

उन्नत बैकअप/पुनर्स्थापन **pgmoneta** के
साथे

Contents

1	प्रस्तावना	3
2	परिचय	4
2.1	विशेषताएँ	4
2.2	प्लेटफॉर्मस	5
3	स्थापना	6
3.1	रॉकी लिनक्स 9.x	6
3.2	PostgreSQL 17	6
3.3	pgmoneta	8
4	कॉन्फिगरेशन	10
4.1	pgmoneta.conf	10
4.1.1	pgmoneta	10
4.1.2	सर्वर अनुभाग	26
4.1.3	pgmoneta_users.conf	31
4.1.4	pgmoneta_admins.conf	31
5	बैकअप	32
5.1	पूर्ण बैकअप बनाएं	32
5.2	बैकअप देखना	32
5.3	वृद्धिशील बैकअप बनाएं	33
5.4	बैकअप देखें	34
5.5	बैकअप जानकारी	34
5.6	क्रॉनटैब बनाना	35
5.7	बैकअप अखंडता सत्यापित करें	36
6	रिटेंशन पॉलिसी	37
6.1	रिटेंशन कॉन्फिगरेशन	37
6.2	रिटेंशन को परिभाषित करना	37
6.3	रिटेंशन चेक	37
7	बैकअप रखना	39
7.1	बैकअप्स की सूची	39
7.2	बैकअप को रखना	39
7.3	बैकअप का वर्णन करना	40
7.4	बैकअप को फिर से रिटेंशन में डालना	42

8	पुनर्स्थापना	43
8.1	बैकअप पुनर्स्थापित करना	43
9	प्रोमेथियस मेट्रिक्स	45
9.1	प्रोमेथियस मेट्रिक्स तक पहुँच	45
10	प्रशासनिक पहुँच	46
10.1	कॉन्फिगरेशन	46
10.2	व्यवस्थापक	46
10.3	pgmoneta को पुनः प्रारंभ करें	46
10.4	pgmoneta से कनेक्ट करें	46
11	शटडाउन	47
11.1	पिंग	47
11.2	शटडाउन	47
12	Docker के साथ pgmoneta निष्पादित करना	48
12.1	पूर्वापेक्षाएँ	48
12.2	यदि आवश्यक हो तो कॉन्फिगरेशन फ़ाइल को अपडेट करें:	48
12.3	चरण 1: PostgreSQL के लिए बाहरी पहुँच सक्षम करें	48
12.4	चरण 2: रिपॉजिटरी को क्लोन करें	49
12.5	चरण 3: डॉकर इमेज बनाएं	49
12.6	चरण 4: pgmoneta को डॉकर कंटेनर के रूप में चलाएँ	49
12.7	चरण 5: कंटेनर को सत्यापित करें	50
13	समस्या निवारण	51
13.1	सर्वर के लिए संस्करण प्राप्त नहीं कर सके	51
14	आभार	52
14.1	लेखक	52
14.2	कमिटरर्स	52
14.3	योगदान	53
15	लाइसेंस	54
15.1	libart	54

1 प्रस्तावना

Acme Boot एक स्टार्टअप कंपनी है जिसने **PostgreSQL** को अपनी डेटाबेस तकनीक के रूप में उपयोग करने का निर्णय लिया है।

डेटाबेस क्लस्टर के लिए निम्नलिखित तकनीकों का उपयोग किया जाएगा:

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

ध्यान दें कि यह मार्गदर्शिका प्लेटफॉर्म के **pgmoneta** पहलू पर केंद्रित होगी।

2 परिचय

pgmoneta PostgreSQL के लिए एक बैकअप / पुनर्स्थापन समाधान है।

आदर्श रूप से, आपको बैकअप और आपदा पुनर्प्राप्ति की आवश्यकता नहीं होनी चाहिए, लेकिन वास्तविक दुनिया में ऐसा नहीं होता।

संभावित परिदृश्य जो हो सकते हैं:

- डेटा भ्रष्टाचार
- सिस्टम विफलता
- मानवीय त्रुटि
- प्राकृतिक आपदा

इस स्थिति में डेटाबेस प्रशासक की जिम्मेदारी होती है कि वह डेटाबेस सिस्टम को पुनः ऑनलाइन लाए और सही पुनर्प्राप्ति बिंदु तक पहुंचाए।

दो प्रमुख कारक होते हैं:

- रिकवरी प्वाइंट ऑब्जेक्टिव (**RPO**): वह अधिकतम लक्षित समय अवधि जिसमें किसी बड़ी घटना के कारण आईटी सेवा से डेटा खो सकता है।
- रिकवरी टाइम ऑब्जेक्टिव (**RTO**): वह लक्षित अवधि और सेवा सूत्र, जिसके भीतर किसी आपदा (या व्यवधान) के बाद एक व्यावसायिक प्रक्रिया को बहाल किया जाना चाहिए, ताकि व्यावसायिक निरंतरता में रुकावट से जुड़े अस्वीकार्य परिणामों से बचा जा सके।

आप दोनों को जितना संभव हो शून्य के करीब रखना चाहेंगे, क्योंकि **RPO = 0** का मतलब है कि आप डेटा नहीं खोएंगे, और **RTO = 0** का मतलब है कि आपका सिस्टम तुरंत पुनः चालू हो जाएगा। हालाँकि, इसे पूरा करना इतना आसान नहीं है।

pgmoneta ऐसी विशेषताओं पर केंद्रित है जो डेटाबेस सिस्टम को इन लक्ष्यों के करीब पहुंचने में सक्षम बनाएंगी, जिससे 99.99% या उससे अधिक की उच्च उपलब्धता को लागू किया जा सके और इसे मानक उपकरणों के माध्यम से मॉनिटर किया जा सके।

pgmoneta का नाम रोमन मेमोरी की देवी पर रखा गया है।

2.1 विशेषताएँ

- पूर्ण बैकअप
- पुनर्स्थापन
- संपीड़न (gzip, zstd, lz4, bzip2)
- AES एन्क्रिप्शन समर्थन
- सिमलिक समर्थन

- WAL शिपिंग समर्थन
- हॉट स्टैंडबाय
- प्रमिथियस समर्थन
- रिमोट प्रबंधन
- ऑफलाइन मोड
- ट्रांसपोर्ट लेयर सिक्योरिटी (TLS) v1.2+ समर्थन
- डेमन मोड
- उपयोगकर्ता वॉल्ट

2.2 प्लेटफॉर्म

समर्थित प्लेटफॉर्म हैं:

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

3 स्थापना

3.1 रॉकी लिनक्स 9.x

हम वितरण को उनकी वेबसाइट से डाउनलोड कर सकते हैं: Rocky Linux

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

स्थापना और सेटअप इस गाइड के दायरे से बाहर है।

आदर्श रूप से, आप **PostgreSQL** और **pgmoneta** चलाने के लिए समर्पित उपयोगकर्ता खाते का उपयोग करेंगे:

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

pgmoneta के लिए एक कॉन्फिगरेशन डायरेक्टरी जोड़ें:

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

अब, फायरवॉल में उन पोर्ट्स को खोलें जिनकी आवश्यकता होगी:

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

3.2 PostgreSQL 17

हम PostgreSQL 17 को आधिकारिक YUM repository से सामुदायिक बाइनरी के साथ स्थापित करेंगे:

x86_64

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

और इंस्टॉलेशन निम्न कमांड के माध्यम से करें:

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

सबसे पहले, ~/.bashrc को अपडेट करें:

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

Ctrl-d दबाएं और फिर:

```
source ~/.bashrc
```

का उपयोग करके Bash वातावरण को फिर से लोड करें।

अब PostgreSQL इनिशियलाइज़ करें:

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

8 जीबी मेमोरी मशीन के लिए कॉन्फ़िगरेशन अपडेट करें।

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

कृपया, अपने लोकल सेटअप के लिए सेटअप तैयार करने हेतु अन्य स्रोतों की जाँच करें।

अब, हम PostgreSQL शुरू करने के लिए तैयार हैं।

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

डेटाबेस से कनेक्ट करें, रेप्लिकेशन उपयोगकर्ता जोड़ें, और Write-Ahead Log (WAL) स्लॉट बनाएं:

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

3.3 pgmoneta

pgmoneta को आधिकारिक YUM repository से इंस्टॉल करें:

```
dnf install -y pgmoneta
```

सबसे पहले, हमें **pgmoneta** इंस्टॉलेशन के लिए एक मास्टर सुरक्षा कुंजी बनाने की आवश्यकता होगी, इसके लिए निम्न कमांड का उपयोग करें:

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

फिर, हम **pgmoneta** के लिए कॉन्फ़िगरेशन बनाएंगे,

```
cat > /etc/pgmoneta/pgmoneta.conf
[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
```

और फ़ाइल को सहेजने के लिए **Ctrl-d** के साथ समाप्त करें।

फिर, हम उपयोगकर्ता कॉन्फ़िगरेशन बनाएंगे,

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

अब, हम बेस डायरेक्टरी बनाएंगे और **pgmoneta** शुरू करेंगे,

```
mkdir backup
pgmoneta -d
```

4 कॉन्फिगरेशन

4.1 pgmoneta.conf

कॉन्फिगरेशन या तो `-c` ध्वज द्वारा निर्दिष्ट पथ से या `/etc/pgmoneta/pgmoneta.conf` से लोड होती है।

`pgmoneta` की कॉन्फिगरेशन को [और] अक्षरों का उपयोग करके खंडों में विभाजित किया गया है।

मुख्य खंड, जिसे `[pgmoneta]` कहा जाता है, वह है जहाँ आप `pgmoneta` की समग्र गुणों को कॉन्फिगर करते हैं।

अन्य खंडों का नामकरण कोई विशेष आवश्यकता नहीं है, इसलिए आप उन्हें अर्थपूर्ण नाम दे सकते हैं, जैसे `[primary]` मुख्य उदाहरण के लिए। PostgreSQL

सभी गुण `key = value` प्रारूप में होते हैं।

और ; का उपयोग टिप्पणियों के लिए किया जा सकता है; इन्हें पंक्ति के पहले अक्षर के रूप में होना चाहिए। Bool डेटा प्रकार निम्नलिखित मानों को समर्थित करता है: `on, yes, 1, true, off, no, 0` और `false`।

`pgmoneta` को `localhost` पर चलाने के लिए एक sample कॉन्फिगरेशन देखें।

ध्यान दें, PostgreSQL 13+ आवश्यक है, साथ ही `wal_level` को `replica` या `logical` स्तर पर सेट करना आवश्यक है।

4.1.1 pgmoneta

4.1.1.1 सामान्य

गुण	डिफॉल्ट	इकाई	आवश्यक	विवरण
host		String	हाँ	pgmoneta के लिए बाइंड पता
unix_socket_dir		String	हाँ	Unix डोमेन सॉकेट स्थान
base_dir		String	हाँ	बैकअप के लिए बेस निर्देशिका

ध्यान दें, यदि `host` / से शुरू होता है, तो यह एक पथ को दर्शाता है और `pgmoneta` Unix डोमेन सॉकेट का उपयोग करके कनेक्ट होगा।

4.1.1.2 निगरानी

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
metrics	0	Int	नहीं	मीट्रिक्स पोर्ट (अक्षम = 0)
met- rics_cache_max_age	0	String	नहीं	Prometheus (metrics) प्रतिक्रिया को कैश में रखने के लिए सेकंड की संख्या। यदि शून्य पर सेट किया जाता है, तो कैशिंग अक्षम हो जाएगी। इसे एक स्ट्रिंग के रूप में सेट किया जा सकता है, जैसे 2m जो 2 मिनट का संकेत देता है

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
met- rics_cache_max_size	256k	String	नहीं	<p>Prometheus प्रतिक्रियाओं को सर्व करते समय कैश में रखने के लिए अधिकतम डेटा आकार। बदलाव के लिए पुनः आरंभ की आवश्यकता होती है। यह पैरामीटर तब उपयोग किया जाएगा जब</p> <p><code>metrics_cache_max_age</code> या <code>metrics</code> अक्षम हों। इसके मान को केवल तब ध्यान में रखा जाएगा जब</p> <p><code>metrics_cache_max_age</code> शून्य से बड़ा हो। समर्थन करता है प्रत्यय: 'B' (बाइट्स), यदि प्रत्यय न हो तो डिफ़ॉल्ट, 'K' या 'KB' (किलोबाइट्स), 'M' या 'MB' (मेगाबाइट्स), 'G' या 'GB' (गिगाबाइट्स)।</p>

4.1.1.3 प्रबंधन

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
management	0	Int	नहीं	दूरस्थ प्रबंधन पोर्ट (अक्षम = 0)

4.1.1.4 संपीड़न

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
compression	zstd	String	नहीं	संपीड़न प्रकार (none, gzip, client-gzip, server-gzip, zstd, client-zstd, server-zstd, lz4, client-lz4, server-lz4, bzip2, client-bzip2)
compression_level	3	Int	नहीं	संपीड़न स्तर

4.1.1.5 श्रमिक

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
workers	0	Int	नहीं	प्रत्येक प्रक्रिया के लिए श्रमिकों की संख्या। अक्षम करने के लिए 0 का उपयोग करें। अधिकतम CPU की संख्या है।

4.1.1.6 कार्यक्षेत्र

उन्नत बैकअप/पुनर्स्थापन pgmoneta के साथ

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
कार्यक्षेत्र (workspace)	/tmp/pgmoneta- workspace/	String	नहीं	वह डायरेक्टरी जिसे वृद्धिशील (incremental) बैकअप अपने कार्य के लिए उपयोग कर सकता है।

4.1.1.7 भंडारण

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
storage_engine	local	String	नहीं	भंडारण इंजन प्रकार (local, ssh, s3, azure)

4.1.1.8 एन्क्रिप्शन

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
encryption	none	String	नहीं	<p>WAL और डेटा को एन्क्रिप्ट करने का मोड <code>none</code>: कोई एन्क्रिप्शन नहीं</p> <p><code>aes \ aes -256 \ aes -256-cbc</code>: AES CBC (Cipher Block Chaining) मोड 256 बिट कुंजी लंबाई के साथ</p> <p><code>aes-192 \ aes-192-cbc</code>: AES CBC मोड 192 बिट कुंजी लंबाई के साथ</p> <p><code>aes-128 \ aes-128-cbc</code>: AES CBC मोड 128 बिट कुंजी लंबाई के साथ</p> <p><code>aes-256-ctr</code>: AES CTR (Counter) मोड 256 बिट कुंजी लंबाई के साथ</p> <p><code>aes-192-ctr</code>: AES CTR मोड 192 बिट कुंजी लंबाई के साथ</p> <p><code>aes-128-ctr</code>: AES CTR मोड 128 बिट कुंजी लंबाई के साथ</p>

4.1.1.9 स्लॉट प्रबंधन

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
create_slot	no	Bool	नहीं	सभी सर्वर के लिए एक प्रतिकृति स्लॉट बनाएं। वैध मान हैं: yes, no

4.1.1.10 SSH

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
ssh_hostname		String	हाँ	कनेक्शन के लिए रिमोट सिस्टम का होस्टनाम निर्दिष्ट करता है
ssh_username		String	हाँ	कनेक्शन के लिए रिमोट सिस्टम का उपयोगकर्ता नाम निर्दिष्ट करता है
ssh_base_dir		String	हाँ	रिमोट बैकअप के लिए आधार डायरेक्टरी

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
ssh_ciphers	aes-256-ctr, aes-192-ctr, aes-128-ctr	String	नहीं	<p>संचार के लिए समर्थित सिफर।</p> <p><code>aes</code> \ <code>aes-256</code> \ <code>aes-256-cbc</code>: 256-बिट कुंजी लंबाई के साथ AES CBC (सिफर ब्लॉक चेनिंग) मोड <code>aes-192</code> \ <code>aes-192-cbc</code>: 192-बिट कुंजी लंबाई के साथ AES CBC मोड <code>aes-128</code> \ <code>aes-128-cbc</code>: 128-बिट कुंजी लंबाई के साथ AES CBC मोड <code>aes-256-ctr</code>: 256-बिट कुंजी लंबाई के साथ AES CTR (काउंटर) मोड <code>aes-192-ctr</code>: 192-बिट कुंजी लंबाई के साथ AES CTR मोड <code>aes-128-ctr</code>: 128-बिट कुंजी लंबाई के साथ AES CTR मोड। अन्यथा, मूल रूप में।</p>

4.1.1.11 S3

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
s3_aws_region		String	हाँ	AWS क्षेत्र
s3_access_key_id		String	हाँ	IAM एक्सेस कुंजी आईडी
s3_secret_access_key		String	हाँ	IAM गुप्त एक्सेस कुंजी
s3_bucket		String	हाँ	AWS S3 बकेट का नाम

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
s3_base_dir		String	हाँ	S3 बकेट के लिए बेस निर्देशिका

4.1.1.12 Azure

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
azure_storage_ac-count		String	हाँ	Azure स्टोरेज खाता नाम
azure_container		String	हाँ	Azure कंटेनर नाम
azure_shared_key		String	हाँ	Azure स्टोरेज खाता कुंजी
azure_base_dir		String	हाँ	Azure कंटेनर के लिए बेस निर्देशिका

4.1.1.13 संग्रहण

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
retention	7, -, -, -	Array	नहीं	संग्रहण समय दिन, सप्ताह, महीने, साल में

4.1.1.14 सत्यापन

गुण	डिफ़ॉल्ट	इकाई	आवश्यक	विवरण
verification	0	String	नहीं	बैकअप के सत्यापन के बीच का समय। यदि यह मान इकाइयों के बिना निर्दिष्ट किया जाता है, तो इसे सेकंड के रूप में लिया जाता है। इस पैरामीटर को 0 पर सेट करने से सत्यापन अक्षम हो जाता है। यह प्रत्यय के रूप में निम्नलिखित इकाइयों का समर्थन करता है: सेकंड के लिए 'S' (डिफ़ॉल्ट), मिनटों के लिए 'M', घंटों के लिए 'H', दिनों के लिए 'D' और हफ्तों के लिए 'W' डिफ़ॉल्ट 0 (अक्षम) है

4.1.1.15 लॉगिंग

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
log_type	console	String	नहीं	लॉगिंग प्रकार (console, file, syslog)

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
log_level	info	String	नहीं	लॉग स्तर, FATAL , ERROR , WARN , INFO , और DEBUG में से कोई भी (जो DEBUG1 से लेकर DEBUG5 तक अधिक विशिष्ट हो सकता है।) 5 से अधिक डिबग स्तर को DEBUG5 में सेट किया जाएगा। यदि मान्य नहीं है, तो INFO सेट किया जाएगा।
log_path	pgmoneta.log	String	नहीं	लॉग फ़ाइल स्थान। यह एक strftime(3) संगत स्ट्रिंग हो सकती है।
log_rotation_age	0	String	नहीं	वह आयु जो लॉग फ़ाइल के रोटेशन को ट्रिगर करेगी। यदि सकारात्मक संख्या के रूप में व्यक्त किया गया है, तो इसे सेकंड के रूप में प्रबंधित किया जाएगा। सपोर्टेड प्रत्यय: 'S' (सेकंड, डिफ़ॉल्ट), 'M' (मिनट), 'H' (घंटे), 'D' (दिन), 'W' (सप्ताह)। मान 0 रोटेशन को अक्षम कर देगा।

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
log_rotation_size	0	String	नहीं	लॉग फ़ाइल का आकार जो रोटेशन को ट्रिगर करेगा। सपोर्टेड प्रत्यय: 'B' (बाइट्स), यदि प्रत्यय को छोड़ा गया है तो डिफ़ॉल्ट। 'K' या 'KB' (किलोबाइट्स), 'M' या 'MB' (मेगाबाइट्स), 'G' या 'GB' (गीगाबाइट्स)। मान 0 (प्रत्यय के साथ या बिना) इसे अक्षम कर देगा।
log_line_prefix	%Y-%m-%d %H:%M:%S	String	नहीं	हर लॉग पंक्ति के लिए एक strftime(3) संगत स्ट्रिंग को पूर्वसूचक के रूप में उपयोग करने के लिए। यदि इसमें अंतराल (स्पेस) है तो इसे उद्धरण में होना चाहिए।
log_mode	append	String	नहीं	लॉग फ़ाइल में जोड़े या इसे बनाएँ (append, create)

4.1.1.16 परिवहन स्तर सुरक्षा

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
tls	off	Boolean	नहीं	ट्रांसपोर्ट लेयर सुरक्षा (TLS) सक्रिय करें
tls_cert_file		String	नहीं	TLS के लिए प्रमाणपत्र फ़ाइल। यह फ़ाइल pgmoneta या root द्वारा चलाने वाले उपयोगकर्ता की होनी चाहिए।
tls_key_file		String	नहीं	TLS के लिए निजी कुंजी फ़ाइल। यह फ़ाइल pgmoneta या root द्वारा चलाने वाले उपयोगकर्ता की होनी चाहिए। साथ ही, जब root द्वारा मालिक हो तो इसकी अनुमति कम से कम 0640 और अन्यथा 0600 होनी चाहिए।
tls_ca_file		String	नहीं	TLS के लिए प्रमाणपत्र प्राधिकरण (CA) फ़ाइल। यह फ़ाइल pgmoneta या root द्वारा चलाने वाले उपयोगकर्ता की होनी चाहिए।

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
libev	auto	String	नहीं	libev बैकएंड का चयन करें। वैध विकल्प: auto, select, poll, epoll, iouring, devpoll, और port

4.1.1.17 विविध

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
backup_max_rate	0	int	नहीं	बैकअप दर को सीमित करने के लिए प्रति सेकंड जोड़े गए टोकन के बाइट्स की संख्या
net-work_max_rate	0	int	नहीं	नेटवर्क बैकअप दर को सीमित करने के लिए प्रति सेकंड जोड़े गए टोकन के बाइट्स की संख्या
blocking_timeout	30	int	नहीं	कनेक्शन के लिए प्रक्रिया के ब्लॉक होने का समय (अक्षम = 0)
keep_alive	on	boolean	नहीं	सॉकेट्स पर SO_KEEPALIVE
nodelay	on	boolean	नहीं	सॉकेट्स पर TCP_NODELAY चालू रखें
non_blocking	on	boolean	नहीं	सॉकेट्स पर O_NONBLOCK चालू रखें
backlog	16	int	नहीं	listen() के लिए बैकलॉग। न्यूनतम 16
hugepage	try	string	नहीं	ह्यूज पेज समर्थन (off, try, on)

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
pidfile		string	नहीं	PIDफ़ाइल का पथ। यदि निर्दिष्ट नहीं है, तो यह स्वचालित रूप से <code>unix_socket_dir/pgmoneta.<host>.pid</code> पर सेट हो जाएगा, जहां <code><host></code> <code>host</code> पैरामीटर का मान है या <code>all</code> है यदि <code>host = *</code> है।

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
update_process_title	<code>verbose</code>	string	नहीं	ऑपरेटिंग सिस्टम प्रक्रिया शीर्षक को अपडेट करने का व्यवहार। अनुमत सेटिंग्स: <code>never</code> (या <code>off</code>), प्रक्रिया शीर्षक को अपडेट नहीं करता है; <code>strict</code> प्रारंभिक प्रक्रिया शीर्षक की मौजूदा लंबाई को ओवरराइड किए बिना प्रक्रिया शीर्षक सेट करने के लिए; <code>minimal</code> बेस विवरण के लिए प्रक्रिया शीर्षक सेट करने के लिए; <code>verbose</code> (या <code>full</code>) पूर्ण विवरण के लिए प्रक्रिया शीर्षक सेट करने के लिए। ध्यान दें कि <code>strict</code> और <code>minimal</code> केवल उन्हीं सिस्टम पर मान्य हैं जो प्रक्रिया शीर्षक सेट करने के लिए एक देशी तरीका प्रदान नहीं करते। अन्य सिस्टम पर <code>strict</code> और <code>minimal</code> के बीच कोई अंतर नहीं है और डिफ़ॉल्ट व्यवहार <code>minimal</code> माना जाता है, भले ही <code>strict</code> का उपयोग किया गया हो। <code>never</code> और <code>verbose</code> सभी सिस्टम पर हमेशा मान्य होते हैं। Linux सिस्टम पर प्रक्रिया शीर्षक हमेशा 255 अक्षरों तक सीमित होता है, जबकि सिस्टम जो प्रक्रिया शीर्षक सेट करने के लिए देशी तरीका प्रदान करते हैं, वहां यह लंबा हो सकता है।

4.1.2 सर्वर अनुभाग

4.1.2.1 सर्वर

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
host		string	हाँ	PostgreSQL इंस्टेंस का पता
port		int	हाँ	PostgreSQL इंस्टेंस की पोर्ट
user		string	हाँ	प्रतिकृति उपयोगकर्ता का नाम
wal_slot		string	हाँ	WAL के लिए प्रतिकृति स्लॉट

उपयोगकर्ता को **REPLICATION** विकल्प के साथ प्रतिकृति के लिए सक्रिय किया जाना चाहिए ताकि Write-Ahead Log (WAL) स्ट्रीम किया जा सके, और उसे **postgres** डेटाबेस तक पहुंच होनी चाहिए ताकि आवश्यक कॉन्फिगरेशन पैरामीटर प्राप्त किए जा सकें।

4.1.2.2 स्लॉट प्रबंधन

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
create_slot	नहीं	boolean	नहीं	इस सर्वर के लिए प्रतिकृति स्लॉट बनाएँ। वैध मान: हाँ, नहीं

4.1.2.3 पालन

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
follow		string	नहीं	यदि पालन सर्वर विफल हो तो इस सर्वर पर फेलओवर करें

4.1.2.4 रिटेंशन

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
retention		array	नहीं	सर्वर के लिए रिटेंशन दिनों, सप्ताहों, महीनों, वर्षों के हिसाब से

4.1.2.5 WAL शिपिंग

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
wal_shipping		string	नहीं	WAL शिपिंग डायरेक्टरी

4.1.2.6 हॉट स्टैंडबाई

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
hot_standby		string	नहीं	हॉट स्टैंडबाई डायरेक्टरी
hot_standby_overrides		string	नहीं	हॉट स्टैंडबाई डायरेक्टरी में ओवरराइड करने के लिए फाइलें

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
hot_standby_tablespace		string	नहीं	हॉट स्टैंडबाई के लिए टेबलस्पेस मैपिंग्स। स्वरूप [from -> to, ?] +

4.1.2.7 कार्यकर्ता

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
workers	-1	int	नहीं	प्रत्येक प्रोसेस के लिए कार्यों के लिए उपयोग किए जा सकने वाले कार्यकर्ताओं की संख्या। 0 को अक्षम करने के लिए, -1 का अर्थ है वैश्विक सेटिंग का उपयोग करना

4.1.2.8 परिवहन स्तर सुरक्षा

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
tls_cert_file		string	नहीं	TLS के लिए प्रमाणपत्र फ़ाइल। यह फ़ाइल pgmoneta या root द्वारा चलाने वाले उपयोगकर्ता की होनी चाहिए।

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
tls_key_file		string	नहीं	TLS के लिए निजी कुंजी फ़ाइल। यह फ़ाइल pgmoneta या root द्वारा चलाने वाले उपयोगकर्ता की होनी चाहिए। साथ ही, जब root द्वारा मालिक हो तो इसकी अनुमति कम से कम 0640 और अन्यथा 0600 होनी चाहिए।
tls_ca_file		string	नहीं	TLS के लिए प्रमाणपत्र प्राधिकरण (CA) फ़ाइल। यह फ़ाइल pgmoneta या root द्वारा चलाने वाले उपयोगकर्ता की होनी चाहिए।

4.1.2.9 विविध

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
backup_max_rate	-1	int	नहीं	बैकअप दर को सीमित करने के लिए हर एक सेकंड में जोड़ी जाने वाली बाइट्स की संख्या। इसे अक्षम करने के लिए 0 का उपयोग करें, -1 का मतलब है वैश्विक सेटिंग का उपयोग करें।
network_max_rate	-1	int	नहीं	नेटवर्क बैकअप दर को सीमित करने के लिए हर एक सेकंड में जोड़ी जाने वाली बाइट्स की संख्या। इसे अक्षम करने के लिए 0 का उपयोग करें, -1 का मतलब है वैश्विक सेटिंग का उपयोग करें।

4.1.2.10 अतिरिक्त

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
extra		string	नहीं	सर्वर साइड पर पुनः प्राप्त के लिए स्रोत निर्देशिका (विवरण अतिरिक्त अनुभाग में दिया गया है)

extra कॉन्फ़िगरेशन को सर्वर सेक्शन में सेट किया जाता है। यह आवश्यक नहीं है, लेकिन यदि आप इस पैरामीटर को कॉन्फ़िगर करते हैं, तो जब आप CLI का उपयोग करके बैकअप करते हैं, जैसे कि `pgmoneta-cli -c pgmoneta.conf backup primary`, तो यह सर्वर साइड पर निर्दिष्ट सभी फाइलों को कॉपी करेगा और उन्हें क्लाइंट साइड पर भेज देगा।

यह `extra` सुविधा सर्वर साइड पर `pgmoneta_ext` एक्सटेंशन इंस्टॉल करने की आवश्यकता रखती है और उपयोगकर्ता `repl` को बनाना पड़ता है (भविष्य में इसे सुधारने की योजना है)। वर्तमान में, यह सुविधा केवल `SUPERUSER` भूमिका के लिए उपलब्ध है।

आप README का अनुसरण करके आसानी से `pgmoneta_ext` सेटअप कर सकते हैं। इसके अलावा, DEVELOPERS दस्तावेज़ में अधिक विस्तृत निर्देश उपलब्ध हैं।

`extra` पैरामीटर का फॉर्मेट फ़ाइल या निर्देशिका का पथ है। आप कई फ़ाइलों या निर्देशिकाओं को कॉमा से अलग करके सूचीबद्ध कर सकते हैं। इसका प्रारूप निम्नलिखित है:

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

4.1.3 pgmoneta_users.conf

`pgmoneta_users` कॉन्फ़िगरेशन सिस्टम में ज्ञात उपयोगकर्ताओं को परिभाषित करता है। यह फ़ाइल `pgmoneta-admin` टूल के माध्यम से बनाई और प्रबंधित की जाती है।

कॉन्फ़िगरेशन या तो `-u` फ़्लैग द्वारा निर्दिष्ट पथ से या `/etc/pgmoneta/pgmoneta_users.conf` से लोड होती है।

4.1.4 pgmoneta_admins.conf

`pgmoneta_admins` कॉन्फ़िगरेशन सिस्टम में ज्ञात प्रशासकों को परिभाषित करता है। यह फ़ाइल `pgmoneta-admin` टूल के माध्यम से बनाई और प्रबंधित की जाती है।

कॉन्फ़िगरेशन को या तो `-A` फ़्लैग द्वारा निर्दिष्ट पथ से या `/etc/pgmoneta/pgmoneta_admins.conf` से लोड किया जाता है।

यदि `pgmoneta` में Transport Layer Security (TLS) और `management` सक्षम है, तो `pgmoneta-cli` TLS का उपयोग करके कनेक्ट कर सकता है। इसके लिए फाइलें `~/.pgmoneta/pgmoneta.key` (अनुमति 0600 होनी चाहिए), `~/.pgmoneta/pgmoneta.crt` और `~/.pgmoneta/root.crt` का उपयोग करना आवश्यक है।

5 बैकअप

5.1 पूर्ण बैकअप बनाएं

हम निम्नलिखित कमांड का उपयोग करके प्राइमरी से पूर्ण बैकअप ले सकते हैं:

```
pgmoneta-cli backup primary
```

और आपको कुछ इस प्रकार का आउटपुट मिलेगा:

```
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 बैकअप देखना

हम निम्नलिखित कमांड के साथ एक सर्वर के सभी बैकअप्स को सूचीबद्ध कर सकते हैं:

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

और आपको कुछ इस प्रकार का आउटपुट मिलेगा:

```
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 वृद्धिशील बैकअप बनाएं

हम निम्नलिखित कमांड का उपयोग करके प्राइमरी से वृद्धिशील (incremental) बैकअप ले सकते हैं:

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

और आपको निम्नलिखित आउटपुट मिलेगा:

```
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
```

वृद्धिशील बैकअप केवल PostgreSQL 17+ में समर्थित है। ध्यान दें कि वर्तमान में वृद्धिशील बैकअप के लिए शाखाएं (branching) अनुमति नहीं दी गई हैं — एक बैकअप में अधिकतम 1 वृद्धिशील बैकअप चाइल्ड हो सकता है।

5.4 बैकअप देखें

हम किसी सर्वर के लिए सभी बैकअप को सूचीबद्ध करने के लिए निम्नलिखित कमांड का उपयोग कर सकते हैं:

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

और आपको निम्नलिखित आउटपुट मिलेगा:

```
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
```

5.5 बैकअप जानकारी

आप बैकअप के बारे में जानकारी सूचीबद्ध कर सकते हैं:

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

और आपको कुछ इस प्रकार का आउटपुट मिलेगा:

```
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: 00000000100000000000000000004F
```

5.6 क्रॉन्टैब बनाना

आइए हम एक `crontab` बनाते हैं ताकि हर दिन बैकअप लिया जा सके:

```
crontab -e
```

और निम्नलिखित को जोड़ें:

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

यह हर दिन सुबह 6 बजे बैकअप लेने के लिए है।

5.7 बैकअप अखंडता सत्यापित करें

pgmoneta बैकअप रूट निर्देशिका में प्रत्येक बैकअप के लिए एक SHA512 चेकसम फ़ाइल (`backup.sha512`) बनाता है, जिसका उपयोग फ़ाइलों की अखंडता को सत्यापित करने के लिए किया जा सकता है।

`sha512sum` का उपयोग करना:

```
cd <path-to-specific-backup-directory>

sha512sum --check backup.sha512
```

`verification` पैरामीटर का उपयोग यह नियंत्रित करने के लिए किया जा सकता है कि pgmoneta कितनी बार बैकअप फ़ाइलों की अखंडता को सत्यापित करता है। आप इसे `pgmoneta.conf` में कॉन्फ़िगर कर सकते हैं:

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

उदाहरण के लिए, `verification = 3600` या `verification = 1H` सेट करने पर हर घंटे अखंडता जांच की जाएगी।

6 रिटेंशन पॉलिसी

रिटेंशन पॉलिसी यह निर्धारित करती है कि एक बैकअप को कितने समय तक रखा जाना चाहिए।

6.1 रिटेंशन कॉन्फ़िगरेशन

कॉन्फ़िगरेशन मुख्य कॉन्फ़िगरेशन सेक्शन में की जाती है, या सर्वर के आधार पर भी की जा सकती है:

प्रॉपर्टी	डिफ़ॉल्ट	यूनिट	आवश्यक	विवरण
retention	7, -, -, -	Array	नहीं	बैकअप का रिटेंशन समय दिन, सप्ताह, महीने, वर्षों में

यह यह बताता है कि डिफ़ॉल्ट रूप से बैकअप्स को 7 दिनों तक रखा जाता है।

6.2 रिटेंशन को परिभाषित करना

रिटेंशन पॉलिसी को परिभाषित करना बहुत महत्वपूर्ण है क्योंकि यह यह निर्धारित करता है कि आप बैकअप्स से सिस्टम को कैसे बहाल करेंगे।

मुख्य बात यह है कि आपको अपनी पॉलिसी तय करनी है, उदाहरण के लिए:

```
7, 4, 12, 5
```

इसका मतलब है कि बैकअप्स को 7 दिन तक रखा जाए, हर सोमवार को 4 हफ्तों के लिए एक बैकअप, प्रत्येक महीने के लिए एक बैकअप, और 5 वार्षिक बैकअप्स।

यह एक कठिन प्रॉपर्टी है जिसे कॉन्फ़िगर करना, क्योंकि यह डेटाबेस क्लस्टर के आकार पर निर्भर करता है और इस प्रकार बैकअप्स का आकार कितना बड़ा होगा।

यदि आप नवीनतम बैकअप और Write-Ahead Log (WAL) से सिस्टम को पुनर्स्थापित करना चाहते हैं, तो डिफ़ॉल्ट **pgmoneta** पॉलिसी शायद पर्याप्त हो सकती है।

ध्यान दें कि यदि किसी बैकअप में कोई वृद्धिशील (incremental) बैकअप चाइल्ड है जो उस पर निर्भर करता है, तो उसे बनाए रखा जाएगा, भले ही वह रिटेंशन नीति (retention policy) के अंतर्गत न आता हो। हम भविष्य के संस्करणों में वृद्धिशील बैकअप को हटाने का समर्थन करेंगे।

6.3 रिटेंशन चेक

रिटेंशन चेक हर 5 मिनट में चलता है, और प्रत्येक रन में एक बैकअप को हटा देगा।

उन्नत बैकअप/पुनर्स्थापन pgmoneta के साथ

आप इसे हर 30 मिनट में बदल सकते हैं:

```
retention_interval = 1800
```

यह [pgmoneta] कॉन्फ़िगरेशन के तहत किया जा सकता है।

7 बैकअप रखना

7.1 बैकअप्स की सूची

पहले, हम अपने वर्तमान बैकअप्स की सूची निम्नलिखित कमांड का उपयोग करके देख सकते हैं:

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

आपको इस प्रकार का आउटपुट मिलेगा:

```
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
```

जैसा कि आप देख सकते हैं, बैकअप 20241012091219 में **Keep** फ्लैग **false** है।

7.2 बैकअप को रखना

अब, यदि आप बैकअप को रखना चाहते हैं, ताकि वह रिटेंशन पॉलिसी द्वारा डिलीट न हो, तो आप निम्नलिखित कमांड का उपयोग कर सकते हैं:

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

और आपको इस प्रकार का आउटपुट मिलेगा:


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

अब आप देख सकते हैं कि बैकअप का **Keep** फ्लैग **true** हो गया है।

ध्यान दें कि **Keep** वर्तमान में केवल पूर्ण बैकअप पर कार्य करता है। हम भविष्य के संस्करणों में वृद्धिशील बैकअप को बनाए रखने का समर्थन करेंगे।

7.3 बैकअप का वर्णन करना

अब, आप अपने बैकअप में विवरण जोड़ना चाह सकते हैं। जैसा कि आप देख सकते हैं:

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

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

यहां एक `Comments` फ़ील्ड है जहाँ आप विवरण जोड़ सकते हैं।

आप कमांड का उपयोग कर सकते हैं:

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

जिससे आपको निम्नलिखित आउटपुट मिलेगा:

```
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: 0000000010000000000000000002
```

जैसा कि आप देख सकते हैं, **Comments** फ़ील्ड में **Type** कुंजी के साथ विवरण है।

annotate कमांड में **add**, **update**, और **remove** कमांड्स होते हैं, जो **Comments** फ़ील्ड को संशोधित करने के लिए उपयोग किए जाते हैं।

7.4 बैकअप को फिर से रिटेंशन में डालना

जब आपको किसी बैकअप की अब आवश्यकता न हो, तो आप उसे पुनः रिटेंशन में डाल सकते हैं:

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

जिससे आपको निम्नलिखित आउटपुट मिलेगा:

```
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
```

अब, **Keep** फ़्लैग फिर से **false** हो गया है।

8 पुनर्स्थापना

8.1 बैकअप पुनर्स्थापित करना

हम निम्नलिखित कमांड का उपयोग करके प्राथमिक से बैकअप पुनर्स्थापित कर सकते हैं:

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

जहां:

- **current** का मतलब है Write-Ahead Log (WAL) की कॉपी करना और पहले स्थिर चेकपॉइंट पर पुनर्स्थापना करना।
- **name=X** का मतलब है Write-Ahead Log (WAL) की कॉपी करना और निर्दिष्ट लेबल पर पुनर्स्थापना करना।
- **xid=X** का मतलब है Write-Ahead Log (WAL) की कॉपी करना और निर्दिष्ट XID पर पुनर्स्थापना करना।
- **time=X** का मतलब है Write-Ahead Log (WAL) की कॉपी करना और निर्दिष्ट टाइमस्टैम्प पर पुनर्स्थापना करना।
- **lsn=X** का मतलब है Write-Ahead Log (WAL) की कॉपी करना और निर्दिष्ट Log Sequence Number (LSN) पर पुनर्स्थापना करना।
- **inclusive=X** का मतलब है कि पुनर्स्थापना निर्दिष्ट जानकारी को शामिल करती है।
- **timeline=X** का मतलब है कि पुनर्स्थापना निर्दिष्ट जानकारी टाइमलाइन पर की जाती है।
- **action=X** का मतलब है पुनर्स्थापना के बाद कौन सा क्रिया निष्पादित की जानी चाहिए। (pause, shutdown)
- **primary** का मतलब है कि क्लस्टर प्राथमिक के रूप में सेटअप है।
- **replica** का मतलब है कि क्लस्टर को एक प्रतिकृति के रूप में सेटअप किया गया है।

More information

इसके बाद, आपको इस प्रकार का आउटपुट मिलेगा:

```
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
```

यह कमांड नवीनतम बैकअप और सभी Write-Ahead Log (WAL) सेगमेंट्स को लेकर `/tmp/primary-20240928065644` डायरेक्टरी में एक अद्यतन प्रति पुनर्स्थापित करेगा।

9 प्रोमैथियस मेट्रिक्स

pgmoneta Prometheus मेट्रिक्स का समर्थन करता है।

हमने मूल कंफिगरेशन में प्रोमैथियस मेट्रिक्स को सक्रिय किया है, निम्नलिखित सेटिंग द्वारा:

```
metrics = 5001
```

`pgmoneta.conf` में।

9.1 प्रोमैथियस मेट्रिक्स तक पहुँच

अब आप मेट्रिक्स तक इस URL के माध्यम से पहुँच सकते हैं:

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

10 प्रशासनिक पहुँच

आप **pgmoneta** को एक रिमोट मशीन से एक्सेस कर सकते हैं यदि आप एक्सेस सक्षम करते हैं।

10.1 कंफिगरेशन

पहले, आपको रिमोट एक्सेस सक्षम करने के लिए निम्नलिखित सेटिंग जोड़नी होगी:

```
management = 5002
```

यह सेटिंग `pgmoneta.conf` में `[pgmoneta]` सेक्शन में जोड़ें।

10.2 व्यवस्थापक

इसके बाद, आपको `pgmoneta_admins.conf` में एक या अधिक व्यवस्थापक जोड़ने होंगे, निम्नलिखित कमांड द्वारा:

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

उदाहरण के लिए, `admin` उपयोगकर्ता नाम और `secretpassword` पासवर्ड के साथ।

10.3 pgmoneta को पुनः प्रारंभ करें

आपको **pgmoneta** को पुनः प्रारंभ करना होगा ताकि परिवर्तनों का प्रभाव पड़े।

10.4 pgmoneta से कनेक्ट करें

फिर आप `pgmoneta-cli` टूल का उपयोग करके **pgmoneta** को इस तरह एक्सेस कर सकते हैं:

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

यह कमांड आपको पासवर्ड दर्ज करने के बाद `status` कमांड को निष्पादित करेगा।

11 शटडाउन

आप **pgmoneta** की स्थिति का परीक्षण कर सकते हैं और इसे लोकल या रिमोट मशीन से शटडाउन कर सकते हैं।

11.1 पिंग

यह जांचने के लिए कि **pgmoneta** चल रहा है या नहीं, आप निम्नलिखित कमांड का उपयोग कर सकते हैं:

```
pgmoneta-cli ping
```

और आउटपुट जांचें।

11.2 शटडाउन

आप **pgmoneta** को शटडाउन करने के लिए निम्नलिखित कमांड का उपयोग कर सकते हैं:

```
pgmoneta-cli shutdown
```

और आउटपुट जांचें।

12 Docker के साथ pgmoneta निष्पादित करना

आप इसे मैन्युअल रूप से संकलित करने के बजाय डॉकर का उपयोग करके **pgmoneta** निष्पादित कर सकते हैं।

12.1 पूर्वापेक्षाएँ

- **[Docker]**[docker] या **[Podman]**[podman] को उस सर्वर पर स्थापित किया जाना चाहिए जहाँ यह PostgreSQL चला रहा है।
- सुनिश्चित करें कि PostgreSQL बाहरी कनेक्शन की अनुमति देने के लिए कॉन्फ़िगर किया गया है।

12.2 यदि आवश्यक हो तो कॉन्फ़िगरेशन फ़ाइल को अपडेट करें:

```
[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 चरण 1: PostgreSQL के लिए बाहरी पहुँच सक्षम करें

बाहर से कनेक्शन की अनुमति देने के लिए PostgreSQL स्थानीय सर्वर के `postgresql.conf` को संशोधित करें:

```
listen_addresses = '*'
```

`pg_hba.conf` को दूरस्थ कनेक्शन की अनुमति देने के लिए अद्यतन करें:

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

उन्नत बैकअप/पुनर्स्थापन pgmoneta के साथ

फिर, परिवर्तनों को प्रभावी बनाने के लिए PostgreSQL को पुनरारंभ करें:

```
sudo systemctl restart postgresql
```

12.4 चरण 2: रिपॉजिटरी को क्लोन करें

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

12.5 चरण 3: डॉकर इमेज बनाएं

दो डॉकरफाइल उपलब्ध हैं:

1. Alpine इमेज

Docker का उपयोग करके

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

Podman का उपयोग करके

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

2. Rocky Linux 9 इमेज

Docker का उपयोग करके

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

Podman का उपयोग करके

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

12.6 चरण 4: pgmoneta को डॉकर कंटेनर के रूप में चलाएँ

एक बार इमेज बन जाने पर, कंटेनर को निम्न का उपयोग करके चलाएँ:

• Docker का उपयोग करके

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

• Podman का उपयोग करके

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

12.7 चरण 5: कंटेनर को सत्यापित करें

जाँचें कि कंटेनर चल रहा है या नहीं:

- **Docker** का उपयोग करके

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

- **Podman** का उपयोग करके

```
podman ps | grep pgmoneta
```

किसी भी त्रुटि के लिए लॉग की जाँच करें:

- **Docker** का उपयोग करके

```
docker logs pgmoneta
```

- **Podman** का उपयोग करके

```
podman logs pgmoneta
```

आप उजागर मीट्रिक्स का निरीक्षण यहां भी कर सकते हैं:

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

आप निम्न का उपयोग करके कंटेनर को बंद कर सकते हैं

- **Docker** का उपयोग करके

```
docker stop pgmoneta
```

- **Podman** का उपयोग करके

```
podman stop pgmoneta
```

आप कंटेनर में exec कर सकते हैं और cli कमांड को चला सकते हैं

```
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
```

अधिक cli कमांड के लिए यह देखें।

आप `/usr/local/bin` पर तीनों बाइनरी तक पहुँच सकते हैं।

13 समस्या निवारण

13.1 सर्वर के लिए संस्करण प्राप्त नहीं कर सके

यदि आपको स्टार्टअप के दौरान **FATAL** संदेश मिलता है, तो अपने PostgreSQL लॉगिन्स की जांच करें:

```
psql postgres
```

और

```
psql -U repl postgres
```

फिर, PostgreSQL लॉग्स में किसी भी त्रुटि की जांच करें।

अगर फिर भी समस्या बनी रहती है, तो `pgmoneta.conf` में `log_level` को `DEBUG5` पर सेट करने से त्रुटि के बारे में अधिक जानकारी मिल सकती है।

14 आभार

14.1 लेखक

pgmoneta को निम्नलिखित लेखकों द्वारा बनाया गया है:

```
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 कमिटर्स

```
Jesper Pedersen <jesper.pedersen@comcast.net>
Haoran Zhang <andrewzhr9911@gmail.com>
```

14.3 योगदान

pgmoneta में योगदान GitHub पर प्रबंधित किए जाते हैं।

- प्रश्न पूछें
- समस्या उठाएं
- विशेषता अनुरोध
- कोड सबमिशन

योगदान का स्वागत है!

कृपया हमारी Code of Conduct नीतियों का पालन करें जब आप हमारे समुदाय के साथ संवाद करें।

यदि आपको यह प्रोजेक्ट उपयोगी लगे, तो GitHub पर इसे एक स्टार दें। और, प्रोजेक्ट को X पर फॉलो करने के लिए स्वतंत्र महसूस करें।

15 लाइसेंस

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 license

15.1 libart

हमारा अनुकूलनशील रेडिक्स ट्री (ART) कार्यान्वयन आधारित है The Adaptive Radix Tree: ARTful Indexing for Main-Memory DatabasesART_paper और libartlibart पर जो 3-BSD license के तहत है।

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.