

Advanced Backup/Restore with pgmoneta

Contents

1	Preface	3
2	Introduction	4
2.1	Features	4
2.2	Platforms	5
3	Installation	6
3.1	Rocky Linux 9.x	6
3.2	PostgreSQL 17	6
3.3	pgmoneta	8
4	Configuration	10
4.1	pgmoneta.conf	10
4.1.1	pgmoneta	10
4.1.2	Server section	23
4.2	pgmoneta_users.conf	27
4.3	pgmoneta_admins.conf	27
5	Backup	28
5.1	Create a full backup	28
5.2	View backups	28
5.3	Sorting backups	29
5.4	Create an incremental backup	29
5.5	View backups	30
5.6	Backup information	31
5.7	Create a crontab	32
5.8	Verify backup integrity	32
6	Retention policy	33
6.1	Retention configuration	33
6.2	Defining retention	33
6.3	Retention check	33
7	Keeping backups	35
7.1	List backups	35
7.2	Keep a backup	35
7.3	Describe a backup	36
7.4	Put a backup back into retention	38

7.5	Cascade mode	38
8	Restore	40
8.1	Restore a backup	40
9	Prometheus metrics	42
9.1	Access Prometheus metrics	42
10	Administration access	43
10.1	Configuration	43
10.2	Administrators	43
10.3	Restart pgmoneta	43
10.4	Connect to pgmoneta	43
11	Shutdown	44
11.1	ping	44
11.2	mode	44
11.3	shutdown	44
12	Running pgmoneta with Docker	45
12.1	Prerequisites	45
12.2	Update the configuration file if needed:	45
12.3	Step 1: Enable External PostgreSQL Access	45
12.4	Step 2: Clone the Repository	46
12.5	Step 3: Build the Docker Image	46
12.6	Step 4: Run pgmoneta as a Docker Container	46
12.7	Step 5: Verify the Container	47
13	Troubleshooting	48
13.1	Could not get version for server	48
14	Acknowledgement	49
14.1	Authors	49
14.2	Committers	49
14.3	Contributing	50
15	License	51
15.1	libart	51

1 Preface

Acme Boot is a startup company that have decided to use **PostgreSQL** as its database technology.

The following technologies will be used for the database cluster

- **Rocky Linux 9.x**
- **PostgreSQL 17.x**
- **pgmoneta**

Note, that this guide will focus on the **pgmoneta** aspect of the platform.

2 Introduction

pgmoneta is a backup / restore solution for PostgreSQL.

Ideally, you would not need to do backups and disaster recovery, but that isn't how the real World works.

Possible scenarios that could happen

- Data corruption
- System failure
- Human error
- Natural disaster

and then it is up to the database administrator to get the database system back on-line, and to the correct recovery point.

Two key factors are

- Recovery Point Objective (RPO): Maximum targeted period in which data might be lost from an IT service due to a major incident
- Recovery Time Objective (RTO): The targeted duration of time and a service level within which a business process must be restored after a disaster (or disruption) in order to avoid unacceptable consequences associated with a break in business continuity

You would like to have both of these as close to zero as possible, since RPO of 0 means that you won't lose data, and RTO of 0 means that your system recovers at once. However, that is easier said than done.

pgmoneta is focused on having features that will allow database systems to get as close to these goals as possible such that high availability of 99.99% or more can be implemented, and monitored through standard tools.

pgmoneta is named after the Roman Goddess of Memory.

2.1 Features

- Full backup
- Restore
- Compression (gzip, zstd, lz4, bzip2)
- AES encryption support
- Symlink support
- WAL shipping support

- Hot standby
- Prometheus support
- Remote management
- Offline detection
- Transport Layer Security (TLS) v1.2+ support
- Daemon mode
- User vault

2.2 Platforms

The supported platforms are

- Fedora 39+
- RHEL 9
- RockyLinux 9
- FreeBSD
- OpenBSD

3 Installation

3.1 Rocky Linux 9.x

We can download the Rocky Linux distro from their web site

```
https://rockylinux.org/download
```

The installation and setup is beyond the scope of this guide.

Ideally, you would use dedicated user accounts to run **PostgreSQL** and **pgmoneta**

```
useradd postgres
usermod -a -G wheel postgres
useradd pgmoneta
usermod -a -G wheel pgmoneta
```

Add a configuration directory for **pgmoneta**

```
mkdir /etc/pgmoneta
chown -R pgmoneta:pgmoneta /etc/pgmoneta
```

and let's open the ports in the firewall that we will need

```
firewall-cmd --permanent --zone=public --add-port=5001/tcp
firewall-cmd --permanent --zone=public --add-port=5002/tcp
```

3.2 PostgreSQL 17

We will install PostgreSQL 17 from the official YUM repository with the community binaries,

x86_64

```
dnf -qy module disable postgresql
dnf install -y https://download.postgresql.org/pub/repos/yum/repos/EL
-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

aarch64

```
dnf -qy module disable postgresql
dnf install -y https://download.postgresql.org/pub/repos/yum/repos/EL
-9-aarch64/pgdg-redhat-repo-latest.noarch.rpm
```

and do the install via

```
dnf install -y postgresql17 postgresql17-server postgresql17-contrib
```

First, we will update `~/.bashrc` with

```
cat >> ~/.bashrc
export PGHOST=/tmp
export PATH=/usr/pgsql-17/bin/:$PATH
```

then Ctrl-d to save, and

```
source ~/.bashrc
```

to reload the Bash environment.

Then we can do the PostgreSQL initialization

```
mkdir DB
initdb -k DB
```

and update configuration - for a 8 GB memory machine.

postgresql.conf

```
listen_addresses = '*'
port = 5432
max_connections = 100
unix_socket_directories = '/tmp'
password_encryption = scram-sha-256
shared_buffers = 2GB
huge_pages = try
max_prepared_transactions = 100
work_mem = 16MB
dynamic_shared_memory_type = posix
wal_level = replica
wal_log_hints = on
max_wal_size = 16GB
min_wal_size = 2GB
log_destination = 'stderr'
logging_collector = on
log_directory = 'log'
log_filename = 'postgresql.log'
log_rotation_age = 0
log_rotation_size = 0
log_truncate_on_rotation = on
log_line_prefix = '%p [%m] [%x] '
log_timezone = UTC
datestyle = 'iso, mdy'
timezone = UTC
lc_messages = 'en_US.UTF-8'
lc_monetary = 'en_US.UTF-8'
lc_numeric = 'en_US.UTF-8'
lc_time = 'en_US.UTF-8'
```

pg_hba.conf

local	all	all		trust
host	postgres	repl	127.0.0.1/32	scram-sha-256
host	postgres	repl	:::1/128	scram-sha-256
host	replication	repl	127.0.0.1/32	scram-sha-256
host	replication	repl	:::1/128	scram-sha-256

Please, check with other sources in order to create a setup for your local setup.

Now, we are ready to start PostgreSQL

```
pg_ctl -D DB -l /tmp/ start
```

Lets connect, add the replication user, and create the Write-Ahead Log (WAL) slot that we need for **pgmoneta**

```
psql postgres
CREATE ROLE repl WITH LOGIN REPLICATION PASSWORD 'repl';
SELECT pg_create_physical_replication_slot('repl', true, false);
\q
```

3.3 pgmoneta

We will install **pgmoneta** from the official YUM repository as well,

```
dnf install -y pgmoneta
```

First, we will need to create a master security key for the **pgmoneta** installation, by

```
pgmoneta-admin -g master-key
```

By default, this will ask for a key interactively. Alternatively, a key can be provided using either the `--password` command line argument, or the `PGMONETA_PASSWORD` environment variable. Note that passing the key using the command line might not be secure.

Then we will create the configuration for **pgmoneta**,

```
cat > /etc/pgmoneta/pgmoneta.conf
[pgmoneta]
host = *
metrics = 5001

base_dir = /home/pgmoneta/backup

compression = zstd

retention = 7
```

Advanced Backup/Restore with pgmoneta

```
log_type = file
log_level = info
log_path = /tmp/pgmoneta.log

unix_socket_dir = /tmp/

[primary]
host = localhost
port = 5432
user = repl
wal_slot = repl
```

and end with a Ctrl-d to save the file.

Then, we will create the user configuration,

```
pgmoneta-admin -f /etc/pgmoneta/pgmoneta_users.conf -U repl -P repl user
add
```

Lets create the base directory, and start **pgmoneta** now, by

```
mkdir backup
pgmoneta -d
```

4 Configuration

4.1 pgmoneta.conf

The configuration is loaded from either the path specified by the `-c` flag or `/etc/pgmoneta/pgmoneta.conf`.

The configuration of `pgmoneta` is split into sections using the `[` and `]` characters.

The main section, called `[pgmoneta]`, is where you configure the overall properties of `pgmoneta`.

Other sections doesn't have any requirements to their naming so you can give them meaningful names like `[primary]` for the primary PostgreSQL instance.

All properties are in the format `key = value`.

The characters `#` and `;` can be used for comments; must be the first character on the line. The `Bool` data type supports the following values: `on`, `yes`, `1`, `true`, `off`, `no`, `0` and `false`.

See a sample configuration for running `pgmoneta` on `localhost`.

Note, that PostgreSQL 13+ is required, as well as having `wal_level` at `replica` or `logical` level.

4.1.1 pgmoneta

4.1.1.1 General

Property	Default	Unit	Required	Description
host		String	Yes	The bind address for pgmoneta
unix_socket_dir		String	Yes	The Unix Domain Socket location
base_dir		String	Yes	The base directory for the backup

Note, that if `host` starts with a `/` it represents a path and `pgmoneta` will connect using a Unix Domain Socket.

4.1.1.2 Monitoring

Property	Default	Unit	Required	Description
metrics	0	Int	No	The metrics port (disable = 0)
metrics_cache_max_age	0	String	No	The time to keep a Prometheus (metrics) response in cache. If this value is specified without units, it is taken as seconds. Setting this parameter to 0 disables caching. It supports the following units as suffixes: 'S' for seconds (default), 'M' for minutes, 'H' for hours, 'D' for days, and 'W' for weeks.
metrics_cache_max_size	256k	String	No	The maximum amount of data to keep in cache when serving Prometheus responses. Changes require restart. This parameter determines the size of memory allocated for the cache even if <code>metrics_cache_max_age</code> or <code>metrics</code> are disabled. Its value, however, is taken into account only if <code>metrics_cache_max_age</code> is set to a non-zero value. Supports suffixes: 'B' (bytes), the default if omitted, 'K' or 'KB' (kilobytes), 'M' or 'MB' (megabytes), 'G' or 'GB' (gigabytes).

Property	Default	Unit	Required	Description
metrics_cert_file		String	No	Certificate file for TLS for Prometheus metrics. This file must be owned by either the user running pgmoneta or root.
metrics_key_file		String	No	Private key file for TLS for Prometheus metrics. This file must be owned by either the user running pgmoneta or root. Additionally permissions must be at least 0640 when owned by root or 0600 otherwise.
metrics_ca_file		String	No	Certificate Authority (CA) file for TLS for Prometheus metrics. This file must be owned by either the user running pgmoneta or root.

4.1.1.3 Management

Property	Default	Unit	Required	Description
management	0	Int	No	The remote management port (disable = 0)

4.1.1.4 Compression

Property	Default	Unit	Required	Description
compression	zstd	String	No	The compression type (none, gzip, client-gzip, server-gzip, zstd, client-zstd, server-zstd, lz4, client-lz4, server-lz4, bzip2, client-bzip2)
compression_level	3	Int	No	The compression level

4.1.1.5 Workers

Property	Default	Unit	Required	Description
workers	0	Int	No	The number of workers that each process can use for its work. Use 0 to disable. Maximum is CPU count

4.1.1.6 Workspace

Property	Default	Unit	Required	Description
workspace	/tmp/pgmoneta-workspace/	String	No	The directory for the workspace that incremental backup can use for its work

4.1.1.7 Storage

Property	Default	Unit	Required	Description
storage_engine	local	String	No	The storage engine type (local, ssh, s3, azure)

4.1.1.8 Encryption

Property	Default	Unit	Required	Description
encryption	none	String	No	The encryption mode for encrypt wal and data <code>none</code> : No encryption <code>aes</code> \ <code>aes-256</code> \ <code>aes-256-cbc</code> : AES CBC (Cipher Block Chaining) mode with 256 bit key length <code>aes-192</code> \ <code>aes-192-cbc</code> : AES CBC mode with 192 bit key length <code>aes-128</code> \ <code>aes-128-cbc</code> : AES CBC mode with 128 bit key length <code>aes-256-ctr</code> : AES CTR (Counter) mode with 256 bit key length <code>aes-192-ctr</code> : AES CTR mode with 192 bit key length <code>aes-128-ctr</code> : AES CTR mode with 128 bit key length

4.1.1.9 Slot management

Property	Default	Unit	Required	Description
create_slot	no	Bool	No	Create a replication slot for all server. Valid values are: yes, no

4.1.1.10 SSH

Property	Default	Unit	Required	Description
ssh_hostname		String	Yes	Defines the hostname of the remote system for connection
ssh_username		String	Yes	Defines the username of the remote system for connection
ssh_base_dir		String	Yes	The base directory for the remote backup
ssh_ciphers	aes-256-ctr, aes-192-ctr, aes-128-ctr	String	No	The supported ciphers for communication. <code>aes</code> \ <code>aes-256</code> \ <code>aes-256-cbc</code> : AES CBC (Cipher Block Chaining) mode with 256 bit key length <code>aes-192</code> \ <code>aes-192-cbc</code> : AES CBC mode with 192 bit key length <code>aes-128</code> \ <code>aes-128-cbc</code> : AES CBC mode with 128 bit key length <code>aes-256-ctr</code> : AES CTR (Counter) mode with 256 bit key length <code>aes-192-ctr</code> : AES CTR mode with 192 bit key length <code>aes-128-ctr</code> : AES CTR mode with 128 bit key length. Otherwise verbatim

4.1.1.11 S3

Property	Default	Unit	Required	Description
s3_aws_region		String	Yes	The AWS region
s3_access_key_id		String	Yes	The IAM access key ID
s3_secret_access_key		String	Yes	The IAM secret access key
s3_bucket		String	Yes	The AWS S3 bucket name
s3_base_dir		String	Yes	The base directory for the S3 bucket

4.1.1.12 Azure

Property	Default	Unit	Required	Description
azure_storage_account		String	Yes	The Azure storage account name
azure_container		String	Yes	The Azure container name
azure_shared_key		String	Yes	The Azure storage account key
azure_base_dir		String	Yes	The base directory for the Azure container

4.1.1.13 Retention

Property	Default	Unit	Required	Description
retention	7, -, -, -	Array	No	The retention time in days, weeks, months, years

4.1.1.14 Verification

Property	Default	Unit	Required	Description
verification	0	String	No	The time between verification of a backup. If this value is specified without units,

it is taken as seconds. Setting this parameter to 0 disables verification. It supports the following units as suffixes: 'S' for seconds (default), 'M' for minutes, 'H' for hours, 'D' for days, and 'W' for weeks. Default is 0 (disabled) |

4.1.1.15 Logging

Property	Default	Unit	Required	Description
log_type	console	String	No	The logging type (console, file, syslog)
log_level	info	String	No	The logging level, any of the (case insensitive) strings FATAL , ERROR , WARN , INFO and DEBUG (that can be more specific as DEBUG1 thru DEBUG5). Debug level greater than 5 will be set to DEBUG5 . Not recognized values will make the log_level be INFO
log_path	pgmoneta.log	String	No	The log file location. Can be a strftime(3) compatible string.

Property	Default	Unit	Required	Description
log_rotation_age	0	String	No	The time after which log file rotation is triggered. If this value is specified without units, it is taken as seconds. Setting this parameter to 0 disables log rotation based on time. It supports the following units as suffixes: 'S' for seconds (default), 'M' for minutes, 'H' for hours, 'D' for days, and 'W' for weeks.
log_rotation_size	0	String	No	The size of the log file that will trigger a log rotation. Supports suffixes: 'B' (bytes), the default if omitted, 'K' or 'KB' (kilobytes), 'M' or 'MB' (megabytes), 'G' or 'GB' (gigabytes). A value of 0 (with or without suffix) disables.
log_line_prefix	%Y-%m-%d %H:%M:%S	String	No	A strftime(3) compatible string to use as prefix for every log line. Must be quoted if contains spaces.
log_mode	append	String	No	Append to or create the log file (append, create)

4.1.1.16 Transport Level Security

Advanced Backup/Restore with pgmoneta

Property	Default	Unit	Required	Description
tls	<code>off</code>	Bool	No	Enable Transport Layer Security (TLS)
tls_cert_file		String	No	Certificate file for TLS. This file must be owned by either the user running pgmoneta or root.
tls_key_file		String	No	Private key file for TLS. This file must be owned by either the user running pgmoneta or root. Additionally permissions must be at least 0640 when owned by root or 0600 otherwise.
tls_ca_file		String	No	Certificate Authority (CA) file for TLS. This file must be owned by either the user running pgmoneta or root.
libev	<code>auto</code>	String	No	Select the libev backend to use. Valid options: <code>auto</code> , <code>select</code> , <code>poll</code> , <code>epoll</code> , <code>iouring</code> , <code>devpoll</code> and <code>port</code>

4.1.1.17 Miscellaneous

Property	Default	Unit	Required	Description
backup_max_rate	0	Int	No	The number of bytes of tokens added every one second to limit the backup rate

Property	Default	Unit	Required	Description
network_max_rate	0	Int	No	The number of bytes of tokens added every one second to limit the network backup rate
blocking_timeout	30	String	No	The number of seconds the process will be blocking for a connection. If this value is specified without units, it is taken as seconds. Setting this parameter to 0 disables it. It supports the following units as suffixes: 'S' for seconds (default), 'M' for minutes, 'H' for hours, 'D' for days, and 'W' for weeks.
keep_alive	on	Bool	No	Have <code>SO_KEEPALIVE</code> on sockets
nodelay	on	Bool	No	Have <code>TCP_NODELAY</code> on sockets
non_blocking	on	Bool	No	Have <code>O_NONBLOCK</code> on sockets
backlog	16	Int	No	The backlog for <code>listen()</code> . Minimum 16
hugepage	try	String	No	Huge page support (<code>off</code> , try , <code>on</code>)

Property	Default	Unit	Required	Description
pidfile		String	No	Path to the PID file. If not specified, it will be automatically set to <code>unix_socket_dir/pgmoneta.<host>.pid</code> where <code><host></code> is the value of the <code>host</code> parameter or <code>all</code> if <code>host = *</code> .

Property	Default	Unit	Required	Description
update_process_title	<code>verbose</code>	String	No	The behavior for updating the operating system process title. Allowed settings are: <code>never</code> (or <code>off</code>), does not update the process title; <code>strict</code> to set the process title without overriding the existing initial process title length; <code>minimal</code> to set the process title to the base description; <code>verbose</code> (or <code>full</code>) to set the process title to the full description. Please note that <code>strict</code> and <code>minimal</code> are honored only on those systems that do not provide a native way to set the process title (e.g., Linux). On other systems, there is no difference between <code>strict</code> and <code>minimal</code> and the assumed behaviour is <code>minimal</code> even if <code>strict</code> is used. <code>never</code> and <code>verbose</code> are always honored, on every system. On Linux systems the process title is always trimmed to 255 characters, while on system that provide a native way to set the process title it can be longer.

4.1.2 Server section

4.1.2.1 Server

Property	Default	Unit	Required	Description
host		String	Yes	The address of the PostgreSQL instance
port		Int	Yes	The port of the PostgreSQL instance
user		String	Yes	The replication user name
wal_slot		String	Yes	The replication slot for WAL

The `user` specified must have the `REPLICATION` option in order to stream the Write-Ahead Log (WAL), and must have access to the `postgres` database in order to get the necessary configuration parameters.

4.1.2.2 Slot management

Property	Default	Unit	Required	Description
create_slot	no	Bool	No	Create a replication slot for this server. Valid values are: yes, no

4.1.2.3 Follow

Property	Default	Unit	Required	Description
follow		String	No	Failover to this server if follow server fails

4.1.2.4 Retention

Property	Default	Unit	Required	Description
retention		Array	No	The retention for the server in days, weeks, months, years

4.1.2.5 WAL shipping

Property	Default	Unit	Required	Description
wal_shipping		String	No	The WAL shipping directory

4.1.2.6 Hot standby

Property	Default	Unit	Required	Description
hot_standby		String	No	Hot standby directory. Single directory or comma separated directories up to 8 (e.g., /path/to/hot/standby1,/path/to/hot/standby2)
hot_standby_overric		String	No	Files to override in the hot standby directory. If multiple hot standbys are specified then this setting is separated by a
hot_standby_tablespaces		String	No	Tablespace mappings for the hot standby. Syntax is [from -> to,?]+. If multiple hot standbys are specified then this setting is separated by a

4.1.2.7 Workers

Property	Default	Unit	Required	Description
workers	-1	Int	No	The number of workers that each process can use for its work. Use 0 to disable, -1 means use the global setting. Maximum is CPU count

4.1.2.8 Transport Level Security

Property	Default	Unit	Required	Description
tls_cert_file		String	No	Certificate file for TLS. This file must be owned by either the user running pgmoneta or root.
tls_key_file		String	No	Private key file for TLS. This file must be owned by either the user running pgmoneta or root. Additionally permissions must be at least 0640 when owned by root or 0600 otherwise.
tls_ca_file		String	No	Certificate Authority (CA) file for TLS. This file must be owned by either the user running pgmoneta or root.

4.1.2.9 Miscellaneous

Property	Default	Unit	Required	Description
backup_max_rate	-1	Int	No	The number of bytes of tokens added every one second to limit the backup rate. Use 0 to disable, -1 means use the global setting
network_max_rate	-1	Int	No	The number of bytes of tokens added every one second to limit the network backup rate. Use 0 to disable, -1 means use the global setting

4.1.2.10 Extra

Property	Default	Unit	Required	Description
extra		String	No	The source directory for retrieval on the server side (details are in the extra section)

The [extra](#) configuration is set in the server section. It is not required, but if you configure this parameter, when you perform a backup using the CLI `pgmoneta-cli -c pgmoneta.conf backup primary`, it will also copy all specified files on the server side and send them back to the client side.

This [extra](#) feature requires the server side to install the `pgmoneta_ext` extension and also make the user `repl` a `SUPERUSER` (this will be improved in the future). Currently, this feature is only available to the `SUPERUSER` role.

You can set up `pgmoneta_ext` by following the README to easily install the extension. There are also more detailed instructions available in the DEVELOPERS documentation.

The format for the [extra](#) parameter is a path to a file or directory. You can list more than one file or directory separated by commas. The format is as follows:

```
extra = /tmp/myfile1, /tmp/myfile2, /tmp/mydir1, /tmp/mydir2
```

4.2 pgmoneta_users.conf

The `pgmoneta_users` configuration defines the users known to the system. This file is created and managed through the `pgmoneta-admin` tool.

The configuration is loaded from either the path specified by the `-u` flag or `/etc/pgmoneta/pgmoneta_users.conf`.

4.3 pgmoneta_admins.conf

The `pgmoneta_admins` configuration defines the administrators known to the system. This file is created and managed through the `pgmoneta-admin` tool.

The configuration is loaded from either the path specified by the `-A` flag or `/etc/pgmoneta/pgmoneta_admins.conf`.

If `pgmoneta` has both Transport Layer Security (TLS) and `management` enabled then `pgmoneta-cli` can connect with TLS using the files `~/.pgmoneta/pgmoneta.key` (must be 0600 permission), `~/.pgmoneta/pgmoneta.crt` and `~/.pgmoneta/root.crt`.

5 Backup

5.1 Create a full backup

We can take a full backup from the primary with the following command

```
pgmoneta-cli backup primary
```

and you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: 1
  Output: 0
  Timestamp: 20240928065644
Outcome:
  Status: true
  Time: 00:00:20
Request:
  Server: primary
Response:
  Backup: 20240928065644
  BackupSize: 8531968
  Compression: 2
  Encryption: 0
  MajorVersion: 17
  MinorVersion: 0
  RestoreSize: 48799744
  Server: primary
  ServerVersion: 0.18.0
```

5.2 View backups

We can list all backups for a server with the following command

```
pgmoneta-cli list-backup primary
```

and you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: 2
  Output: 0
  Timestamp: 20240928065812
Outcome:
  Status: true
  Time: 00:00:00
Request:
```

```
Server: primary
Response:
Backups:
  - Backup: 20240928065644
    BackupSize: 8531968
    Comments: ''
    Compression: 2
    Encryption: 0
    Incremental: false
    Keep: false
    RestoreSize: 48799744
    Server: primary
    Valid: 1
    WAL: 0
MajorVersion: 17
MinorVersion: 0
Server: primary
ServerVersion: 0.18.0
```

5.3 Sorting backups

You can sort the backup list by timestamp using the `--sort` option:

```
pgmoneta-cli list-backup primary --sort asc
```

for ascending order (oldest first), or

```
pgmoneta-cli list-backup primary --sort desc
```

for descending order (newest first).

5.4 Create an incremental backup

We can take an incremental backup from the primary with the following command

```
pgmoneta-cli backup primary 20240928065644
```

and you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: 1
  Output: 0
  Timestamp: 20240928065730
Outcome:
  Status: true
  Time: 00:00:20
```

```
Request:
  Server: primary
Response:
  Backup: 20240928065750
  BackupSize: 124312
  Compression: 2
  Encryption: 0
  Incremental: true
  MajorVersion: 17
  MinorVersion: 0
  RestoreSize: 48799744
  Server: primary
  ServerVersion: 0.18.0
```

Incremental backups are supported when using PostgreSQL 17+. Note that currently branching is not allowed for incremental backup – a backup can have at most 1 incremental backup child.

5.5 View backups

We can list all backups for a server with the following command

```
pgmoneta-cli list-backup primary
```

and you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: 2
  Output: 0
  Timestamp: 20240928065812
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Server: primary
Response:
  Backups:
    - Backup: 20240928065644
      BackupSize: 8531968
      Comments: ''
      Compression: 2
      Encryption: 0
      Incremental: false
      Keep: false
      RestoreSize: 48799744
      Server: primary
      Valid: 1
      WAL: 0
  MajorVersion: 17
```

```
MinorVersion: 0
Server: primary
ServerVersion: 0.18.0
```

5.6 Backup information

You can list the information about a backup

```
pgmoneta-cli -c pgmoneta.conf info primary newest
```

and you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: info
  Output: text
  Timestamp: 20241025163541
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Backup: newest
  Server: primary
Response:
  Backup: 20241019163516
  BackupSize: 6.54MB
  CheckpointHiLSN: 0
  CheckpointLoLSN: 4F0000B8
  Comments: ''
  Compression: zstd
  Elapsed: 4
  Encryption: none
  EndHiLSN: 0
  EndLoLSN: 4F000158
  EndTimeline: 1
  Incremental: false
  Keep: true
  MajorVersion: 17
  MinorVersion: 0
  NumberOfTablespaces: 0
  RestoreSize: 45.82MB
  Server: primary
  ServerVersion: 0.18.0
  StartHiLSN: 0
  StartLoLSN: 4F000060
  StartTimeline: 1
  Tablespaces: {}
  Valid: yes
  WAL: 000000001000000000000000004F
```


5.7 Create a crontab

Lets create a `crontab` such that a backup is made every day,

First, take a full backup if you are using PostgreSQL 17+,

```
pgmoneta-cli backup primary
```

then you can use incremental backup for your daily jobs,

```
crontab -e
```

and insert

```
0 6 * * * pgmoneta-cli backup primary latest
```

for taking an incremental backup every day at 6 am.

Otherwise use the full backup in the cron job.

5.8 Verify backup integrity

pgmoneta creates a SHA512 checksum file(`backup.sha512`) for each backup at the backup root directory, which can be used to verify the integrity of the files.

Using `sha512sum`:

```
cd <path-to-specific-backup-directory>
sha512sum --check backup.sha512
```

The `verification` parameter can be use to control how frequently pgmoneta verifies the integrity of backup files. You can configure this in `pgmoneta.conf`:

```
[pgmoneta]
.
.
.
verification = 3600
```

For example, setting `verification = 3600` or `verification = 1H` will perform integrity checks every hour.

6 Retention policy

The retention policy decide for how long a backup should be kept.

6.1 Retention configuration

The configuration is done in the main configuration section, or by a server basis with

Property	Default	Unit	Required	Description
retention	7, -, -, -	Array	No	The retention time in days, weeks, months, years

which means by default that backups are kept for 7 days.

6.2 Defining retention

Defining a retention policy is very important because it defines how you will be able to restore your system from the backups.

The key is to decide what your policy is, for example

7, 4, 12, 5

to keep backups for 7 days, one backup each Monday for 4 weeks, one backup for each month, and 5 yearly backups.

This is a very hard property to configure since it depends on the size of the database cluster and therefore how big your backups are.

If you want to restore from the latest backup plus the Write-Ahead Log (WAL) then the default **pgmoneta** policy maybe is enough.

Note that if a backup has an incremental backup child that depends on it, its data will be rolled up to its child before getting deleted.

6.3 Retention check

The retention check runs every 5 minutes, and will delete one backup per run.

You can change this to every 30 minutes by

```
retention_interval = 1800
```

under the [pgmoneta] configuration.

7 Keeping backups

7.1 List backups

First, we can list our current backups using

```
pgmoneta-cli list-backup primary
```

you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: list-backup
  Output: text
  Timestamp: 20241018092853
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Server: primary
Response:
  Backups:
    - Backup: 20241012091219
      BackupSize: 6.11MB
      Comments: ''
      Compression: zstd
      Encryption: none
      Keep: false
      RestoreSize: 39.13MB
      Server: primary
      Valid: yes
      WAL: 0
  MajorVersion: 17
  MinorVersion: 0
  Server: primary
  ServerVersion: 0.18.0
```

As you can see backup 20241012091219 has a **Keep** flag of **false**.

7.2 Keep a backup

Now, in order to keep the backup which means that it won't be deleted by the retention policy you can issue the following command,

```
pgmoneta-cli retain primary 20241012091219
```

and get output like,

```
Header:
  ClientVersion: 0.18.0
  Command: retain
  Output: text
  Timestamp: 20241018094129
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Backup: 20241012091219
  Server: primary
Response:
  Backups:
    - 20241012091219
  Cascade: false
  Comments: ''
  Compression: none
  Encryption: none
  Keep: true
  MajorVersion: 17
  MinorVersion: 0
  Server: primary
  ServerVersion: 0.18.0
  Valid: yes
```

and you can see that the backup has a **Keep** flag of **true**.

7.3 Describe a backup

Now, you may want to add a description to your backup, and as you can see

```
Header:
  ClientVersion: 0.18.0
  Command: retain
  Output: text
  Timestamp: 20241018094129
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Backup: 20241012091219
  Server: primary
Response:
  Backups:
    - 20241012091219
  Cascade: false
  Comments: ''
  Compression: none
  Encryption: none
```

```
Keep: true
MajorVersion: 17
MinorVersion: 0
Server: primary
ServerVersion: 0.18.0
Valid: yes
```

there is a `Comments` field to do that.

You can use the command,

```
pgmoneta-cli annotate primary 20241012091219 add Type "Main fall backup"
```

which will give

```
Header:
  ClientVersion: 0.18.0
  Command: annotate
  Output: text
  Timestamp: 20241018095906
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Action: add
  Backup: 20241012091219
  Comment: Main fall backup
  Key: Type
  Server: primary
Response:
  Backup: 20241012091219
  BackupSize: 6.11MB
  CheckpointHiLSN: 0
  CheckpointLoLSN: 33554560
  Comments: Type|Main fall backup
  Compression: zstd
  Elapsed: 1
  Encryption: none
  EndHiLSN: 0
  EndLoLSN: 33554776
  EndTimeline: 1
  Keep: true
  MajorVersion: 17
  MinorVersion: 0
  NumberOfTablespaces: 0
  RestoreSize: 39.13MB
  Server: primary
  ServerVersion: 0.18.0
  StartHiLSN: 0
  StartLoLSN: 33554472
  StartTimeline: 1
```

```
Tablespaces:
```

```
Valid: yes
```

```
WAL: 00000001000000000000000002
```

As you can see the `Comments` field with the `Type` key.

The `annotate` command has `add`, `update` and `remove` commands to modify the `Comments` field.

7.4 Put a backup back into retention

When you don't need a backup anymore you can put into retention again by,

```
pgmoneta-cli expunge primary 20241012091219
```

will give,

```
Header:
  ClientVersion: 0.18.0
  Command: expunge
  Output: text
  Timestamp: 20241018101839
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Backup: 20241012091219
  Server: primary
Response:
  Backup: 20241012091219
  Comments: Type|Main fall backup
  Compression: none
  Encryption: none
  Keep: false
  MajorVersion: 17
  MinorVersion: 0
  Server: primary
  ServerVersion: 0.18.0
  Valid: yes
```

and now, the `Keep` flag is back to **false**.

7.5 Cascade mode

You can retain/expunge the entire incremental backup chain using the `--cascade` option. This will retain/expunge all backups along the line until the root full backup.

Say you have an incremental backup chain: 20250625055547 (`incremental`) -> 20250625055528 (`incremental`) -> 20250625055517(`full`). Running `pgmoneta-cli retain --cascade primary 20250625055547` will also retain backup 20250625055528 and backup 20250625055517.

This will give

```
Header:
  ClientVersion: 0.18.0
  Command: retain
  Compression: none
  Encryption: none
  Output: text
  Timestamp: 20250625055654
Outcome:
  Status: true
  Time: 00:00:0.1032
Request:
  Backup: newest
  Cascade: true
  Server: primary
Response:
  BackupSize: 0.00B
  Backups:
    - 20250625055547
    - 20250625055528
    - 20250625055517
  BiggestFileSize: 0.00B
  Cascade: true
  Comments: ''
  Compression: none
  Delta: 0.00B
  Encryption: none
  Keep: true
  MajorVersion: 17
  MinorVersion: 0
  RestoreSize: 0.00B
  Server: primary
  ServerVersion: 0.18.0
  Valid: yes
  WAL: 0.00B
```


8 Restore

8.1 Restore a backup

We can restore a backup from the primary with the following command

```
pgmoneta-cli restore primary newest current /tmp
```

where

- `current` means copy the Write-Ahead Log (WAL), and restore to first stable checkpoint
- `name=X` means copy the Write-Ahead Log (WAL), and restore to the label specified
- `xid=X` means copy the Write-Ahead Log (WAL), and restore to the XID specified
- `time=X` means copy the Write-Ahead Log (WAL), and restore to the timestamp specified
- `lsn=X` means copy the Write-Ahead Log (WAL), and restore to the Log Sequence Number (LSN) specified
- `inclusive=X` means that the restore is inclusive of the specified information
- `timeline=X` means that the restore is done to the specified information timeline
- `action=X` means which action should be executed after the restore (pause, shutdown)
- `primary` means that the cluster is setup as a primary
- `replica` means that the cluster is setup as a replica

More information

And, you will get output like

```
Header:
  ClientVersion: 0.18.0
  Command: 3
  Output: 0
  Timestamp: 20240928130406
Outcome:
  Status: true
  Time: 00:00:00
Request:
  Backup: newest
  Directory: /tmp
  Position: current
  Server: primary
Response:
  Backup: 20240928065644
  BackupSize: 8531968
  Comments: ''
  Compression: 2
  Encryption: 0
  MajorVersion: 17
```

```
MinorVersion: 0  
RestoreSize: 48799744  
Server: primary  
ServerVersion: 0.18.0
```

This command take the latest backup and all Write-Ahead Log (WAL) segments and restore it into the `/tmp/primary-20240928065644` directory for an up-to-date copy.

9 Prometheus metrics

pgmoneta support Prometheus metrics.

We enabled the Prometheus metrics in the original configuration by setting

```
metrics = 5001
```

in `pgmoneta.conf`.

9.1 Access Prometheus metrics

You can now access the metrics via

```
http://localhost:5001/metrics
```

10 Administration access

You can access **pgmoneta** from a remote machine if you enable access.

10.1 Configuration

First, you need to enable remote access by adding

```
management = 5002
```

in `pgmoneta.conf` in the `[pgmoneta]` section.

10.2 Administrators

Next, you will need to add one or more administrators in `pgmoneta_admins.conf` through

```
pgmoneta-admin -f /etc/pgmoneta/pgmoneta_admins.conf user add
```

for example with a user name of `admin` and `secretpassword` as the password.

10.3 Restart pgmoneta

You have to restart **pgmoneta** to make the changes take effect.

10.4 Connect to pgmoneta

Then you will use the `pgmoneta-cli` tool to access **pgmoneta** with

```
pgmoneta-cli -h myhost -p 5002 -U admin status
```

to execute the `status` command after have entered the password.

11 Shutdown

You can test the status of **pgmoneta** and shutdown either locally or from a remote machine.

11.1 ping

You can check if **pgmoneta** is running by

```
pgmoneta-cli ping
```

and check the output.

11.2 mode

pgmoneta detects when a server is down. You can bring a server online or offline using the mode command, like

```
pgmoneta-cli mode primary offline
```

or

```
pgmoneta-cli mode primary online
```

pgmoneta will keep basic services running for an offline server such that you can verify a backup or do a restore.

11.3 shutdown

You can shutdown **pgmoneta** by

```
pgmoneta-cli shutdown
```

and check the output.

12 Running pgmoneta with Docker

You can run **pgmoneta** using Docker instead of compiling it manually.

12.1 Prerequisites

- **[Docker]**[docker] or **[Podman]**[podman] must be installed on the server where PostgreSQL is running.
- Ensure PostgreSQL is configured to allow external connections.

12.2 Update the configuration file if needed:

```
[pgmoneta]
host = *
metrics = 5001

base_dir = /home/pgmoneta/backup

compression = zstd

retention = 7

log_type = file
log_level = info
log_path = /tmp/pgmoneta.log

unix_socket_dir = /tmp/

[primary]
host = localhost
port = 5432
user = repl
wal_slot = repl
```

12.3 Step 1: Enable External PostgreSQL Access

Modify the local PostgreSQL server's `postgresql.conf` file to allow connections from outside:

```
listen_addresses = '*'
```

Update `pg_hba.conf` to allow remote connections:

```
host      all      all      0.0.0.0/0      scram-sha-256
```

Then, restart PostgreSQL for the changes to take effect:

```
sudo systemctl restart postgresql
```

12.4 Step 2: Clone the Repository

```
git clone https://github.com/pgmoneta/pgmoneta.git
cd pgmoneta
```

12.5 Step 3: Build the Docker Image

There are two Dockerfiles available: 1. **Alpine-based image**

Using Docker

```
docker build -t pgmoneta:latest -f ./contrib/docker/Dockerfile.alpine .
```

Using Podman

```
podman build -t pgmoneta:latest -f ./contrib/docker/Dockerfile.alpine .
```

2. **Rocky Linux 9-based image**

Using Docker

```
docker build -t pgmoneta:latest -f ./contrib/docker/Dockerfile.rocky9 .
```

Using Podman

```
podman build -t pgmoneta:latest -f ./contrib/docker/Dockerfile.rocky9 .
```

12.6 Step 4: Run pgmoneta as a Docker Container

Once the image is built, run the container using:

- **Using Docker**

```
docker run -d --name pgmoneta --network host pgmoneta:latest
```

- **Using Podman**

```
podman run -d --name pgmoneta --network host pgmoneta:latest
```

12.7 Step 5: Verify the Container

Check if the container is running:

- **Using Docker**

```
docker ps | grep pgmoneta -->
```

- **Using Podman**

```
podman ps | grep pgmoneta
```

Check logs for any errors:

- **Using Docker**

```
docker logs pgmoneta
```

- **Using Podman**

```
podman logs pgmoneta
```

You can also inspect the exposed metrics at:

```
http://localhost:5001/metrics
```

You can stop the container using

- **Using Docker**

```
docker stop pgmoneta
```

- **Using Podman**

```
podman stop pgmoneta
```

You can exec into the container and run the cli commands as

```
docker exec -it pgmoneta /bin/bash
#or using podman
podman exec -it pgmoneta /bin/bash

cd /etc/pgmoneta
/usr/local/bin/pgmoneta-cli -c pgmoneta.conf shutdown
```

See this for more cli commands.

You can access the three binaries at `/usr/local/bin`

13 Troubleshooting

13.1 Could not get version for server

If you get this `FATAL` during startup check your PostgreSQL logins

```
psql postgres
```

and

```
psql -U repl postgres
```

And, check the PostgreSQL logs for any error.

Setting `log_level` to `DEBUG5` in `pgmoneta.conf` could provide more information about the error.

14 Acknowledgement

14.1 Authors

pgmoneta was created by the following authors:

```
Jesper Pedersen <jesperpedersen.db@gmail.com>
David Fetter <david@fetter.org>
Will Leinweber <will@bitfission.com>
Luca Ferrari <fluca1978@gmail.com>
Nikita Bugrovsky <nbugrovs@redhat.com>
Mariam Fahmy <mariamfahmy66@gmail.com>
Jichen Xu <kyokitisin@gmail.com>
Saurav Pal <resyfer.dev@gmail.com>
Bokket <bokkett@gmail.com>
Haoran Zhang <andrewzhr9911@gmail.com>
Hazem Alrawi <hazemalrawi7@gmail.com>
Shahryar Soltanpour <shahryar.soltanpour@gmail.com>
Shikhar Soni <shikharish05@gmail.com>
Nguyen Cong Nhat Le <lenguyencongnhat2001@gmail.com>
Chao Gu <chadraven369@gmail.com>
Luchen Zhao <lucian.zlc@gmail.com>
Joan Jeremiah J <joanjeremiah04@gmail.com>
Iury Santos <iuryroberto@gmail.com>
Palak Chaturvedi <palakchaturvedi2843@gmail.com>
Jakub Jirutka <jakub@jirutka.cz>
Mario Rodas
Annupamaa <annu242005@gmail.com>
Ashutosh Sharma <ash2003sharma@gmail.com>
Mohab Yaser <mohabyaseroofficial2003@gmail.com>
Georg Pfuetzenreuter <mail@georg-pfuetzenreuter.net>
Ahmed Ashour <a8087027@gmail.com>
Sangkeun J.C. Kim <jchrys@me.com>
Tejas Tyagi <tejastyagi.tt@gmail.com>
Aryan Arora <aryanarora.w1@gmail.com>
Arshdeep Singh <balارش535@gmail.com>
Din Xin Chen <s990346@gmail.com>
Mingzhuo Yin <yinmingzhuo@gmail.com>
Vanes Angelo <k124k3n@gmail.com>
Bassam Adnan <mailbassam@gmail.com>
```

14.2 Committers

```
Jesper Pedersen <jesperpedersen.db@gmail.com>
Haoran Zhang <andrewzhr9911@gmail.com>
```

14.3 Contributing

Contributions to **pgmoneta** are managed on GitHub

- Ask a question
- Raise an issue
- Feature request
- Code submission

Contributions are most welcome!

Please, consult our Code of Conduct policies for interacting in our community.

Consider giving the project a star on GitHub if you find it useful. And, feel free to follow the project on X as well.

15 License

Copyright (C) 2025 The pgmoneta community

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, **this** list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, **this** list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from **this** software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BSD-3-Clause

15.1 libart

Our adaptive radix tree (ART) implementation is based on The Adaptive Radix Tree: ARTful Indexing for Main-Memory Databases and libart which has a 3-BSD license as

Copyright (c) 2012, Armon Dadgar
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, **this** list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, **this** list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the organization nor the names of its contributors may be used to endorse or promote products derived from **this** software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ARMON DADGAR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.