

## Barman

### Instalacja Debian Linux

```

echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list
apt-get update
apt-get upgrade
apt-get install barman

mkdir /etc/barman.d
chown barman /etc/barman.d

mkdir /mnt/backup/barman
chown barman:barman /mnt/backup/barman

# uzytkownik barman musi miec mozliwosc logowania sie przez ssh na uzytkownika postgres bez hasla: ssh postgres@127.0.0.1
su -c "ssh-keygen" barman
cat /var/lib/barman/.ssh/id_rsa.pub >> /var/lib/postgresql/.ssh/authorized_keys
chown postgres:postgres /var/lib/postgresql/.ssh/authorized_keys

su -c "ssh-keygen" postgres
cat /var/lib/postgresql/.ssh/id_rsa.pub >> /var/lib/barman/.ssh/authorized_keys
chown barman:barman /var/lib/barman/.ssh/authorized_keys

# sprawdzamy i akceptujemy certyfikat
su - barman
psql -Upostgres -h localhos -c "SELECT version();"
ssh postgres@127.0.0.1
ssh barman@127.0.0.1
exit

vim /etc/barman.conf
vim /etc/barman.d/edokumenty.conf

```

### Konfiguracja

/etc/barman.conf

```

[barman]
; Main directory
barman_home = /mnt/backup/barman

; System user
barman_user = barman

; Log location
log_file = /var/log/barman/barman.log

; Default compression level: possible values are None (default), bzip2, gzip or custom
compression = gzip

; Incremental backup support: possible values are None (default), link or copy
reuse_backup = link

; Pre/post backup hook scripts
;pre_backup_script = env | grep ^BARMAN
;post_backup_script = env | grep ^BARMAN

; Pre/post archive hook scripts

```

```

;pre_archive_script = env | grep ^BARMAN
;post_archive_script = env | grep ^BARMAN

; Directory of configuration files. Place your sections in separate files with .conf extension
; For example place the 'main' server section in /etc/barman.d/main.conf
configuration_files_directory = /etc/barman.d

; Minimum number of required backups (redundancy) - default 0
minimum_redundancy = 1

; Global retention policy (REDUNDANCY or RECOVERY WINDOW) - default empty
retention_policy = RECOVERY WINDOW OF 2 WEEKS

; Global bandwidth limit in KBPS - default 0 (meaning no limit)
;bandwidth_limit = 4000

; Immediate checkpoint for backup command - default false
;immediate_checkpoint = false

; Enable network compression for data transfers - default false
;network_compression = false

; Identify the standard behavior for backup operations: possible values are
; exclusive_backup (default), concurrent_backup
;backup_options = exclusive_backup

; Number of retries of data copy during base backup after an error - default 0
basebackup_retry_times = 3

; Number of seconds of wait after a failed copy, before retrying - default 30
basebackup_retry_sleep = 40

; Time frame that must contain the latest backup date.
; If the latest backup is older than the time frame, barman check
; command will report an error to the user.
; If empty, the latest backup is always considered valid.
; Syntax for this option is: "i (DAYS | WEEKS | MONTHS)" where i is an
; integer > 0 which identifies the number of days | weeks | months of
; validity of the latest backup for this check. Also known as 'smelly backup'.
last_backup_maximum_age = 2 WEEKS

```

/etc/barman.d/edokumenty.conf

```

[edokumenty]
;;Human readable description
description = "eDokumenty PostgreSQL Database"

;;
;; ; SSH options
ssh_command = ssh postgres@127.0.0.1

;;
;; ; PostgreSQL connection string
conninfo = host=localhost user=postgres

```