1. Install Dependencies

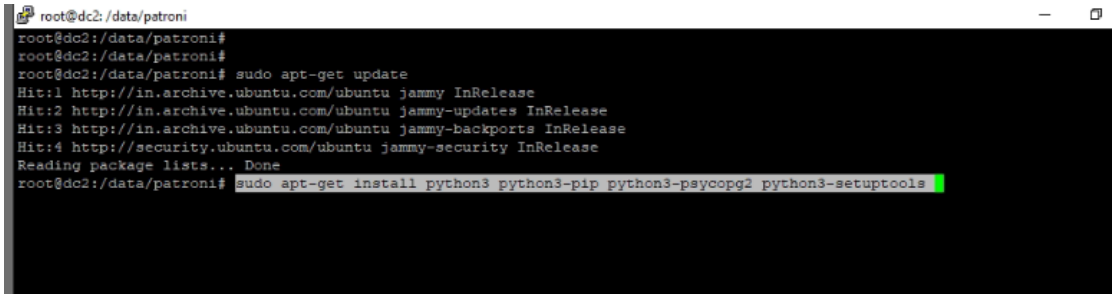
On Ubuntu:

sudo apt-get update

sudo apt-get install python3 python3-pip python3-psycopg2 python3-setuptools

On CentOS/RHEL:

sudo yum install python3 python3-pip python3-psycopg2 python3-setuptools git gcc



2. Download and Install Barman from Source Code:-

Website :- https://github.com/EnterpriseDB/barman/releases

git clone https://github.com/EnterpriseDB/barman.git

Install Barman: Install Barman using pip from the source:

chown postgres:postgres /barman ----- where barman source code file install

Create the Barman User:- (optional)

sudo useradd -m -d /var/lib/barman -s /bin/bash barman
sudo mkdir /var/log/barman
sudo chown barman:barman /var/log/barman

```
root@dc2: /barman/data
root@dc2:/barman/data#
root@dc2:/barman/data#
root@dc2:/barman/data# chown postgres:postgres /barman/ -R
root@dc2:/barman/data# cd data
-bash: cd: data: No such file or directory
root@dc2:/barman/data# ll
total 212
drwxr-xr-x 10 postgres postgres  4096 Oct  6 20:50 ./
drwxr-xr-x  3 postgres postgres  4096 Oct  6 20:50 ../
drwxr-xr-x  3 postgres postgres  4096 Oct  6 20:50 actions/
-rw-r--r--  1 postgres postgres  1415 Oct  6 20:50 AUTHORS
drwxr-xr-x  5 postgres postgres  4096 Oct  6 20:50 barman/
drwxr-xr-x  9 postgres postgres  4096 Oct  6 20:50 doc/
drwxr-xr-x  8 postgres postgres  4096 Oct  6 20:50 .git/
drwxr-xr-x  3 postgres postgres  4096 Oct  6 20:50 .github/
-rw-r--r--  1 postgres postgres   313 Oct  6 20:50 .gitignore
-rw-r--r--  1 postgres postgres   184 Oct  6 20:50 .gitleaks.toml
-rw-r--r--  1 postgres postgres    25 Oct  6 20:50 .hadolint.yaml
-rw-r--r--  1 postgres postgres   271 Oct  6 20:50 INSTALL.md
-rw-r--r--  1 postgres postgres    49 Oct  6 20:50 .isort.cfg
-rw-r--r--  1 postgres postgres 35149 Oct  6 20:50 LICENSE
-rw-r--r--  1 postgres postgres   196 Oct  6 20:50 MANIFEST.in
-rw-r--r--  1 postgres postgres   495 Oct  6 20:50 .markdownlint.yml
-rw-r--r--  1 postgres postgres 69403 Oct  6 20:50 NEWS
-rw-r--r--  1 postgres postgres    30 Oct  6 20:50 .python-black
-rw-r--r--  1 postgres postgres  2193 Oct  6 20:50 README.rst
-rw-r--r--  1 postgres postgres    15 Oct  6 20:50 requirements-tox.txt
drwxr-xr-x  2 postgres postgres  4096 Oct  6 20:50 scripts/
-rw-r--r--  1 postgres postgres   244 Oct  6 20:50 setup.cfg
-rwxr-xr-x  1 postgres postgres  5199 Oct  6 20:50 setup.py*
drwxr-xr-x  3 postgres postgres  4096 Oct  6 20:50 sphinx/
drwxr-xr-x  3 postgres postgres  4096 Oct  6 20:50 tests/
-rw-r--r--  1 postgres postgres   451 Oct  6 20:50 TODO
-rw-r--r--  1 postgres postgres   878 Oct  6 20:50 tox.ini
-rw-r--r--  1 postgres postgres   246 Oct  6 20:50 .yamllint.yml
root@dc2:/barman/data#
```

sudo pip3 install .

```
root@dc2:/barman/data# sudo pip3 install .
root@dc2:/barman/data# sudo pip3 install .
Processing /barman/data
  Preparing metadata (setup.py) ... done
Requirement already satisfied: psycopg2>=2.4.2 in /usr/local/lib/python3.10/dist-packages (from barman==3.11.1) (2.9.9)
Requirement already satisfied: python-dateutil in /usr/lib/python3/dist-packages (from barman==3.11.1) (2.8.1)
Building wheels for collected packages: barman
  Building wheel for barman (setup.py) ... done
  Created wheel for barman: filename=barman-3.11.1-py2.py3-none-any.whl size=427596 sha256=712fc6e7d8d4509ab3c57f3a3365d113
f13107f3b9472b6efe0521a805fb7e
  Stored in directory: /tmp/pip-ephem-wheel-cache-64ubga06/wheels/b1/a6/d1/1ef0aade7f0a0eede232aa9cdd63824e7df92f03cd257a7a
Successfully built barman
Installing collected packages: barman
Successfully installed barman-3.11.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package
nager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@dc2:/barman/data# chown root:root /barman/ -R
```

```
Check Barman's documentation for more help.
root@dc2:/usr/bin# which barman
/usr/local/bin/barman
root@dc2:/usr/bin#
root@dc2:/usr/bin#
root@dc2:/usr/bin#
root@dc2:/usr/bin#
```

Create the Barman configuration file:-

Vi /etc/barman.conf

```
  GNU nano 6.2                                        /etc/barman.conf *
[barman]
barman_user = postgres   #add user where you want to excute barman command
barman_home = /home/postgres
#configuration_files_directory = /etc/barman.conf
log_file = /var/log/barman/barman.log
compression = gzip
log_level = INFO
immediate_checkpoint=true
basebackup_retry_t times = 3
basebackup_retry_sleep = 30
last_backup_maximum_age = 5 DAYS
#path_prefix=/usr/lib/pgsql-15.6/bin

[pg_server]
description = "PostgreSQL Server Backup"
conninfo = host=192.168.29.83 user=postgres dbname=postgres
#ssh_command = ssh postgres@10.83.40.101
archiver=on
backup_method = postgres
streaming_archiver=on
backup_directory = /data/barman/backups
#backup_options = concurrent_backup
retention_policy = RECOVERY WINDOW OF 7 DAYS
wal_retention_policy = main
retention_policy_mode = auto
minimum_redundancy=2
```

```
logout
root@dc2:/usr/bin# nano /etc/barman.conf
root@dc2:/usr/bin# chown postgres:postgres /etc/barman.conf
root@dc2:/usr/bin#
```

[barman]
Barman_user = postgres
barman _home = /home/postgres
#configuration_files_directory = /etc/barman.conf
log_file = /var/log/barman/barman.log
compression = gzip
log_level = INFO
immediate_checkpoint=true
basebackup_retry_times = 3
basebackup_retry_sleep = 30
last_backup_maximum_age = 5 DAYS
#path_prefix=/usr/lib/pgsql-15.6/bin

[pg_server]
description = "PostgreSQL Server Backup"
conninfo = host=192.168.29.83 user=postgres dbname=postgres
#ssh command = ssh postgres@10.83.40.101
archiver=on
backup_method = postgres
streaming_archiver=on
backup_directory = /data/barman/backups
#backup_options = concurrent_backup
retention_policy = RECOVERY WINDOW OF 7 DAYS
wal_retention_policy = main
retention_policy_mode = auto
minimum_redundancy=2

(OR)

1. First makdir barman.d in /etc and give postgres premission:-

mkdir /etc/barman.d

Chown postgres:postgres /etc/barman.d

2. Inside barman.d directory create server config :-

Vi pg_server:-

```
description = "PostgreSQL Server Backup"
conninfo = host=192.168.29.83 user=postgres dbname=postgres
#ssh command = ssh postgres@10.83.40.101
archiver=on
backup_method = postgres
streaming_archiver=on
backup_directory = /data/barman/backups
#backup_options = concurrent_backup
retention_policy = RECOVERY WINDOW OF 7 DAYS
wal_retention_policy = main
retention_policy_mode = auto
minimum_redundancy=2
#path_prefix=/usr/lib/pgsql-15.6/bin
```

3. Create /ect/barman.conf:-

Vi /etc/barman.conf

```
Barman_user = postgres
barman _home = /home/postgres
configuration_files_directory = /etc/barman.d
log_file = /var/log/barman/barman.log
compression = gzip
log_level = INFO
immediate_checkpoint=true
basebackup_retry_times = 3
basebackup_retry_sleep = 30
last_backup_maximum_age = 5 DAYS
```

IMPORTANT: Barman uses external tools to manage compressed backups. Depending on the `backup_compression` and `backup_compression_format` You may need to install one or more tools on the Postgres server and the Barman server. The following table will help you choose according to your configuration.

backup_compression	backup_compression_format	Postgres server	Barman server
gzip	plain	**tar**	None
gzip	tar	**tar**	**tar**
lz4	plain	**tar, lz4**	None
lz4	tar	**tar, lz4**	**tar, lz4**
zstd	plain	**tar, zstd**	None
zstd	tar	**tar, zstd**	**tar, zstd**
none	tar	**tar**	**tar**

Note: If you are using the Barman user instead of the Postgres user in the conninfo parameter, follow the steps below:

CREATE USER barman WITH REPLICATION PASSWORD 'barman@123';

GRANT EXECUTE ON FUNCTION pg_backup_start(text, boolean) to barman;
GRANT EXECUTE ON FUNCTION pg_backup_stop(boolean) to barman;
GRANT EXECUTE ON FUNCTION pg_switch_wal() to barman;
GRANT EXECUTE ON FUNCTION pg_create_restore_point(text) to barman;

GRANT pg_read_all_settings TO barman;
GRANT pg_read_all_stats TO barman;

GRANT pg_checkpoint TO barman;

######command show where we have restore wal file:-

barman show-server server-name(pg-server) | grep incoming_wals_directory

3. PostgreSQL Configuration:-

Update pg_hba.conf **to allow Barman access**:

`#ip4`
host all all 192.168.29.83/32 trust

`#replication privilege`
host replication all 192.168.29.83/32 trust

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
host    all             all             192.168.29.83/32            trust
# IPv6 local connections:
host    all             all             ::1/128                 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     trust
host    replication     all             127.0.0.1/32            trust
host    replication     all             ::1/128                 trust
host    replication     all             192.168.29.83/32                trust
```

Enable WAL archiving:

sudo nano /var/lib/pgsql/14/data/postgresql.conf

wal_level = replica
archive_mode = on
archive_command = 'rsync -a %p
barman@10.83.40.101:/var/lib/barman/postgresql/incoming/%f'

Archive command= cp %p/var/lib/barman/postgresql/incoming/%f

max_wal_senders = 3
wal_keep_size = 128MB

To referesh braman:-

barman cron **----------to start wal archiving**

```
root@dc2: /usr/local/lib
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$ barman cron
Starting WAL archiving for server pg_server
Starting streaming archiver for server pg_server
postgres@dc2:/data/patroni$
```

4. Test Barman:-

Check Barman configuration:

barman check pg_server

```
root@dc2: /usr/local/lib
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$ barman cron
Starting WAL archiving for server pg_server
Starting streaming archiver for server pg_server
postgres@dc2:/data/patroni$ barman check pg_server
Server pg_server:
        PostgreSQL: OK
        superuser or standard user with backup privileges: OK
        PostgreSQL streaming: OK
        wal_level: OK
        directories: OK
        retention policy settings: OK
        backup maximum age: FAILED (interval provided: 5 days, latest backup age: No available backups)
        backup minimum size: OK (0 B)
        wal maximum age: OK (no last_wal_maximum_age provided)
        wal size: OK (0 B)
        compression settings: OK
        failed backups: OK (there are 0 failed backups)
        minimum redundancy requirements: FAILED (have 0 backups, expected at least 2)
        pg_basebackup: OK
        pg_basebackup compatible: OK
        pg_basebackup supports tablespaces mapping: OK
        systemid coherence: OK (no system Id stored on disk)
        pg_receivexlog: OK
        pg_receivexlog compatible: OK
        receive-wal running: OK
        archive_mode: OK
        archive_command: OK
        continuous archiving: OK
        archiver errors: OK
postgres@dc2:/data/patroni$
```

Perform a backup: Trigger a backup using:

barman backup pg_server

```
root@dc2: /usr/local/lib
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$ barman backup pg_server
Starting backup using postgres method for server pg_server in /barman/backups/base/20241006T215052
Backup start at LSN: 0/20000D8 (000000010000000000000002, 000000D8)
Starting backup copy via pg_basebackup for 20241006T215052
Copy done (time: 3 seconds)
Finalising the backup.
This is the first backup for server pg_server
WAL segments preceding the current backup have been found:
        000000010000000000000001 from server pg_server has been removed
Backup size: 21.5 MiB
Backup end at LSN: 0/4000000 (000000010000000000000003, 00000000)
Backup completed (start time: 2024-10-06 21:50:52.752085, elapsed time: 3 seconds)
Processing xlog segments from streaming for pg_server
        000000010000000000000002
        000000010000000000000003
Processing xlog segments from file archival for pg_server
        000000010000000000000002
        000000010000000000000003
        000000010000000000000003.00000028.backup
postgres@dc2:/data/patroni$
```

List backups: Check available backups with:

barman list-backup pg_server

```
root@dc2: /usr/local/lib
postgres@dc2:/data/patroni$ barman list-backup pg_server
pg_server 20241006T215052 - F - Sun Oct  6 21:50:55 2024 - Size: 21.5 MiB - WAL Size: 0 B
postgres@dc2:/data/patroni$
```

Restore a backup (if needed): Recover the backup with:

barman recover <server_name> <backup_id> <restore_destination_directory>

barman recover pg_server latest /data/patroni

```
root@dc2: /usr/local/lib
postgres@dc2:/data/patroni$ barman recover pg_server  latest /data/patroni
Starting local restore for server pg_server using backup 20241006T215052
Destination directory: /data/patroni
Copying the base backup.
Copying required WAL segments.
Generating archive status files
Identify dangerous settings in destination directory.

IMPORTANT
These settings have been modified to prevent data losses

postgresql.conf line 256: archive_command = false

Recovery completed (start time: 2024-10-06 21:55:51.404845+05:30, elapsed time: 2 seconds)
Your PostgreSQL server has been successfully prepared for recovery!
postgres@dc2:/data/patroni$
```

Crontab:-

0 0 * * * /usr/local/bin/barman backup postgresql

If you set password for user:-

```
root@dc2: /usr/local/lib
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$
postgres@dc2:/data/patroni$ barman check pg_server
Server pg_server:
        PostgreSQL: FAILED
        directories: OK
        retention policy settings: OK
        backup maximum age: OK (interval provided: 5 days, latest backup age: 19 minutes, 59 seconds)
        backup minimum size: OK (21.5 MiB)
        wal maximum age: OK (no last_wal_maximum_age provided)
        wal size: OK (0 B)
        compression settings: OK
        failed backups: OK (there are 0 failed backups)
        minimum redundancy requirements: FAILED (have 1 backups, expected at least 2)
        pg_basebackup: OK
        pg_basebackup compatible: OK
        systemid coherence: OK
        pg_receivexlog: OK
        pg_receivexlog compatible: OK
        receive-wal running: FAILED (See the Barman log file for more details)
        archiver errors: OK
postgres@dc2:/data/patroni$ nano /etc/barman.conf
```

```
root@dc2: ~
  GNU nano 6.2                                  /etc/barman.conf
[barman]
barman_user = postgres
barman_home = /home/postgres
#configuration_files_directory = /etc/barman.conf
log_file = /var/log/barman/barman.log
compression = gzip
log_level = INFO
immediate_checkpoint=true
basebackup_retry_times = 3
basebackup_retry_sleep = 30
last_backup_maximum_age = 5 DAYS
#path_prefix="/usr/lib/pgsql-15.6/bin

[pg_server]
description = "PostgreSQL Server Backup"
conninfo = host=192.168.29.83 user=postgres dbname=postgres password=123
#ssh_command = ssh postgres@10.83.40.101
archiver=on
backup_method = postgres
streaming_archiver=on
backup_directory = /barman/backups
#backup_options = concurrent_backup
retention_policy = RECOVERY WINDOW OF 7 DAYS
wal_retention_policy = main
retention_policy_mode = auto
minimum_redundancy=2
```

Increment backup:-

Postgres & barman install on same server than used below conf file

Nano /etc/barman.conf:-

```
[barman]
barman_user = postgres
#barman_home = /var/lib/barman
barman_home = /home/postgres
#configuration_files_directory = /etc/barman.conf
log_file = /var/log/barman/barman.log
compression = gzip
log_level = INFO
immediate_checkpoint = true
basebackup_retry_times = 3
basebackup_retry_sleep = 30
last_backup_maximum_age = 5 DAYS
#path_prefix = /usr/lib/pgsql-15.6/bin/:/usr/local/bin:/usr/bin
#path_prefix = /usr/lib/pgsql-15.6/bin
```

```
[pg_server_2]
description = "PostgreSQL Local Server Backup"
conninfo = host=10.83.40.101 user=postgres dbname=postgres
#ssh_command = ssh  postgres@10.83.40.101 -q
reuse_backup = link
archiver = on
backup_method = local-rsync
streaming_archiver = on
parallel_jobs = 2
backup_directory = /data/barman/backups
backup_options = concurrent_backup
retention_policy = RECOVERY WINDOW OF 7 DAYS
wal_retention_policy = main
retention_policy_mode = auto
path_prefix = /usr/lib/pgsql-15.6/bin/
#minimum_redundancy=2
```

For full backup:-

off: standard full backup (default)

```
barman backup --reuse-backup=off pg_server
```

For incremental backup:-

link:- file-level incremental backup, by reusing the last backup for a server and creating a hard link of the unchanged files (for backup space and time reduction)

barman backup --reuse-backup=link pg_server

copy:- file-level incremental backup, by reusing the last backup for a server and creating a copy of

the unchanged files (just for backup time reduction)

barman backup --reuse-backup=copy pg_server

To restore these backup:-

barman recover <server_name> <backup_id> /path/to/recover/dir

barman recover pg_server 20241015T053502 /data/test

COMMAND

1. Cron:-

barman cron

The cron command ensures that WAL streaming is started for those servers that have requested it, by transparently executing the receive-wal command.

In order to stop the operations started by the cron command, comment out the cron entry and execute:

barman receive-wal --stop SERVER_NAME

2.list-servers

You can display the list of active servers that have been configured for your backup system with:
barman list-servers

A machine readable output can be obtained with the --minimal option:
barman list-servers --minimal

3.check
barman check <server_name>

4.list-backups

barman list-backups <server_name>

5.verify-backup

barman verify-backup <server_name> <backup_id>

6.Backup ID shortcuts:-

Barman allows you to use special keywords to identify a specific backup:
• last/latest: identifies the newest backup in the catalog
• first/oldest: identifies the oldest backup in the catalog
• last-failed: identifies the newest failed backup in the catalog

barman delete <server_name> oldest

7.check-backup:-

barman check-backup <server_name> <backup_id>

Summary of Key Directories:

- `identity.json` : Contains server identity and metadata.
- `errors` : Stores error logs for backup operations.
- `wals` : Stores archived WAL files for point-in-time recovery.
- `base` : Stores full (base) backups of the PostgreSQL database.
- `streaming` : Stores real-time streamed WAL files.
- `incoming` : Temporarily stores WAL files before processing.

These directories help Barman manage backups, WAL archiving, and provide mech
the PostgreSQL database to a specific point in time.