_64
libsmartcols-2.40.2-15.el10_1.x86_64
libssh-config-0.11.1-5.el10_1.noarch
linux-firmware-20260107-19.2.el10_1.noarch
microcode_ctl-4:20250812-1.20251111.1.el10_1.noarch
openssh-clients-9.9p1-12.el10_1.rocky.0.1.x86_64
openssl-libs-1:3.5.1-7.el10_1.x86_64
python3-libs-3.12.12-3.el10_1.x86_64
rocky-release-10.1-1.4.el10.noarch
tzdata-2025c-1.el10.noarch
util-linux-core-2.40.2-15.el10_1.x86_64
NetworkManag
binutils-2.4
crypto-polic
curl-8.12.1-
elfutils-def
elfutils-lib
glibc-2.39-5
glibc-common
iwlib-dm-
kernel-modul
kernel-tools
libblkid-2.4
libcurl-8.12
libmount-2.4
libssh-0.11.
libuuid-2.40
linux-firmwa
openssh-9.9p
openssh-serv
python3-3.12
rocky-gpg-ke
rocky-repos-
util-linux-2
zlib-ng-comp

Installé:
kernel-6.12.0-124.35.1.el10_1.x86_64
kernel-modules-core-6.12.0-124.35.1.el10_1.x86_64
kernel-core-6.12.0-124.35.1.el10_1.x86_64
kernel-modules-extra-6.12.0-124.35.1.el10_1.x86_64

Terminé !
[root@bthstk01p adminbiol]# df -h | grep backup
/dev/mapper/rl-backup 20G 424M 20G 3% /backup
[root@bthstk01p adminbiol]#
```

inside or press Ctrl+G.

### Activation de SSH

```
usage: 'firewall-cmd --help' for usage information or see firewall-cmd(1) man page
firewall-cmd: error: unrecognized arguments: -add-service=ssh
[root@bthstk01p adminbiol]# firewall-cmd --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[root@bthstk01p adminbiol]# firewall-cmd --reload
success
[root@bthstk01p adminbiol]#
```

### Création des dossiers backup pour la base de données

```
[root@bthstk01p adminbiol]# firewall-cmd --reload
success
[root@bthstk01p adminbiol]# mkdir -p /backup/postgresql
[root@bthstk01p adminbiol]# mkdir -p /backup/logs
[root@bthstk01p adminbiol]# chown -R postgres:postgres /backup/postgresql 2>/dev/null || chown
[root@bthstk01p adminbiol]# chmod 700 /backup/postgresql
[root@bthstk01p adminbiol]# chmod 700 /backup/logs
[root@bthstk01p adminbiol]#
```

## Activation de ssh

```
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys
"/etc/ssh/sshd_config" 130L, 3659B written
[root@bthstk01p adminbio]# systemctl restart sshd
[root@bthstk01p adminbio]# ssh-copy-id root@192.168.114.9
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any tha
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it i
root@192.168.114.9's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.114.9'"
and check to make sure that only the key(s) you wanted were added.

[root@bthstk01p adminbio]#
```

## Activation ssh et copy de la clé du master a la base de données

```
[root@bthanu01p adminbio]# ssh-keygen -t rsa -b 4096 -N "" -f /root/.ssh/id_rsa
Generating public/private rsa key pair.
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:EeHtIhy5CP+nGU3y+Ygyf9A5iS913j82+bMzSUIJduE root@bthanu01p
The key's randomart image is:
+---[RSA 4096]-----+
|      o.      |
|      oo     |
|     .oo.oo   |
|    ooo.ooE   |
|   o+Soo+o    |
|  .oo=*...    |
| o o+o+ .. o |
| o=.oo. .X   |
| o+oo. .o@   |
+---[SHA256]-----+
[root@bthanu01p adminbio]# ssh-copy-id root@192.168.74.161
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.74.161's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.74.161'"
and check to make sure that only the key(s) you wanted were added.

[root@bthanu01p adminbio]# ssh root@192.168.74.161 "hostname"
bthstk01p
```

## SCRIPT pour sauvegarder la BDD

```
GNU nano 8.1 /opt/backup_postgresql.sh
#!/bin/bash

#variables
DATE=$(date +%Y/%m/%d_%H/%M/%S)
BACKUP_DIR="/tmp/backup/pgsql"
REMOTE_HOST="192.168.74.161"
REMOTE_DIR="/backup/postgresql"
LOG_FILE="/var/log/backup_postgresql.log"

#création du dossier local
mkdir -p $BACKUP_DIR

#DUMP de la base de données
echo "[${DATE}] Début du backup" >> $LOG_FILE
sudo -u postgres pg_dumpall > $BACKUP_DIR/biotech_${DATE}.sql

#Compression
gzip $BACKUP_DIR/biotech_${DATE}.sql

#Transfert vers le serveur de stockage
scp $BACKUP_DIR/biotech_${DATE}.sql.gz root@$REMOTE_HOST:$REMOTE_DIR/

#Nettoyage local
rm -rf $BACKUP_DIR

echo "[${DATE}] Backup terminé et transféré" >> $LOG_FILE
```

## Ça fonctionne sa sauvegarde sans demander de mot de passe

```
[root@bthbdd01p adminbio]# chmod +x /opt/backup_postgresql.sh
[root@bthbdd01p adminbio]# /opt/backup_postgresql.sh
biotech_20260310_124129.sql.gz
[root@bthbdd01p adminbio]# _
```

100% 1540 265.9KB/s 00:00

```
[root@bthbdd01p adminbio]# ssh root@192.168.74.161 "hostname"
bthstk01p
[root@bthbdd01p adminbio]# /opt/backup_postgresql.sh
biotech_20260310_123302.sql.gz
[root@bthbdd01p adminbio]# _
```

100% 1537 560.6KB/s 00:00

## Vérification que le fichier est bien arrivé sur bthstk01p depuis la base de données bthbdd01p

```
[root@bthbdd01p adminbio]# ssh root@192.168.114.9 "ls -la /backup/postgresql/"
total 8
drwx-----. 2 root root 82 19 févr. 13:22 .
drwxr-xr-x. 4 root root 36 19 févr. 11:44 ..
-rw-r--r--. 1 root root 1213 19 févr. 13:07 biotech_20260219_130730.sql.gz
-rw-r--r--. 1 root root 1214 19 févr. 13:22 biotech_20260219_132248.sql.gz
[root@bthbdd01p adminbio]#
```



## Création de la table

```
[root@bthbdd01p adminbio]# grep -i "test_crash" /tmp/biotech_20260219_132248.sql
[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "CREATE TABLE test_crash (id
ERREUR: le type « varchar » n'existe pas
LIGNE 1 : CREATE TABLE test_crash (id SERIAL, nom VARCHAR(50), date_cre...

[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "CREATE TABLE test_crash (id
CREATE TABLE
[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "INSERT INTO test_crash (nom)
INSERT 0 1
[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "SELECT * FROM test_crash;"
 id |      nom
-----+-----
  1 | Donnee avant crash
(1 ligne)

[root@bthbdd01p adminbio]#
```

## Test crash

```
SET
SET
psql:/tmp/biotech_20260219_135657.sql:100: ERREUR: la base de données « biotech » existe déjà
ALTER DATABASE
Vous êtes maintenant connecté à la base de données « biotech » en tant qu'utilisateur « postgres ».
SET
SET
SET
SET
SET
SET
set_config
-----
(1 ligne)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
ALTER TABLE
COPY 1
setval
-----
  1
(1 ligne)

GRANT
Vous êtes maintenant connecté à la base de données « postgres » en tant qu'utilisateur « postgres ».
SET
SET
SET
SET
SET
set_config
-----
(1 ligne)

SET
SET
SET
SET
[root@bthbdd01p adminbio]#
```

```

(1 ligne)

SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
ALTER TABLE
COPY 1
  setval
-----
      1
(1 ligne)

GRANT
Vous êtes maintenant connecté à la base de données « postgres » en tant qu'utilisateur « pos
SET
SET
SET
SET
SET
SET
  set_config
-----

(1 ligne)

SET
SET
SET
SET
[root@bthbdd01p adminbio]# sudo -u postgres psql -d biotech -c "SELECT * FROM test_crash;"
 id |      nom
-----+-----
  1 | Donnee avant crash
(1 ligne)

[root@bthbdd01p adminbio]#

```

## 8. Informations de connexion

### 8.1 Comptes système

VM	Utilisateur	Mot de passe	Acces
Toutes	adminbio	Biotech2026!	SSH + sudo
Toutes	root	Biotech2026!	Console uniquement

### 8.2 Comptes applicatifs

Service	Utilisateur	Mot de passe
LDAP (389 DS)	cn=Directory Manager	Biotech2026!
PostgreSQL	biotech_admin	Biotech2026!
PostgreSQL (réplication)	replicator	Repl2026!

## 9. Contraintes PRA

Indicateur	Valeur	Description
PDMA	4 heures	Perte de Données Maximale Admissible
DIMA	4 heures	Durée d'Interruption Maximale Admissible

La sauvegarde quotidienne a 21h00 et la réplication Master/Slave permettent de respecter ces contraintes.

## 10. Contacts

Role	Nom	Email
Expert Données/Sauvegarde	[Votre nom]	[Votre email]
Responsable projet	[Nom]	[Email]