



Miria PostgreSQL Cluster Using Patroni And HAProxy Documentation

Miria 4.0

Publication Number: MIRIA-POST-PDF-EN-0223-REV1

Publication Date: February 2023





©2023 Atempo SAS. All rights reserved.

All names and products contained herein are the trademarks or registered trademarks of their respective holders.

The information contained herein is the confidential and proprietary information of Atempo SAS. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Atempo SAS.

Atempo SAS
23 Avenue Carnot
91300 Massy - France

Contents

CHAPTER 1 - Introduction	2
Schema	2
Architecture	3
CHAPTER 2 - Installation	4
Install PostgreSQL 13	4
Install Patroni	6
Install ETCD	8
Install HAProxy	12
CHAPTER 3 - Configuration	14
Configure ETCD	14
Configure Patroni	15
Configure HAProxy	22
On Miria Server	25
On one of PostgreSQL node	25
On Other PostgreSQL Nodes	27
On Miria Server	28

CHAPTER 1 - Introduction

This article is a step-by-step guide on how to create a Miria highly available PostgreSQL cluster architecture using Patroni and HAProxy.

Patroni is an open-source python package that manages Postgres configuration. It can be configured to handle tasks like replication, backups, and restorations.

EtcD is a fault-tolerant, distributed key-value store used to store the state of the Postgres cluster. Using Patroni, all of the Postgres nodes make use of etcd to keep the Postgres cluster up and running. In production, it makes sense to use a larger etcd cluster so that if one etcd node fails, it doesn't affect Postgres servers.

With the Postgres cluster set up, we need a method to connect to the master regardless of which of the servers in the cluster is the master. This is where HAProxy steps in. All Postgres clients/applications will connect to the HAProxy which will provide a connection to the master node in the cluster.

HAProxy is an open-source, high-performance load balancer and reverse proxy for TCP and HTTP applications. HAProxy can be used to distribute loads and improve the performance of websites and applications.

Schema

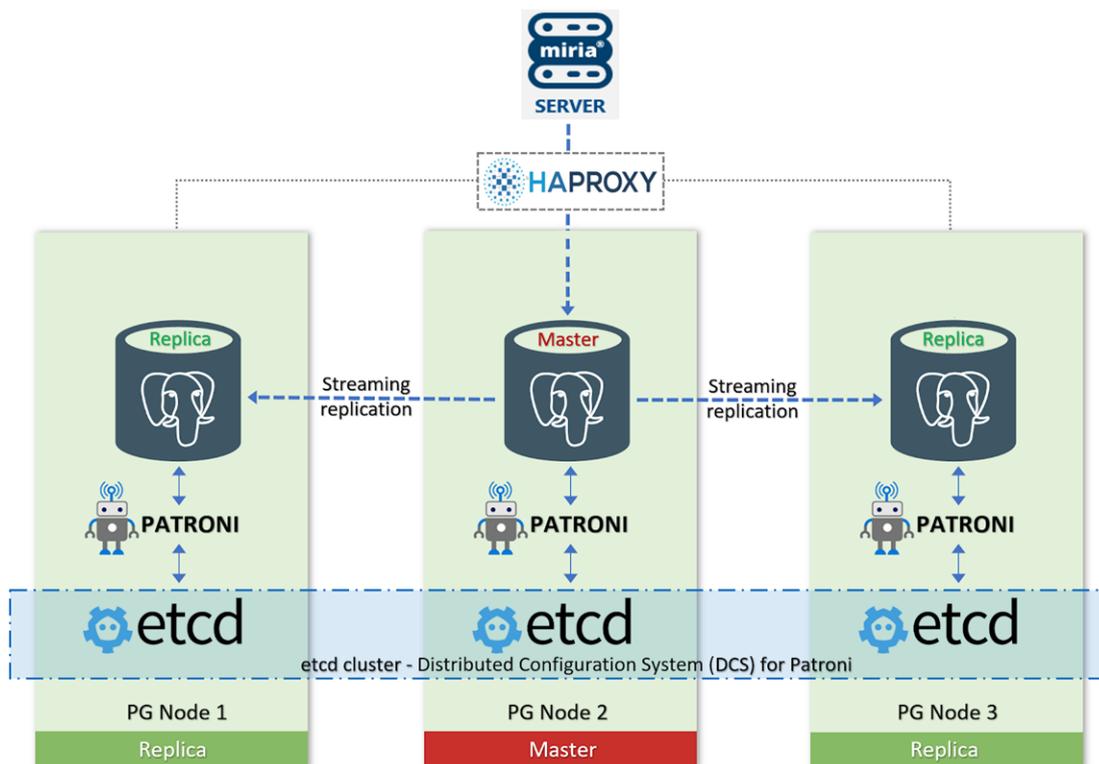


Figure 1: Miria PostgreSQL cluster architecture using Patroni and HAProxy

Architecture

OS release : Centos 8

PostgreSQL release : 13.9

ETCD release : 3.5.7

HAProxy release : 1.8.27

In our example, etcd and HAProxy will be located on the same server.

- postgresql0 : 172.18.224.64
- postgresql1 : 172.18.222.8
- etcdhaproxy : 172.18.224.27
- miriaserver: 172.18.224.166

Before you start the installation ensure you're running the latest release of CentOS 8 Linux system. You can run the yum update command to pull the latest updates.

- yum update
- systemctl reboot

CHAPTER 2 - Installation

Install PostgreSQL 13

The following commands must be executed on both PostgreSQL nodes (`postgresq10` & `postgresq11` in our example).

Run the following commands to add the repository that provides PostgreSQL packages to your CentOS 8 server:

```
[root@postgresq10 ~]# dnf install https://download.postgresql.org/pub/repos/yum/repoprms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
CentOS Stream 8 - AppStream                15 kB/s | 4.4 kB    00:00
CentOS Stream 8 - AppStream                8.9 MB/s | 27 MB   00:03
CentOS Stream 8 - BaseOS                   787 B/s | 3.9 kB   00:05
CentOS Stream 8 - BaseOS                   5.1 MB/s | 27 MB   00:05
CentOS Stream 8 - Extras                   12 kB/s | 2.9 kB   00:00
CentOS Stream 8 - Extras common packages   1.1 kB/s | 3.0 kB   00:02
pgdg-redhat-repo-latest.noarch.rpm        30 kB/s | 13 kB    00:00
Dependencies resolved.

=====
Package                               Architecture      Version           Repository         Size
=====
Installing:
pgdg-redhat-repo                       noarch            42.0-32           @commandline       13 k
=====

Transaction Summary
=====
Install 1 Package

Total size: 13 k
Installed size: 14 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                     1/1
  Installing     : pgdg-redhat-repo-42.0-32.noarch    1/1
  Verifying     : pgdg-redhat-repo-42.0-32.noarch    1/1

Installed:
pgdg-redhat-repo-42.0-32.noarch

Complete!
[root@postgresq10 ~]#
```

We need to disable the PostgreSQL AppStream repository on CentOS 8 Linux which contains some other version of PostgreSQL.

```
[root@postgresq10 ~]# dnf -y module disable postgresql
PostgreSQL common RPMs for RHEL / Rocky 8 - x86_64      133 B/s | 195 B    00:01
PostgreSQL common RPMs for RHEL / Rocky 8 - x86_64      1.6 MB/s | 1.7 kB  00:00
Importing GPG key 0x442DF0F8:
  Userid       : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
  Fingerprint: 68C9 E2B9 1A37 D136 FE74 D176 1F16 D2E1 442D F0F8
  From         : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
PostgreSQL common RPMs for RHEL / Rocky 8 - x86_64      451 kB/s | 702 kB   00:01
PostgreSQL 15 for RHEL / Rocky 8 - x86_64              788 B/s | 195 B    00:00
PostgreSQL 15 for RHEL / Rocky 8 - x86_64              1.6 MB/s | 1.7 kB   00:00
Importing GPG key 0x442DF0F8:
  Userid       : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
  Fingerprint: 68C9 E2B9 1A37 D136 FE74 D176 1F16 D2E1 442D F0F8
  From         : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
PostgreSQL 15 for RHEL / Rocky 8 - x86_64              175 kB/s | 168 kB   00:00
PostgreSQL 14 for RHEL / Rocky 8 - x86_64              430 B/s | 195 B    00:00
PostgreSQL 14 for RHEL / Rocky 8 - x86_64              1.6 MB/s | 1.7 kB   00:00
Importing GPG key 0x442DF0F8:
  Userid       : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
  Fingerprint: 68C9 E2B9 1A37 D136 FE74 D176 1F16 D2E1 442D F0F8
  From         : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
PostgreSQL 14 for RHEL / Rocky 8 - x86_64              429 kB/s | 465 kB   00:01
PostgreSQL 13 for RHEL / Rocky 8 - x86_64              239 B/s | 195 B    00:00
PostgreSQL 13 for RHEL / Rocky 8 - x86_64              1.6 MB/s | 1.7 kB   00:00
Importing GPG key 0x442DF0F8:
```

```

Userid      : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
Fingerprint: 68C9 E2B9 1A37 D136 FE74 D176 1F16 D2E1 442D F0F8
From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
PostgreSQL 13 for RHEL / Rocky 8 - x86_64           519 kB/s | 757 kB   00:01
PostgreSQL 12 for RHEL / Rocky 8 - x86_64           333 B/s | 195 B    00:00
PostgreSQL 12 for RHEL / Rocky 8 - x86_64           1.6 MB/s | 1.7 kB   00:00
Importing GPG key 0x442DF0F8:
Userid      : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
Fingerprint: 68C9 E2B9 1A37 D136 FE74 D176 1F16 D2E1 442D F0F8
From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
PostgreSQL 12 for RHEL / Rocky 8 - x86_64           1.1 MB/s | 905 kB   00:00
PostgreSQL 11 for RHEL / Rocky 8 - x86_64           836 B/s | 195 B    00:00
PostgreSQL 11 for RHEL / Rocky 8 - x86_64           1.6 MB/s | 1.7 kB   00:00
Importing GPG key 0x442DF0F8:
Userid      : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
Fingerprint: 68C9 E2B9 1A37 D136 FE74 D176 1F16 D2E1 442D F0F8
From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
PostgreSQL 11 for RHEL / Rocky 8 - x86_64           1.0 MB/s | 1.1 MB   00:01
Last metadata expiration check: 0:00:01 ago on Mon 23 Jan 2023 04:17:29 PM CET.
Dependencies resolved.
=====
==
Package                Architecture          Version              Repository           Size
=====
Disabling modules:
postgresql

Transaction Summary
=====
==
Complete!
[root@postgresql0 ~]#

```

Now install PostgreSQL 13 packages on your CentOS 8 and various extension modules that are included in the PostgreSQL distribution.

```

[root@postgresql0 ~]# dnf install postgresql13-server postgresql13-contrib
Last metadata expiration check: 0:02:05 ago on Mon 23 Jan 2023 04:17:29 PM CET.
Dependencies resolved.
=====
Package                Architecture          Version              Repository           Size
=====
Installing:
postgresql13-contrib   x86_64                13.9-1PGDG.rhel8    pgdg13               643 k
postgresql13-server    x86_64                13.9-1PGDG.rhel8    pgdg13               5.5 M
Installing dependencies:
libc                    x86_64                60.3-2.el8_1        baseos                8.8 M
perl-Carp               noarch                1.42-396.el8        baseos                30 k
perl-Data-Dumper        x86_64                2.167-399.el8       baseos                58 k
perl-Digest             noarch                1.17-395.el8        appstream              27 k
perl-Digest-MD5         x86_64                2.55-396.el8        appstream              37 k
perl-Encode             x86_64                4:2.97-3.el8        baseos                 1.5 M
[...]
Installed:
libc-60.3-2.el8_1.x86_64          perl-Carp-1.42-396.el8.noarch
perl-Data-Dumper-2.167-399.el8.x86_64  perl-Digest-1.17-395.el8.noarch
perl-Digest-MD5-2.55-396.el8.x86_64  perl-Encode-4:2.97-3.el8.x86_64
perl-Errno-1.28-421.el8.x86_64       perl-Exporter-5.72-396.el8.noarch
perl-File-Path-2.15-2.el8.noarch      perl-File-Temp-0.230.600-1.el8.noarch
perl-Getopt-Long-1:2.50-4.el8.noarch  perl-HTTP-Tiny-0.074-1.el8.noarch
perl-IO-1.38-421.el8.x86_64          perl-IO-Socket-IP-0.39-5.el8.noarch
perl-IO-Socket-SSL-2.066-4.module_el8.4.0+517+be1595ff.noarch  perl-MIME-Base64-3.15-396.el8.x86_64
perl-Mozilla-CA-20160104-7.module_el8.3.0+416+dee7bcef.noarch  perl-Net-SSLeay-1.88-1.module_el8.4.0+517+be1595ff.x86_64
perl-PathTools-3.74-1.el8.x86_64      perl-Pod-Escapes-1:1.07-395.el8.noarch
perl-Pod-Perldoc-3.28-396.el8.noarch  perl-Pod-Simple-1:3.35-395.el8.noarch
perl-Pod-Usage-4:1.69-395.el8.noarch  perl-Scalar-List-Utils-3:1.49-2.el8.x86_64
perl-Socket-4:2.027-3.el8.x86_64     perl-Storable-1:3.11-3.el8.x86_64
perl-Term-ANSIColor-4.06-396.el8.noarch  perl-Term-Cap-1.17-395.el8.noarch
perl-Text-ParseWords-3.30-395.el8.noarch  perl-Text-Tabs+Wrap-2013.0523-395.el8.noarch
perl-Time-Local-1:1.280-1.el8.noarch  perl-URI-1.73-3.el8.noarch
perl-Unicode-Normalize-1.25-396.el8.x86_64  perl-constant-1.33-396.el8.noarch
perl-Interpreter-4:5.26.3-421.el8.x86_64  perl-libnet-3.11-3.el8.noarch
perl-libs-4:5.26.3-421.el8.x86_64       perl-macros-4:5.26.3-421.el8.x86_64
perl-parent-1:0.237-1.el8.noarch        perl-podlators-4.11-1.el8.noarch
perl-threads-1:2.21-2.el8.x86_64       perl-threads-shared-1.58-2.el8.x86_64
postgresql13-13.9-1PGDG.rhel8.x86_64  postgresql13-contrib-13.9-1PGDG.rhel8.x86_64
postgresql13-libs-13.9-1PGDG.rhel8.x86_64  postgresql13-server-13.9-1PGDG.rhel8.x86_64

Complete!
[root@postgresql0 ~]#

```

Before running the service, you must first initialize the database instance after installing PostgreSQL 13.

```
[root@postgresql ~]# /usr/pgsql-13/bin/postgresql-13-setup initdb
Initializing database ... OK
[root@postgresql ~]#
```

Start the PostgreSQL database server and check the service status to confirm it is running.

Important: PostgreSQL service must be disabled to prevent it from starting at boot. Patroni will do the rolling start for PostgreSQL.

```
[root@postgresql ~]# systemctl start postgresql-13
[root@postgresql ~]#
[root@postgresql ~]# systemctl status postgresql-13
• postgresql-13.service - PostgreSQL 13 database server
  Loaded: loaded (/usr/lib/systemd/system/postgresql-13.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-01-23 16:23:53 CET; 40s ago
  Docs: https://www.postgresql.org/docs/13/static/
  Process: 13247 ExecStartPre=/usr/pgsql-13/bin/postgresql-13-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
  Main PID: 13254 (postmaster)
  Tasks: 8 (Limit: 49443)
  Memory: 16.9M
  CGroup: /system.slice/postgresql-13.service
          └─13254 /usr/pgsql-13/bin/postmaster -D /var/lib/pgsql/13/data/
              └─13255 postgres: logger
                  └─13257 postgres: checkpointer
                      └─13258 postgres: background writer
                          └─13259 postgres: walwriter
                              └─13260 postgres: autovacuum launcher
                                  └─13261 postgres: stats collector
                                      └─13262 postgres: logical replication launcher

Jan 23 16:23:53 postgresql0 systemd[1]: Starting PostgreSQL 13 database server...
Jan 23 16:23:53 postgresql0 postmaster[13254]: 2023-01-23 16:23:53.897 CET [13254]LOG:redirecting log output to logging collector process
Jan 23 16:23:53 postgresql0 postmaster[13254]: 2023-01-23 16:23:53.897 CET [13254]HINT:Future log output will appear in directory "log".
Jan 23 16:23:53 postgresql0 systemd[1]: Started PostgreSQL 13 database server.
[root@postgresql ~]#
```

Update the PostgreSQL admin user password with a strong password.

```
[root@postgresql ~]# su - postgres
[postgres@postgresql ~]#
[postgres@postgresql ~]# psql -c "ALTER USER postgres PASSWORD 'postgres'"
ALTER ROLE
[postgres@postgresql ~]# exit
logout
[root@postgresql ~]#
```

Install Patroni

The following commands must be executed on both PostgreSQL nodes (`postgresql0` & `postgresql1` in our example).

Install pip for Python 3 and upgrade it to latest release.

```
[root@postgresql ~]# yum install python3-pip
Last metadata expiration check: 0:14:33 ago on Mon 23 Jan 2023 04:17:29 PM CET.
Dependencies resolved.
=====
Package                Architecture      Version           Repository        Size
=====
Installing:
python3-pip            noarch           9.0.3-22.el8     appstream         20 k
Installing dependencies:
python3-setuptools     noarch           39.2.0-6.el8     baseos            163 k
python36               x86_64          3.6.8-38.module_... appstream         19 k
=====
Transaction Summary
=====
Install 3 Packages

Total download size: 202 k
Installed size: 466 k
```

```

Is this ok [y/N]: y
Downloading Packages:
(1/3): python3-pip-9.0.3-22.el8.noarch.rpm                309 kB/s | 20 kB    00:00
(2/3): python36-3.6.8-38.module_el8.5.0+895+a459eca8.x86_64.rpm  280 kB/s | 19 kB    00:00
(3/3): python3-setuptools-39.2.0-6.el8.noarch.rpm          890 kB/s | 163 kB   00:00
-----
Total                                                    197 kB/s | 202 kB    00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : python3-setuptools-39.2.0-6.el8.noarch 1/3
  Installing     : python36-3.6.8-38.module_el8.5.0+895+a459eca8.x86_64 2/3
  Running scriptlet: python36-3.6.8-38.module_el8.5.0+895+a459eca8.x86_64 2/3
  Installing     : python3-pip-9.0.3-22.el8.noarch        3/3
  Running scriptlet: python3-pip-9.0.3-22.el8.noarch        3/3
  Verifying      : python3-pip-9.0.3-22.el8.noarch        1/3
  Verifying      : python36-3.6.8-38.module_el8.5.0+895+a459eca8.x86_64 2/3
  Verifying      : python3-setuptools-39.2.0-6.el8.noarch 3/3

Installed:
python3-pip-9.0.3-22.el8.noarch      python3-setuptools-39.2.0-6.el8.noarch      python36-3.6.8-38.module_el8.5.0+895+a459eca8.x86_64

Complete!
[root@postgresq10 ~]#

```

```

[root@postgresq10 ~]# python3 -m pip install --upgrade pip
WARNING: Running pip install with root privileges is generally not a good idea. Try `__main__.py install --user` instead.
Collecting pip
  Downloading
  https://files.pythonhosted.org/packages/a4/6d/6463d49a933f547439d6b5b98b46af8742cc03ae83543e4d7688c2420f8b/pip-21.3.1-py3-none-any.whl
  (1.7MB)
  100% |#####| 1.7MB 325kB/s
Installing collected packages: pip
Successfully installed pip-21.3.1
[root@postgresq10 ~]#

```

Use the PIP command to install the Patroni and other dependencies.

```

[root@postgresq10 ~]# python3 -m pip install patroni
Collecting patroni
  Downloading patroni-2.1.7-py3-none-any.whl (200 kB)
  |#####| 200 kB 4.0 MB/s
Collecting urllib3!=1.21,>=1.19.1
  Downloading urllib3-1.26.14-py2.py3-none-any.whl (140 kB)
  |#####| 140 kB 13.2 MB/s
Collecting psutil>=2.0.0
  Downloading psutil-5.9.4-cp36-abi3-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl(280 kB)
  |#####| 280 kB 11.8 MB/s
Requirement already satisfied: python-dateutil in /usr/lib/python3.6/site-packages (from patroni)(2.6.1)
Collecting click>=4.1
  Downloading click-8.0.4-py3-none-any.whl (97 kB)
  |#####| 97 kB 4.6 MB/s
Collecting PyYAML
  Downloading PyYAML-6.0-cp36-cp36m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl(603 kB)
  |#####| 603 kB 13.7 MB/s
Collecting prettytable>=0.7
  Downloading prettytable-2.5.0-py3-none-any.whl (24 kB)
Requirement already satisfied: six>=1.7 in /usr/lib/python3.6/site-packages (from patroni) (1.11.0)
Collecting ydiff>=1.2.0
  Downloading ydiff-1.2.tar.gz (42 kB)
  |#####| 42 kB 1.0 MB/s
  Preparing metadata (setup.py) ... done
Collecting importlib-metadata
  Downloading importlib_metadata-4.8.3-py3-none-any.whl (17 kB)
Collecting wcwidth
  Downloading wcwidth-0.2.6-py2.py3-none-any.whl (29 kB)
Collecting zipp>=0.5
  Downloading zipp-3.6.0-py3-none-any.whl (5.3 kB)
Collecting typing-extensions>=3.6.4
  Downloading typing_extensions-4.1.1-py3-none-any.whl (26 kB)
Using legacy 'setup.py install' for ydiff, since package 'wheel' is not installed.
Installing collected packages: zipp, typing-extensions, wcwidth, importlib-metadata, ydiff, urllib3, PyYAML, psutil, prettytable, click, patroni
  Running setup.py install for ydiff ... done
Successfully installed PyYAML-6.0 click-8.0.4 importlib-metadata-4.8.3 patroni-2.1.7 prettytable-2.5.0 psutil-5.9.4
typing-extensions-4.1.1 urllib3-1.26.14 wcwidth-0.2.6 ydiff-1.2 zipp-3.6.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[root@postgresq10 ~]#

```

```
[root@postgresql0 ~]# python3 -m pip install python-etcd
Collecting python-etcd
  Downloading python-etcd-0.4.5.tar.gz (37 kB)
    Preparing metadata (setup.py) ... done
Requirement already satisfied: urllib3>=1.7.1 in /usr/local/lib/python3.6/site-packages (from python-etcd) (1.26.14)
Collecting dnspython>=1.13.0
  Downloading dnspython-2.2.1-py3-none-any.whl (269 kB)
    |#####| 269 kB 13.8 MB/s
Using legacy 'setup.py install' for python-etcd, since package 'wheel' is not installed.
Installing collected packages: dnspython, python-etcd
  Running setup.py install for python-etcd ... done
Successfully installed dnspython-2.2.1 python-etcd-0.4.5
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[root@postgresql0 ~]#
```

```
[root@postgresql0 ~]# python3 -m pip install psycpg2-binary
Collecting psycpg2-binary
  Downloading psycpg2_binary-2.9.5-cp36-cp36m-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (3.0 MB)
    |#####| 3.0 MB 23.5 MB/s
Installing collected packages: psycpg2-binary
Successfully installed psycpg2-binary-2.9.5
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
[root@postgresql0 ~]#
```

Install ETCD

The following commands must be executed on ETCD server (**etcdhaproxy** in our example).

Ensure you have curl and wget installed on your CentOS 8.

```
[root@etcdhaproxy ~]# yum install curl wget
Last metadata expiration check: 0:31:30 ago on Mon 23 Jan 2023 04:42:53 PM CET.
Package curl-7.61.1-27.el8.x86_64 is already installed.
Dependencies resolved.
=====
Package                Architecture          Version                Repository              Size
=====
Installing:
wget                   x86_64                1.19.5-11.el8         appstream               734 k
Upgrading:
curl                   x86_64                7.61.1-28.el8         baseos                  352 k
libcurl                x86_64                7.61.1-28.el8         baseos                  302 k
Installing dependencies:
libmetalink            x86_64                0.1.3-7.el8           baseos                  32 k
=====
Transaction Summary
=====
Install 2 Packages
Upgrade 2 Packages

Total download size: 1.4 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): libmetalink-0.1.3-7.el8.x86_64.rpm                270 kB/s | 32 kB  00:00
(2/4): libcurl-7.61.1-28.el8.x86_64.rpm                 2.0 MB/s | 302 kB 00:00
(3/4): curl-7.61.1-28.el8.x86_64.rpm                   1.2 MB/s | 352 kB 00:00
(4/4): wget-1.19.5-11.el8.x86_64.rpm                   2.3 MB/s | 734 kB 00:00
-----
Total                                                    2.1 MB/s | 1.4 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  Upgrading      : libcurl-7.61.1-28.el8.x86_64          1/1
  Installing     : libmetalink-0.1.3-7.el8.x86_64       2/6
  Installing     : wget-1.19.5-11.el8.x86_64           3/6
  Running scriptlet: wget-1.19.5-11.el8.x86_64         3/6
  Upgrading      : curl-7.61.1-28.el8.x86_64           4/6
  Cleanup        : curl-7.61.1-27.el8.x86_64           5/6
  Cleanup        : libcurl-7.61.1-27.el8.x86_64        6/6
  Running scriptlet: libcurl-7.61.1-27.el8.x86_64      6/6
  Verifying      : wget-1.19.5-11.el8.x86_64           1/6
  Verifying      : libmetalink-0.1.3-7.el8.x86_64      2/6
  Verifying      : curl-7.61.1-28.el8.x86_64          3/6
  Verifying      : curl-7.61.1-27.el8.x86_64          4/6
```

```

Verifying      : libcurl-7.61.1-28.el8.x86_64           5/6
Verifying      : libcurl-7.61.1-27.el8.x86_64           6/6

Upgraded:
  curl-7.61.1-28.el8.x86_64                libcurl-7.61.1-28.el8.x86_64
Installed:
  libmetalink-0.1.3-7.el8.x86_64            wget-1.19.5-11.el8.x86_64

Complete!
[root@etcdhaproxy ~]#

```

Check the latest etcd release on releases page before you proceed to get the latest release tag.

```

[root@etcdhaproxy ~]# ETCD_RELEASE=$(curl -s https://api.github.com/repos/etcd-io/etcd/releases/latest|grep tag_name | cut -d '"' -f 4)
[root@etcdhaproxy ~]# echo $ETCD_RELEASE
v3.5.7
[root@etcdhaproxy ~]#

```

```

[root@etcdhaproxy ~]# wget https://github.com/etcd-io/etcd/releases/download/${ETCD_RELEASE}/etcd-${ETCD_RELEASE}-linux-amd64.tar.gz
--2023-01-23 17:18:02-- https://github.com/etcd-io/etcd/releases/download/v3.5.7/etcd-v3.5.7-linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/11225014/ad747a14-32c7-4acc-b4e4-d82d52e6702d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230123%2Fus-east-1%2Fs%2Faws4_request&X-Amz-Date=20230123T161803Z&X-Amz-Expires=300&X-Amz-Signature=b6fdd9ba31deeddae6c15e4e483cacc379d64bf643aac8730064592311b456e5&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=11225014&response-content-disposition=attachment%3B%20filename%3Detcd-v3.5.7-linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2023-01-23 17:18:03-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/11225014/ad747a14-32c7-4acc-b4e4-d82d52e6702d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230123%2Fus-east-1%2Fs%2Faws4_request&X-Amz-Date=20230123T161803Z&X-Amz-Expires=300&X-Amz-Signature=b6fdd9ba31deeddae6c15e4e483cacc379d64bf643aac8730064592311b456e5&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=11225014&response-content-disposition=attachment%3B%20filename%3Detcd-v3.5.7-linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18458320 (18M) [application/octet-stream]
Saving to: 'etcd-v3.5.7-linux-amd64.tar.gz'

etcd-v3.5.7-linux-amd64.tar.gz
100%[=====>] 17.60M  10.1MB/s   in 1.7s

2023-01-23 17:18:06 (10.1 MB/s) - 'etcd-v3.5.7-linux-amd64.tar.gz' saved [18458320/18458320]

[root@etcdhaproxy ~]#

```

Extract downloaded archive file and move etcd and etcdctl binary files to `/usr/local/bin` directory.

```

[root@etcdhaproxy ~]# tar xvf etcd-${ETCD_RELEASE}-linux-amd64.tar.gz
etcd-v3.5.7-linux-amd64/
etcd-v3.5.7-linux-amd64/README.md
etcd-v3.5.7-linux-amd64/READMEv2-etcdctl.md
etcd-v3.5.7-linux-amd64/etcdctl
etcd-v3.5.7-linux-amd64/etcdctl
etcd-v3.5.7-linux-amd64/Documentation/
etcd-v3.5.7-linux-amd64/Documentation/README.md
etcd-v3.5.7-linux-amd64/Documentation/dev-guide/
etcd-v3.5.7-linux-amd64/Documentation/dev-guide/apispec/
etcd-v3.5.7-linux-amd64/Documentation/dev-guide/apispec/swagger/
etcd-v3.5.7-linux-amd64/Documentation/dev-guide/apispec/swagger/v3election.swagger.json
etcd-v3.5.7-linux-amd64/Documentation/dev-guide/apispec/swagger/rpc.swagger.json
etcd-v3.5.7-linux-amd64/Documentation/dev-guide/apispec/swagger/v3lock.swagger.json
etcd-v3.5.7-linux-amd64/README-etcdctl.md
etcd-v3.5.7-linux-amd64/README-etcdctl.md
etcd-v3.5.7-linux-amd64/etcd
[root@etcdhaproxy ~]#

```

```

[root@etcdhaproxy ~]# cd etcd-${ETCD_RELEASE}-linux-amd64
[root@etcdhaproxy etcd-v3.5.7-linux-amd64]# mv etcd* /usr/local/bin
[root@etcdhaproxy etcd-v3.5.7-linux-amd64]#

```

We're going to use systemd to manage etcd service. First, create data directory for etcd.

```

[root@etcdhaproxy etcd-v3.5.7-linux-amd64]# mkdir -p /var/lib/etcd
[root@etcdhaproxy etcd-v3.5.7-linux-amd64]#

```

Create etcd system user and `set /var/lib/etcd/` directory ownership to etcd user.

```
[root@etcdhaproxy ~]# groupadd --system etcd
[root@etcdhaproxy ~]# useradd -s /sbin/nologin --system -g etcd etcd
[root@etcdhaproxy ~]# chown -R etcd:etcd /var/lib/etcd
[root@etcdhaproxy ~]# chmod 0775 /var/lib/etcd
[root@etcdhaproxy ~]#
```

Create a new systemd service file for etcd.

```
[root@etcdhaproxy ~]# vi /etc/systemd/system/etcd.service
```

Copy the following content in the file.

```
[Unit]
Description=etcd key-value store
Documentation=https://github.com/etcd-io/etcd
After=network.target

[Service]
User=etcd
Type=notify
Environment=ETCD_DATA_DIR=/var/lib/etcd
Environment=ETCD_NAME=%m
ExecStart=/usr/local/bin/etcd
Restart=always
RestartSec=10s
LimitNOFILE=40000
EnvironmentFile=-/etc/etcd/etcd.conf

[Install]
WantedBy=multi-user.target
```

Reload systemd service & start etcd service.

```
[root@etcdhaproxy ~]# systemctl daemon-reload
[root@etcdhaproxy ~]# systemctl start etcd
Job for etcd.service failed because the control process exited with error code.
See "systemctl status etcd.service" and "journalctl -xe" for details.
[root@etcdhaproxy ~]#
```

If the previous command fails (as shown above), it may be due to SELinux in enforcing mode. In this case, generate a local policy module to allow access to data directories.

```
[root@etcdhaproxy ~]# yum install policycoreutils-python-utils
Last metadata expiration check: 0:51:37 ago on Tue 24 Jan 2023 07:23:09 AM CET.
Dependencies resolved.
=====
Package Architecture Version Repository Size
=====
Installing:
policycoreutils-python-utils noarch 2.9-21.1.el8 baseos 253 k
Upgrading:
policycoreutils x86_64 2.9-21.1.el8 baseos 375 k
Installing dependencies:
checkpolicy x86_64 2.9-1.el8 baseos 348 k
python3-audit x86_64 3.0.7-4.el8 baseos 87 k
python3-libsemanage x86_64 2.9-9.el8 baseos 128 k
python3-policycoreutils noarch 2.9-21.1.el8 baseos 2.2 M
python3-setools x86_64 4.3.0-3.el8 baseos 624 k
=====
Transaction Summary
=====
Install 6 Packages
Upgrade 1 Package

Total download size: 4.0 M
Is this ok [y/N]: y
Downloading Packages:
(1/7): python3-audit-3.0.7-4.el8.x86_64.rpm 760 kB/s | 87 kB 00:00
(2/7): policycoreutils-python-utils-2.9-21.1.el8.noarch.rpm 1.7 MB/s | 253 kB 00:00
(3/7): python3-libsemanage-2.9-9.el8.x86_64.rpm 1.6 MB/s | 128 kB 00:00
(4/7): checkpolicy-2.9-1.el8.x86_64.rpm 1.6 MB/s | 348 kB 00:00
(5/7): policycoreutils-2.9-21.1.el8.x86_64.rpm 1.8 MB/s | 375 kB 00:00
(6/7): python3-setools-4.3.0-3.el8.x86_64.rpm 1.7 MB/s | 624 kB 00:00
(7/7): python3-policycoreutils-2.9-21.1.el8.noarch.rpm 2.6 MB/s | 2.2 MB 00:00
```

```

-----
Total                                                                                               3.7 MB/s | 4.0 MB   00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                                                                                          1/1
  Running scriptlet: policycoreutils-2.9-21.1.el8.x86_64                                       1/1
  Upgrading                                                                                          1/8
  Running scriptlet: policycoreutils-2.9-21.1.el8.x86_64                                       1/8
  Installing                                                                                         2/8
  Installing                                                                                         3/8
  Installing                                                                                         4/8
  Installing                                                                                         5/8
  Installing                                                                                         6/8
  Installing                                                                                         7/8
  Running scriptlet: policycoreutils-2.9-20.el8.x86_64                                          8/8
  Cleanup                                                                                           8/8
  Running scriptlet: policycoreutils-2.9-20.el8.x86_64                                          8/8
  Verifying                                                                                         1/8
  Verifying                                                                                         2/8
  Verifying                                                                                         3/8
  Verifying                                                                                         4/8
  Verifying                                                                                         5/8
  Verifying                                                                                         6/8
  Verifying                                                                                         7/8
  Verifying                                                                                         8/8

Upgraded:
  policycoreutils-2.9-21.1.el8.x86_64

Installed:
  checkpolicy-2.9-1.el8.x86_64          policycoreutils-python-utils-2.9-21.1.el8.noarch  python3-audit-3.0.7-4.el8.x86_64
  python3-libsemanage-2.9-9.el8.x86_64  python3-policycoreutils-2.9-21.1.el8.noarch       python3-setools-4.3.0-3.el8.x86_64

Complete!
[root@etcdhaproxy ~]#

```

```

[root@etcdhaproxy ~]# ausearch -c '(etcd)' --raw | audit2allow -M my-etcd
***** IMPORTANT *****
To make this policy package active, execute:

semodule -i my-etcd.pp

[root@etcdhaproxy ~]#
[root@etcdhaproxy ~]# semodule -X 300 -i my-etcd.pp
[root@etcdhaproxy ~]# restorecon -Rv /usr/local/bin/etcd
Relabeled /usr/local/bin/etcd from unconfined_u:object_r:admin_home_t:s0 to unconfined_u:object_r:bin_t:s0
[root@etcdhaproxy ~]#

```

Enable service to start when system is rebooted.

```

[root@etcdhaproxy ~]# systemctl enable etcd.service
Created symlink /etc/systemd/system/multi-user.target.wants/etcd.service → /etc/systemd/system/etcd.service.

```

Check service status to confirm it is running.

```

[root@etcdhaproxy ~]# systemctl status etcd
● etcd.service - etcd key-value store
   Loaded: loaded (/etc/systemd/system/etcd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2023-01-24 08:17:32 CET; 32s ago
     Docs: https://github.com/etcd-io/etcd
    Main PID: 12134 (etcd)
      Tasks: 8 (limit: 49443)
     Memory: 10.4M
    CGroup: /system.slice/etcd.service
            └─12134 /usr/local/bin/etcd

Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.207+0100","caller":"etcdserver/server.go:2062","msg":"published local >
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.208+0100","caller":"etcdserver/server.go:2571","msg":"setting up initi>
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.209+0100","caller":"embed/serve.go:100","msg":"ready to serve client r>
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.209+0100","caller":"membership/cluster.go:584","msg":"set initial clus>
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.210+0100","caller":"api/capability.go:75","msg":"enabled capabilities >
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.210+0100","caller":"etcdserver/server.go:2595","msg":"cluster version >
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.210+0100","caller":"etcdmain/main.go:44","msg":"notifying init daemon"}

```

```

Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.211+0100","caller":"embed/serve.go:146","msg":"serving client traffic >
Jan 24 08:17:32 etcdhaproxy etcd[12134]:
{"level":"info","ts":"2023-01-24T08:17:32.211+0100","caller":"etcdmain/main.go:50","msg":"successfully notified >
Jan 24 08:17:32 etcdhaproxy systemd[1]:
Started etcd key-value store.
[root@etcdhaproxy ~]#

```

Install HAProxy

The following commands must be executed on HAProxy server (**etcdhaproxy** in our example).

Install HAProxy.

```

[root@etcdhaproxy ~]# yum install haproxy
Last metadata expiration check: 1:00:10 ago on Tue 24 Jan 2023 07:23:09 AM CET.
Dependencies resolved.
=====
Package                Architecture          Version              Repository           Size
=====
Installing:
haproxy                 x86_64                1.8.27-5.e18        appstream            1.4 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 1.4 M
Installed size: 4.2 M
Is this ok [y/N]: y
Downloading Packages:
haproxy-1.8.27-5.e18.x86_64.rpm                1.5 MB/s | 1.4 MB    00:00
-----
Total                                           1.2 MB/s | 1.4 MB    00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                :
  Running scriptlet: haproxy-1.8.27-5.e18.x86_64      1/1
  Installing              : haproxy-1.8.27-5.e18.x86_64      1/1
  Running scriptlet: haproxy-1.8.27-5.e18.x86_64      1/1
  Verifying                : haproxy-1.8.27-5.e18.x86_64      1/1

Installed:
haproxy-1.8.27-5.e18.x86_64

Complete!
[root@etcdhaproxy ~]#

```



CHAPTER 3 - Configuration

Configure ETCD

The following commands must be executed on ETCD server (`etcdhaproxy` in our example).

Create `/etc/etcd` directory and create the file `etcd.conf` to store ETCD configuration.

```
[root@etcdhaproxy ~]# mkdir /etc/etcd
[root@etcdhaproxy ~]#
```

```
[root@etcdhaproxy ~]# vi /etc/etcd/etcd.conf
```

Add the following content in the file

Note: IP addresses and ports have to be customized based on your configuration.

```
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380,http://127.0.0.1:7001"
ETCD_LISTEN_CLIENT_URLS="http://172.18.224.27:2379,http://127.0.0.1:2379"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://172.18.224.27:2380"
ETCD_INITIAL_CLUSTER="etcd0=http://172.18.224.27:2380,"
ETCD_ADVERTISE_CLIENT_URLS="http://172.18.224.27:2379"
ETCD_INITIAL_CLUSTER_TOKEN="cluster1"
ETCD_INITIAL_CLUSTER_STATE="new"
```

Restart ETCD service and check service status to confirm it is running.

```
[root@etcdhaproxy ~]# systemctl restart etcd
[root@etcdhaproxy ~]# systemctl status etcd
● etcd.service - etcd key-value store
   Loaded: loaded (/etc/systemd/system/etcd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2023-01-24 08:27:26 CET; 8s ago
     Docs: https://github.com/etcd-io/etcd
  Main PID: 12489 (etcd)
    Tasks: 7 (limit: 49443)
   Memory: 12.6M
    CGroup: /system.slice/etcd.service
            └─12489 /usr/local/bin/etcd

Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.555+0100","logger":"raft","caller":"etcdserver/zap_raft.go:77","msg":>
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.556+0100","logger":"raft","caller":"etcdserver/zap_raft.go:77","msg":>
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.560+0100","caller":"etcdserver/server.go:2062","msg":"published local >
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.560+0100","caller":"embed/serve.go:100","msg":"ready to serve client r>
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.561+0100","caller":"embed/serve.go:100","msg":"ready to serve client r>
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.563+0100","caller":"embed/serve.go:146","msg":"serving client traffic >
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.563+0100","caller":"embed/serve.go:146","msg":"serving client traffic >
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.563+0100","caller":"etcdmain/main.go:44","msg":"notifying init daemon"}
Jan 24 08:27:26 etcdhaproxy etcd[12489]: {"level":"info","ts":"2023-01-24T08:27:26.565+0100","caller":"etcdmain/main.go:50","msg":"successfully notified >
Jan 24 08:27:26 etcdhaproxy systemd[1]: Started etcd key-value store.
[root@etcdhaproxy ~]#
```

Configure Firewall (2379/tcp) to allow Patroni nodes to access to ETCD server & reload it.

```
[root@etcdhaproxy ~]# firewall-cmd --zone=public --permanent --add-port 2379/tcp
success
```

```
[root@etcdhaproxy ~]# firewall-cmd --reload
success
[root@etcdhaproxy ~]#
```

Configure Patroni

The following commands must be executed on both PostgreSQL nodes (`postgresql0` & `postgresql1` in our example).

A YAML file is used to store Patroni configuration.

Note: IP addresses and ports have to be customized based on your configuration.

The YAML patroni files used for our examples are the following:

- `patroni0.yml`
- `patroni1.yml`

Here is their content:

patroni0.yml

```
scope: miria
#namespace: /service/
name: postgresql0

restapi:
listen: 172.18.224.64:8008
connect_address: 172.18.224.64:8008
# cafile: /etc/ssl/certs/ssl-cacert-snakeoil.pem
# certfile: /etc/ssl/certs/ssl-cert-snakeoil.pem
# keyfile: /etc/ssl/private/ssl-cert-snakeoil.key
# authentication:
# username: username
# password: password

#ctl:
# insecure: false # Allow connections to Patroni REST API without verifying certificates
# certfile: /etc/ssl/certs/ssl-cert-snakeoil.pem
# keyfile: /etc/ssl/private/ssl-cert-snakeoil.key
# cacert: /etc/ssl/certs/ssl-cacert-snakeoil.pem

etcd3:
#Provide host to do the initial discovery of the cluster topology:
host: 172.18.224.27:2379
#Or use "hosts" to provide multiple endpoints
#Could be a comma separated string:
#hosts: host1:port1,host2:port2
#or an actual yaml list:
#hosts:
#- host1:port1
#- host2:port2
#Once discovery is complete Patroni will use the list of advertised clientURLs
#It is possible to change this behavior through by setting:
#use_proxies: true
```

```

#raft:
# data_dir: .
# self_addr: 127.0.0.1:2222
# partner_addrs:
# - 127.0.0.1:2223
# - 127.0.0.1:2224

bootstrap:
# this section will be written into Etcd:<namespace>/<scope>/config after initializing new
cluster
# and all other cluster members will use it as a `global configuration`
dcs:
ttl: 30
loop_wait: 10
retry_timeout: 10
maximum_lag_on_failover: 1048576
# master_start_timeout: 300
# synchronous_mode: false
#standby_cluster:
#host: 127.0.0.1
#port: 1111
#primary_slot_name: patroni
postgresql:
use_pg_rewind: true
# use_slots: true
parameters:
# wal_level: hot_standby
# hot_standby: "on"
# max_connections: 100
# max_worker_processes: 8
# wal_keep_segments: 8
# max_wal_senders: 10
# max_replication_slots: 10
# max_prepared_transactions: 0
# max_locks_per_transaction: 64
# wal_log_hints: "on"
# track_commit_timestamp: "off"
# archive_mode: "on"
# archive_timeout: 1800s
# archive_command: mkdir -p ../wal_archive && test ! -f ../wal_archive/%f && cp %p ../wal_
archive/%f
# recovery_conf:
# restore_command: cp ../wal_archive/%f %p

# some desired options for 'initdb'
initdb: # Note: It needs to be a list (some options need values, others are switches)
- encoding: UTF8
- data-checksums

pg_hba: # Add following lines to pg_hba.conf after running 'initdb'
# For kerberos gss based connectivity (discard @.*$)
#- host replication replicator 127.0.0.1/32 gss include_realm=0
#- host all all 0.0.0.0/0 gss include_realm=0
- host replication postgres 127.0.0.1/32 md5
- host replication postgres 172.18.224.64/0 md5
- host replication postgres 172.18.222.8/0 md5
- host all all 0.0.0.0/0 md5
# - hostssl all all 0.0.0.0/0 md5

```

```

# Additional script to be launched after initial cluster creation (will be passed the connection
URL as parameter)
# post_init: /usr/local/bin/setup_cluster.sh
# Some additional users users which needs to be created after initializing new cluster
users:
admin:
password: postgres
options:
- createrole
- createdb

postgresql:
listen: 172.18.224.64:5432
connect_address: 172.18.224.64:5432

# proxy_address: 127.0.0.1:5433 # The address of connection pool (e.g., pgbouncer) running next
to Patroni/Postgres. Only for service discovery.
data_dir: /var/lib/pgsql/13/data
bin_dir: /usr/pgsql-13/bin
# config_dir:
pgpass: /tmp/pgpass0
authentication:
replication:
username: postgres
password: postgres
superuser:
username: postgres
password: postgres
rewind: # Has no effect on postgres 10 and lower
username: postgres
password: postgres
# Server side kerberos spn
# krb_srvname: postgres
parameters:
# Fully qualified kerberos ticket file for the running user
# same as KRB5CCNAME used by the GSS
# krb_server_keyfile: /var/spool/keytabs/postgres
unix_socket_directories: '..' # parent directory of data_dir
# Additional fencing script executed after acquiring the leader lock but before promoting the
replica
#pre_promote: /path/to/pre_promote.sh
pg_hba: # Add following lines to pg_hba.conf
# For kerberos gss based connectivity (discard @.*$)
#- host replication replicator 127.0.0.1/32 gss include_realm=0
#- host all all 0.0.0.0/0 gss include_realm=0
- host replication postgres 127.0.0.1/32 md5
- host replication postgres 172.18.224.64/0 md5
- host replication postgres 172.18.222.8/0 md5
- host all all 0.0.0.0/0 md5
# - hostssl all all 0.0.0.0/0 md5

#watchdog:
# mode: automatic # Allowed values: off, automatic, required
# device: /dev/watchdog
# safety_margin: 5

tags:
nofailover: false

```

```
noloadbalance: false
clonefrom: false
nosync: false
```

patroni1.yml

```
scope: miria
#namespace: /service/
name: postgresql1

restapi:
listen: 172.18.222.8:8008
connect_address: 172.18.222.8:8008
# cafile: /etc/ssl/certs/ssl-cacert-snakeoil.pem
# certfile: /etc/ssl/certs/ssl-cert-snakeoil.pem
# keyfile: /etc/ssl/private/ssl-cert-snakeoil.key
# authentication:
# username: username
# password: password

#ctl:
# insecure: false # Allow connections to Patroni REST API without verifying certificates
# certfile: /etc/ssl/certs/ssl-cert-snakeoil.pem
# keyfile: /etc/ssl/private/ssl-cert-snakeoil.key
# cacert: /etc/ssl/certs/ssl-cacert-snakeoil.pem

etcd3:
#Provide host to do the initial discovery of the cluster topology:
host: 172.18.224.27:2379
#Or use "hosts" to provide multiple endpoints
#Could be a comma separated string:
#hosts: host1:port1,host2:port2
#or an actual yaml list:
#hosts:
#- host1:port1
#- host2:port2
#Once discovery is complete Patroni will use the list of advertised clientURLs
#It is possible to change this behavior through by setting:
#use_proxies: true

#raft:
# data_dir: .
# self_addr: 127.0.0.1:2222
# partner_addrs:
# - 127.0.0.1:2223
# - 127.0.0.1:2224

bootstrap:
# this section will be written into Etcd:<namespace>/<scope>/config after initializing new
cluster
# and all other cluster members will use it as a `global configuration`
dcs:
ttl: 30
loop_wait: 10
retry_timeout: 10
maximum_lag_on_failover: 1048576
# master_start_timeout: 300
# synchronous_mode: false
#standby_cluster:
```

```

#host: 127.0.0.1
#port: 1111
#primary_slot_name: patroni
postgresql:
use_pg_rewind: true
# use_slots: true
parameters:
# wal_level: hot_standby
# hot_standby: "on"
# max_connections: 100
# max_worker_processes: 8
# wal_keep_segments: 8
# max_wal_senders: 10
# max_replication_slots: 10
# max_prepared_transactions: 0
# max_locks_per_transaction: 64
# wal_log_hints: "on"
# track_commit_timestamp: "off"
# archive_mode: "on"
# archive_timeout: 1800s
# archive_command: mkdir -p ../wal_archive && test ! -f ../wal_archive/%f && cp %p ../wal_
archive/%f
# recovery_conf:
# restore_command: cp ../wal_archive/%f %p

# some desired options for 'initdb'
initdb: # Note: It needs to be a list (some options need values, others are switches)
- encoding: UTF8
- data-checksums

pg_hba: # Add following lines to pg_hba.conf after running 'initdb'
# For kerberos gss based connectivity (discard @.*$)
#- host replication replicator 127.0.0.1/32 gss include_realm=0
#- host all all 0.0.0.0/0 gss include_realm=0
- host replication postgres 127.0.0.1/32 md5
- host replication postgres 172.18.222.8/0 md5
- host replication postgres 172.18.224.64/0 md5
- host all all 0.0.0.0/0 md5
# - hostssl all all 0.0.0.0/0 md5

# Additional script to be launched after initial cluster creation (will be passed the connection
URL as parameter)
# post_init: /usr/local/bin/setup_cluster.sh
# Some additional users users which needs to be created after initializing new cluster
users:
admin:
password: postgres
options:
- createrole
- createdb

postgresql:
listen: 172.18.222.8:5432
connect_address: 172.18.222.8:5432

# proxy_address: 127.0.0.1:5433 # The address of connection pool (e.g., pgbouncer) running next
to Patroni/Postgres. Only for service discovery.
data_dir: /var/lib/pgsql/13/data
bin_dir: /usr/pgsql-13/bin

```

```

# config_dir:
pgpass: /tmp/pgpass0
authentication:
replication:
username: postgres
password: postgres
superuser:
username: postgres
password: postgres
rewind: # Has no effect on postgres 10 and lower
username: postgres
password: postgres
# Server side kerberos spn
# krbsrvname: postgres
parameters:
# Fully qualified kerberos ticket file for the running user
# same as KRB5CCNAME used by the GSS
# krb_server_keyfile: /var/spool/keytabs/postgres
unix_socket_directories: '..' # parent directory of data_dir
# Additional fencing script executed after acquiring the leader lock but before promoting the
replica
#pre_promote: /path/to/pre_promote.sh
pg_hba: # Add following lines to pg_hba.conf
# For kerberos gss based connectivity (discard @.*$)
#- host replication replicator 127.0.0.1/32 gss include_realm=0
#- host all all 0.0.0.0/0 gss include_realm=0
- host replication postgres 127.0.0.1/32 md5
- host replication postgres 172.18.222.8/0 md5
- host replication postgres 172.18.224.64/0 md5
- host all all 0.0.0.0/0 md5
# - hostssl all all 0.0.0.0/0 md5

#watchdog:
# mode: automatic # Allowed values: off, automatic, required
# device: /dev/watchdog
# safety_margin: 5

tags:
nofailover: false
noloadbalance: false
clonefrom: false
nosync: false

```

You can download templates to customize in the following `patroni0 - template.yml` and the `patroni1 - template.yml` files:

- [patroni0 - template.yml](#)
- [patroni1 - template.yml](#)
- [patroni0 - template.yml](#)
- [patroni1 - template.yml](#)
- [patroni0 - template.yml](#)

Upload the YAML files on each postgresql node. Copy the file content in `/etc/patroni.yml`.

```
[root@postgresql ~]# pwd
/root
[root@postgresql ~]# ls
anaconda-ks.cfg  patroni0.yml
[root@postgresql ~]#
[root@postgresql ~]# cp patroni0.yml /etc/patroni.yml
```

Create a new systemd service file for patroni.

```
[root@postgresql ~]# vi /etc/systemd/system/patroni.service
```

Copy into the file the following content.

```
[Unit]
Description=Runners to orchestrate a high-availability PostgreSQL
After=syslog.target network.target

[Service]
Type=simple
User=postgres
Group=postgres
ExecStart=/usr/local/bin/patroni /etc/patroni.yml
KillMode=process
TimeoutSec=30
Restart=no

[Install]
WantedBy=multi-user.target
```

Reload systemd service.

```
[root@postgresql ~]# systemctl daemon-reload
[root@postgresql ~]#
```

Use `--validate-config` option in order to validate Patroni configuration.

```
[root@postgresql ~]# patroni --validate-config /etc/patroni.yml
[root@postgresql ~]#
```

List all nodes and it's role, status. You can use it for checking status of all nodes, which is the master and which all are slaves/replicas.

Before starting the first node, the list of members must be empty.

```
[root@postgresql ~]# patronictl -c /etc/patroni.yml list
+ Cluster: miria (uninitialized) ---+-----+
| Member | Host | Role | State | TL | Lag in MB |
+-----+-----+-----+-----+---+-----+
+-----+-----+-----+-----+---+-----+
[root@postgresql ~]#
```

PostgreSQL service must be inactive as Patroni will do the rolling start for PostgreSQL.

```
[root@postgresql ~]# systemctl stop postgresql-13
[root@postgresql ~]# systemctl status postgresql-13
• postgresql-13.service - PostgreSQL 13 database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql-13.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://www.postgresql.org/docs/13/static/

Jan 23 16:23:53 postgresql0 systemd[1]:
Starting PostgreSQL 13 database server...
Jan 23 16:23:53 postgresql0 postmaster[13254]:
2023-01-23 16:23:53.897 CET [13254] LOG:  redirecting log output to logging collector process
Jan 23 16:23:53 postgresql0 postmaster[13254]:
2023-01-23 16:23:53.897 CET [13254] HINT:  Future log output will appear in directory "log".
Jan 23 16:23:53 postgresql0 systemd[1]:
Started PostgreSQL 13 database server.
Jan 24 09:00:23 postgresql0 systemd[1]:
```

```
Stopping PostgreSQL 13 database server...
Jan 24 09:00:23 postgresql0 systemd[1]:
postgresql-13.service: Killing process 13255 (postmaster) with signal SIGKILL.
Jan 24 09:00:23 postgresql0 systemd[1]:
postgresql-13.service: Succeeded.
Jan 24 09:00:23 postgresql0 systemd[1]:
Stopped PostgreSQL 13 database server.
[root@postgresql0 ~]#
```

Enable service to start when system is rebooted.

```
[root@postgresql0 ~]# systemctl enable patroni.service
Created symlink /etc/systemd/system/multi-user.target.wants/patroni.service →
/etc/systemd/system/patroni.service.
[root@postgresql0 ~]#
```

Configure HAProxy

The following commands must be executed on HAProxy server (`etcdhaproxy` in our example).

A YAML file is used to store HAProxy configuration

Note: IP addresses and ports have to be customized based on your configuration.

Here is our example of an `haproxy.cfg` file on `etcdhaproxy`:

```
#-----
# Example configuration for a possible web application. See the
# full configuration options online.
#
# https://www.haproxy.org/download/1.8/doc/configuration.txt
#
#-----
#-----
# Global settings
#-----
global
# to have these messages end up in /var/log/haproxy.log you will
# need to:
#
# 1) configure syslog to accept network log events. This is done
# by adding the '-r' option to the SYSLOGD_OPTIONS in
# /etc/sysconfig/syslog
#
# 2) configure local2 events to go to the /var/log/haproxy.log
# file. A line like the following can be added to
# /etc/sysconfig/syslog
#
# local2.* /var/log/haproxy.log
#
log 127.0.0.1 local2

chroot /var/lib/haproxy
pidfile /var/run/haproxy.pid
maxconn 100
user haproxy
group haproxy
```

```

daemon

# turn on stats unix socket
stats socket /var/lib/haproxy/stats
# utilize system-wide crypto-policies
ssl-default-bind-ciphers PROFILE=SYSTEM
ssl-default-server-ciphers PROFILE=SYSTEM

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----
defaults
mode tcp
log global
# option httplog
# option dontlognull
# option http-server-close
# option forwardfor except 127.0.0.0/8
# option redispatch
retries 2
# timeout http-request 10s
# timeout queue 1m
timeout connect 4s
timeout client 30m
timeout server 30m
# timeout http-keep-alive 10s
timeout check 5s
# maxconn 3000

listen stats
mode http
bind *:7000
stats enable
stats uri /

listen miria
bind *:5000
option httpchk
http-check expect status 200
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server postgresql0 172.18.224.64:5432 maxconn 100 check port 8008
server postgresql1 172.18.222.8:5432 maxconn 100 check port 8008

```

You can download the following template to customize in the haproxy - template.cfg file:

- [haproxy - template.cfg](#)
- [haproxy - template.cfg](#)

Save the default current haproxy configuration file and copy the uploaded file in **/etc/haproxy/haproxy.cfg**.

```
[root@etcdhaproxy ~]# cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.orig
```

```

[root@etcdhaproxy ~]# pwd
/root
[root@etcdhaproxy ~]# ls
anaconda-ks.cfg  etcd-v3.5.7-linux-amd64  etcd-v3.5.7-linux-amd64.tar.gz  haproxy.cfg  my-etcd.pp  my-etcd.te
[root@etcdhaproxy ~]# cp haproxy.cfg /etc/haproxy/haproxy.cfg
[root@etcdhaproxy ~]#

```

Start HAProxy service.

```
[root@etcdhaproxy ~]# systemctl start haproxy
Job for haproxy.service failed because the control process exited with error code.
See "systemctl status haproxy.service" and "journalctl -xe" for details.
[root@etcdhaproxy ~]#
```

```
[root@etcdhaproxy ~]# systemctl status haproxy
• haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
  Active: failed (Result: exit-code) since Thu 2023-01-26 17:24:29 CET; 7s ago
  Process: 11948 ExecStart=/usr/sbin/haproxy -Ws -f $CONFIG -f $CFGDIR -p $PIDFILE $OPTIONS (code=exited, status=1/FAILURE)
  Process: 11945 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -f $CFGDIR -c -q $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 11948 (code=exited, status=1/FAILURE)

Jan 26 17:24:29 etcdhaproxy systemd[1]: Starting HAProxy Load Balancer...
Jan 26 17:24:29 etcdhaproxy haproxy[11948]: [ALERT] 025/172429 (11948) : Starting proxy stats: cannot bind socket [0.0.0.0:7000]
Jan 26 17:24:29 etcdhaproxy systemd[1]: haproxy.service: Main process exited, code=exited, status=1/FAILURE
Jan 26 17:24:29 etcdhaproxy systemd[1]: haproxy.service: Failed with result 'exit-code'.
Jan 26 17:24:29 etcdhaproxy systemd[1]: Failed to start HAProxy Load Balancer.
[root@etcdhaproxy ~]#
```

If the previous command fails (as shown above), it may be due to SELinux in enforcing mode.

```
[root@etcdhaproxy ~]# setsebool -P haproxy_connect_any=1
[root@etcdhaproxy ~]#
```

The HAProxy service is enabled and started as follows.

```
[root@etcdhaproxy ~]# systemctl enable haproxy
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /usr/lib/systemd/system/haproxy.service.
[root@etcdhaproxy ~]# systemctl start haproxy
[root@etcdhaproxy ~]#
```

Configure Firewall (5000/tcp) & reload it to allow Miria Server to access to HAProxy server.

```
[root@etcdhaproxy ~]# firewall-cmd --zone=public --permanent --add-port 5000/tcp
success
[root@etcdhaproxy ~]# firewall-cmd --reload
success
[root@etcdhaproxy ~]#
```

Configure Firewall (7000/tcp) & reload it to allow HAProxy web monitoring (optional).

```
[root@etcdhaproxy ~]# firewall-cmd --zone=public --permanent --add-port 7000/tcp
success
[root@etcdhaproxy ~]# firewall-cmd --reload
success
[root@etcdhaproxy ~]#
```

In a web browser, launch `<etcdhaproxy IP@>:7000` to open HAProxy monitoring page.

HAProxy version 1.8.27-493ce0b, released 2020/11/06

Statistics Report for pid 12591

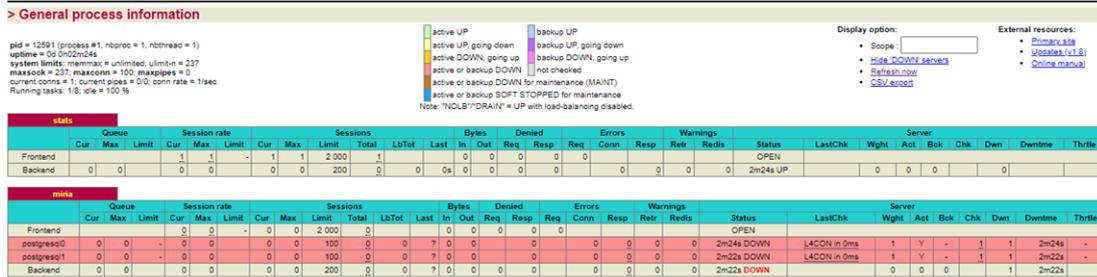


Figure 2: HAProxy monitoring page

On Miria Server

Export the content of the Miria & AMM databases into files.

```
[root@miriaserver]# cd /Miria/Database/PostgreSQL/bin
[root@miriaserver bin]# ./pg_dump -d ADA -U postgres -p 5433 > /root/miriaserver.atempo.dev_5433_DATA_Backup
[root@miriaserver bin]# cd /Miria/AMM/Database/PostgreSQL/bin
[root@miriaserver bin]# ./pg_dump -d AMM -U postgres -p 15432 > /root/miriaserver.atempo.dev_15432_DATA_Backup
```

On one of PostgreSQL node

In the following example, it occurs on a postgresql0 node:

Edit postgres HBA configuration file.

```
[root@postgresql0 ~]# vi /var/lib/pgsql/13/data/pg_hba.conf
```

Replace "peer" by "trust" on "local all" line in order to have the following. "local" is for Unix domain socket connections only.

```
local all all trust
```

Restart Postgresql.

```
[root@postgresql0 ~]# systemctl restart postgresql-13
[root@postgresql0 ~]#
```

Execute the following SQL commands.

```
[root@postgresql0 ~]# psql -U postgres
psql (13.9)
Type "help" for help.

postgres=# CREATE ROLE "ADA" LOGIN PASSWORD 'ADA' SUPERUSER INHERIT CREATEDB CREATEROLE;
CREATE ROLE
postgres=# CREATE DATABASE "ADA" WITH OWNER="ADA" ENCODING='UTF8';
CREATE DATABASE
postgres=# CREATE DATABASE "AMM" ENCODING='UTF8';
CREATE DATABASE
postgres=# quit
[root@postgresql0 ~]#
```

Import the Miria database file exported on the miriaserver.

```
[root@postgresql ~]# psql -U postgres -d ADA -f /root/miriaserver.atempo.dev_5433_DATA_Backup
--
-- PostgreSQL database dump
--

-- Dumped from database version 13.0
-- Dumped by pg_dump version 13.0

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
[...]
```

Import the AMM database file exported on the miriaserver.

```
[root@postgresql ~]# psql -U postgres -d AMM -f /root/miriaserver.atempo.dev_15432_DATA_Backup
SET
SET
SET
SET
SET
set_config
-----
[...]
```

Revert postgres HBA configuration file modifications.

```
[root@postgresql ~]# vi /var/lib/pgsql/13/data/pg_hba.conf
```

```
local all all peer
```

Stop Postgresql. Patroni will manage the start of PostgreSQL database.

```
[root@postgresql ~]# systemctl stop postgresql-13
[root@postgresql ~]# systemctl status postgresql-13
• postgresql-13.service - PostgreSQL 13 database server
  Loaded: loaded (/usr/lib/systemd/system/postgresql-13.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
  Docs: https://www.postgresql.org/docs/13/static/

Jan 26 16:21:26 postgresql0 systemd[1]: postgresql-13.service: Failed with result 'exit-code'.
Jan 26 16:21:26 postgresql0 systemd[1]: Failed to start PostgreSQL 13 database server.
Jan 26 16:35:57 postgresql0 systemd[1]: Starting PostgreSQL 13 database server...
Jan 26 16:35:57 postgresql0 postmaster[15980]: 2023-01-26 16:35:57.350 CET [15980] LOG: redirecting log output to logging collector>
Jan 26 16:35:57 postgresql0 postmaster[15980]: 2023-01-26 16:35:57.350 CET [15980] HINT: Future log output will appear in directory>
Jan 26 16:35:57 postgresql0 systemd[1]: Started PostgreSQL 13 database server.
Jan 26 16:41:28 postgresql0 systemd[1]: Stopping PostgreSQL 13 database server...
Jan 26 16:41:28 postgresql0 systemd[1]: postgresql-13.service: Killing process 15981 (postmaster) with signal SIGKILL.
Jan 26 16:41:28 postgresql0 systemd[1]: postgresql-13.service: Succeeded.
Jan 26 16:41:28 postgresql0 systemd[1]: Stopped PostgreSQL 13 database server.
[root@postgresql ~]#
```

Configure Firewall (8008/tcp) & reload it to allow HAProxy connections.

```
[root@postgresql ~]# firewall-cmd --zone=public --permanent --add-port 8008/tcp
success
[root@postgresql ~]# firewall-cmd --reload
success
[root@postgresql ~]#
```

Configure Firewall (5432/tcp) & reload it to allow postgresql connections from HAProxy.

```
[root@postgresql ~]# firewall-cmd --zone=public --permanent --add-port 5432/tcp
success
[root@postgresql ~]# firewall-cmd --reload
success
[root@postgresql ~]#
```

Start Patroni: current postgresql node must appear in the list as leader (first node where patroni is launched).

```
[root@postgresql ~]# systemctl start patroni
[root@postgresql ~]# patronictl -c /etc/patroni.yml list
+ Cluster: miria (7192189566009963397) +-----+-----+
| Member      | Host          | Role  | State | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| postgresql0 | 172.18.224.64 | Leader | running | 2 |           |
+-----+-----+-----+-----+-----+-----+
[root@postgresql ~]#
```

```
[root@postgresql ~]# tail -10 /var/log/messages
Jan 26 16:43:23 postgresql0 patroni[16112]:
2023-01-26 16:43:23,369 INFO: Software Watchdog activated with 25 second timeout, timing slack 15 seconds
Jan 26 16:43:23 postgresql0 patroni[16148]:
server promoting
Jan 26 16:43:23 postgresql0 patroni[16112]:
2023-01-26 16:43:23,466 INFO: promoted self to leader by acquiring session lock
Jan 26 16:43:23 postgresql0 patroni[16112]:
2023-01-26 16:43:23,478 INFO: cleared rewind state after becoming the leader
Jan 26 16:43:24 postgresql0 patroni[16112]:
2023-01-26 16:43:23,474 INFO: Lock owner: postgresql0; I am postgresql0
Jan 26 16:43:24 postgresql0 patroni[16112]:
2023-01-26 16:43:24,158 INFO: updated leader lock during promote
Jan 26 16:43:24 postgresql0 patroni[16112]:
2023-01-26 16:43:24,809 INFO: no action. I am (postgresql0), the leader with the lock
Jan 26 16:43:34 postgresql0 patroni[16112]:
2023-01-26 16:43:34,730 INFO: no action. I am (postgresql0), the leader with the lock
Jan 26 16:43:44 postgresql0 patroni[16112]:
2023-01-26 16:43:44,696 INFO: no action. I am (postgresql0), the leader with the lock
Jan 26 16:43:54 postgresql0 patroni[16112]:
2023-01-26 16:43:54,671 INFO: no action. I am (postgresql0), the leader with the lock
[root@postgresql ~]#
```

On Other PostgreSQL Nodes

In the following example, it occurs on a postgresql1 node:

Make a copy of current database.

Then delete it. The leader will replicate the database content to the other nodes when patroni will be started.

```
[root@postgresql1 ~]# cp -r /var/lib/pgsql/13/data /var/lib/pgsql/13/data.orig
[root@postgresql1 ~]# rm -fr /var/lib/pgsql/13/data/*
```

Configure Firewall (8008/tcp) & reload it to allow HAProxy connections.

```
[root@postgresql0 ~]# firewall-cmd --zone=public --permanent --add-port 8008/tcp
success
[root@postgresql0 ~]# firewall-cmd --reload
success
[root@postgresql0 ~]#
```

Configure Firewall (5432/tcp) & reload it to allow postgresql connections from HAProxy.

```
[root@postgresql0 ~]# firewall-cmd --zone=public --permanent --add-port 5432/tcp
```

```
success
[root@postgresq10 ~]# firewall-cmd --reload
success
[root@postgresq10 ~]#
```

Start Patroni. Check that it is shown as replica role.

```
[root@postgresq11 ~]# systemctl start patroni
[root@postgresq11 ~]# patronictl -c /etc/patroni.yml list
+ Cluster: miria (7192189566009963397) +-----+
| Member      | Host          | Role   | State  | TL | Lag in MB |
+-----+
| postgresq10 | 172.18.224.64 | Leader | running | 2 |           |
| postgresq11 | 172.18.222.8  | Replica | running | 2 |           |
+-----+
[root@postgresq11 ~]#
```

```
[root@postgresq11 ~]# tail -10 /var/log/messages
Jan 26 16:45:05 postgresq11 patroni[7124]:
2023-01-26 16:45:05.475 CET [7124] HINT: Future log output will appear in directory "log".
Jan 26 16:45:05 postgresq11 patroni[7098]:
2023-01-26 16:45:05.478 INFO: postmaster pid=7124
Jan 26 16:45:05 postgresq11 patroni[7126]:
172.18.222.8:5432 - rejecting connections
Jan 26 16:45:05 postgresq11 patroni[7129]:
172.18.222.8:5432 - rejecting connections
Jan 26 16:45:06 postgresq11 patroni[7135]:
172.18.222.8:5432 - accepting connections
Jan 26 16:45:06 postgresq11 patroni[7098]:
2023-01-26 16:45:06.727 INFO: Lock owner: postgresq10; I am postgresq11
Jan 26 16:45:06 postgresq11 patroni[7098]:
2023-01-26 16:45:06.729 INFO: establishing a new patroni connection to the postgres cluster
Jan 26 16:45:06 postgresq11 patroni[7098]:
2023-01-26 16:45:06.935 INFO: no action. I am (postgresq11), a secondary, and following a leader (postgresq10)
Jan 26 16:45:16 postgresq11 patroni[7098]:
2023-01-26 16:45:16.765 INFO: no action. I am (postgresq11), a secondary, and following a leader (postgresq10)
Jan 26 16:45:24 postgresq11 patroni[7098]:
2023-01-26 16:45:24.773 INFO: no action. I am (postgresq11), a secondary, and following a leader (postgresq10)
[root@postgresq11 ~]#
```

In a Web browser launch `<etcdhaproxy IP@>:7000` to open HAProxy monitoring page. The leader is the green line, where replica is in red.

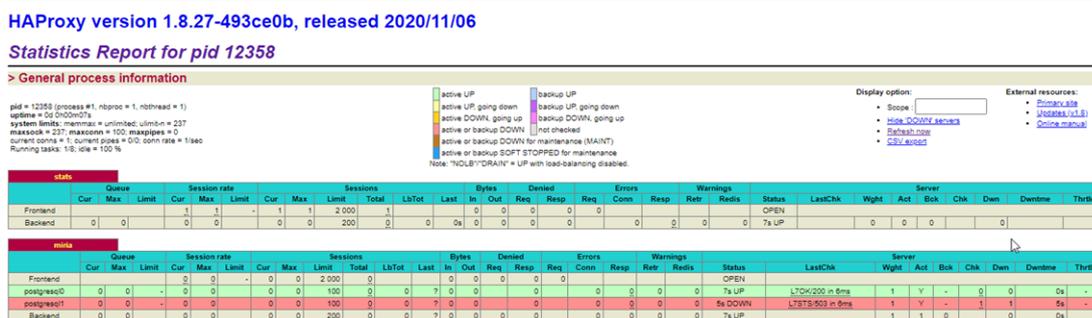


Figure 3: HAProxy monitoring page

On Miria Server

Switch Miria configuration from local database to external one. The target hostname is the haproxy server hostname.

```
[root@miriaserver Bin]# cd /Miria/Binary/Bin
[root@miriaserver Bin]# ./ada_service -switch_data_model -db_admin_identity postgres:postgres -db_name ADA -db_port 5000 -db_hostname etcdhaproxy.atempo.dev
Connection to external DB successfully, SSL:off, mode:prefer
```

```
-----  
Host:          etcdhaproxy.atempo.dev  
DB Name:       ADA  
Port:          5000  
Admin user:    postgres  
-----  
Are you sure to switch into external postgres ? (y/n)  
y  
Database switched successfully  
[root@miriaserver Bin]#
```

Switch AMM configuration from local database to external one. The target hostname is the haproxy server hostname.

```
[root@miriaserver Bin]# cd /Miria/AMM/bin  
[root@miriaserver bin]# /Miria/Binary/3rdParty/python/bin/python3 amm_update_config.py -n etcdhaproxy.atempo.dev -d AMM  
-u postgres -w postgres -p 5000 -s prefer  
Database port number already set. Override it? [y/N] y  
[root@miriaserver bin]#
```